

Ansible Modules for Dell EMC PowerMax

Version 1.0

Product Guide

302-005-901

01

July 2019

Copyright © 2019 Dell Inc. or its subsidiaries. All rights reserved.

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

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

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

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

CHAPTER 1

Introduction

This chapter contains the following topics:

- [Product overview](#)4

Product overview

The Ansible Modules for Dell EMC PowerMax are used to automate and orchestrate the configuration, and deployment of the Dell EMC PowerMax arrays. The capabilities of Ansible modules are managing volumes, storage groups, ports, port group, hosts, host groups, masking views, snapshots, and gather high-level facts about the arrays. The options available for each capability are list, show, create, delete, and modify.

The Ansible modules are called by tasks within the Ansible playbooks. The *Idempotency* feature is enabled for all the modules. The *Idempotency* feature enables the playbook to be run multiple times. The modules use Unisphere RESTAPI to interface with the PowerMax arrays.

List of Ansible Modules for Dell EMC PowerMax

The following are the list of modules:

- Volume module
- Storage group module
- Port module
- Port group module
- Host module
- Host group module
- Masking view module
- Gather facts module
- Snapshot module

CHAPTER 2

Configure Ansible

This chapter contains the following topics:

- [Software Prerequisites](#) 6
- [Setting up Ansible servers](#) 6

Software Prerequisites

This topic provides information about the software prerequisites for the Ansible Modules for Dell EMC PowerMax.

Prerequisites

- Unisphere for PowerMax 9.0 to manage Dell EMC PowerMax arrays.
- Ansible 2.6 or higher.
- Red Hat Enterprise Linux 7.5.
- Python 2.7.12 or higher.
- Python library for Unisphere (PyU4V) 3.0.0.14 must be installed on the client operating system.

Setting up Ansible servers

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

About this task

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

Procedure

1. Install Python library for Unisphere.

Run the following command to install PyU4V:

```
sudo apt install python-pip
pip install PyU4V
```

2. Create a folder `dellemc` if it is not available.
3. Copy the Ansible modules to the appropriate locations in the virtual machine.
 - a. Copy *dellemc_ansible_utils.py* to `/usr/lib/python2.7/site-packages/ansible/module_utils/`
 - b. Copy all the module python files to `/usr/lib/python2.7/site-packages/ansible/modules/storage/dellemc`

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

CHAPTER 3

Ansible modules for Dell EMC PowerMax

This chapter presents the following topics:

• Gather Facts module	8
• Storage group module	13
• Volume module	19
• Host module	23
• Host group module	28
• Port module	34
• Port Group module	35
• Masking View module	40
• Snapshot module	43

Gather Facts module

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

The Gather Facts module supports two sets of operations.

- Operations on the Unisphere server
- Operations on the PowerMax array registered under an Unisphere server

On the Unisphere host, the Gather Facts module lists the registered storage arrays that are managed by the Unisphere host.

On the PowerMax array, the Gather Facts module lists the following entities:

- Arrays in an Unisphere host
- Volumes in a specific array
- Storage groups in a specific array
- Hosts in a specific array
- Host groups in a specific array
- Storage resource pools in a specific array
- Ports in a specific array
- Port groups in a specific array
- Masking views in a specific array

Get list of volumes

The user can get the list of volumes in the storage array by running the appropriate playbook.

The syntax of the playbook is as follows:

```
- name: Get list of volumes
  dell EMC_powermax_gatherfacts:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    gather_subset:
      - vol
```

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

Get list of arrays

The user can get the list of arrays in the storage array by running the appropriate playbook.

The syntax for the playbook is shown as follows:

```
- name: Get array list
```



```
dellemc_powermax_gatherfacts:
  unispherehost: "{{unispherehost}}"
  universion: "{{universion}}"
  verifycert: "{{verifycert}}"
  user: "{{user}}"
  password: "{{password}}"
```

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

Get list of storage group

The user can get the list of storage groups in the array by running the appropriate playbook

The syntax for the playbook is shown as follows:

```
- name: Get list of Storage groups
  dellemc_powermax_gatherfacts:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    gather_subset:
      - sg
```

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

Get list of storage resource pool

The user can get the list of storage resource pool in the array by running the appropriate playbook.

The syntax for the playbook is shown as follows:

```
- name: Get list of Storage Resource Pools
  dellemc_powermax_gatherfacts:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    gather_subset:
      - srp
```

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

Get list of port groups

The user can get the list of port groups in the array by running the appropriate playbook

The syntax for the playbook is shown as follows:

```
- name: Get list of Port Groups
  dellemc_powermax_gatherfacts:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    gather_subset:
      - pg
```

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

Get list of hosts

The user can get the list of hosts in the array by running the appropriate playbook

The syntax for the playbook is shown as follows:

```
- name: Get list of Hosts
  dellemc_powermax_gatherfacts:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    gather_subset:
      - host
```

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

Get list of host groups

The user can get the list of host groups in the array by running the appropriate playbook

The syntax for the playbook is shown as follows:

```
- name: Get list of Host Groups
  dellemc_powermax_gatherfacts:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
```

```
gather_subset:
  - hg
```

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

Get list of ports

The user can get the list of ports in the array by running the appropriate playbook.

The syntax for the playbook is shown as follows:

```
- name: Get list of Ports
  dellemc_powermax_gatherfacts:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    gather_subset:
      - port
```

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

Get list of masking view

The user can get the list of masking views available in the storage array by running the appropriate playbook.

The syntax for the playbook is shown as follows:

```
- name: Get list of Maskng Views
  dellemc_powermax_gatherfacts:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    gather_subset:
      - mv
```

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

Parameters

The following table provides the information about the parameters that are displayed on the console, when the user runs the *playbook* using the Gather facts module:

Table 1 Parameters

Parameter	Choices/default	Comments
<code>unisphrehost</code>		IP or FQDN of the Unisphere host. This parameter is mandatory.
<code>universion</code>		The version of the Unisphere software. The version is 90. This parameter is mandatory.
<code>verifycert</code>	<ul style="list-style-type: none"> • Yes • No 	<p>To validate the SSL certificate.</p> <ul style="list-style-type: none"> • True - Verify the SSL certificate. • False - Do not verify the SSL certificate. <p>This parameter is mandatory.</p>
<code>user</code>		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
<code>password</code>		The password to access the Unisphere server. The password can be encrypted using Ansible vault. This parameter is mandatory.
<code>serial_no</code>		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
<code>gather_subset</code>	<ul style="list-style-type: none"> • vol • srp • sg • pg • host • hg • port 	<p>List of string variables to specify the PowerMax entities for which the information is required. Required only if <code>serial_no</code> is present.</p> <ul style="list-style-type: none"> • vol - volumes

Table 1 Parameters (continued)

Parameter	Choices/default	Comments
	<ul style="list-style-type: none"> mv 	<ul style="list-style-type: none"> srp - storage resource pools sg - storage groups pg - port groups host - hosts hg - host groups port - ports mv - masking views

Storage group module

The Storage group module manages the storage groups available on the PowerMax storage array.

The storage group module has the following functionalities:

- List the volumes of a storage group.
- Create a storage group.
- Delete an existing storage group.
- Add existing volumes to a storage group.
- Remove existing volumes from a storage group.
- Create volumes in a storage group.
- Modify the storage group attributes.
- Add child storage group to a parent storage group.
- Remove child storage group from a parent storage group.

Get details of storage group including the volumes

The user can get the list of storage group in the array. This list contains information about the volumes within the storage group.

The syntax of the playbook is shown as follows:

```
- name: Get storage group details including volumes
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "ansible_sg":
    state: "present"
```

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

Create empty storage group

The user can create an empty storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create empty storage group
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "foo"
    service_level: "Diamond"
    srp: "SRP_1"
    compression: True
    state: "present"
```

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

Delete empty storage group

The user can delete an empty storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Delete the storage Group
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "foo"
    state: "absent"
```

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

Add existing volumes to the existing storage group

The user can add one or more existing volumes to a storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Adding existing volume(s) to existing SG
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
```

```

verifycert: "{{verifycert}}"
user: "{{user}}"
password: "{{password}}"
serial_no: "{{serial_no}}"
sg_name: "foo"
state: "present"
volumes:
  - vol_id: "00028"
  - vol_id: "00018"
  - vol_id: "00025"
vol_state: "present-in-group"

```

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

Create new volumes for existing storage group

The user can create new volumes for an existing storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```

- name: Create new volumes for existing SG
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "foo"
    state: "present"
    volumes:
      - vol_name: "foo"
        size: 1
        cap_unit: "GB"
      - vol_name: "bar"
        size: 1
        cap_unit: "GB"
    vol_state: "present-in-group"

```

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

Remove volumes from storage group

The user can remove one or multiple volumes from a storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```

- name: Remove volume(s) from existing SG
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"

```

```
sg_name: "foo"
state: "present"
volumes:
  - vol_id: "00028"
  - vol_id: "00018"
  - vol_name: "ansible-vol"
vol_state: "absent-in-group"
```

The parameters must be set before the user runs the playbook. See the Parameters table for more information about the parameters.

Add child storage group to parent storage group

The user can add a child storage group to a parent storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Adding child SG to parent SG
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "parent_sg"
    state: "present"
    child_storage_groups:
      - "pie"
      - "bar"
    child_sg_state: "present-in-group"
```

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

Remove child storage group from parent storage group

The user can remove a child storage group from a parent storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Removing child SG from parent SG
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "parent_sg"
    state: "present"
    child_storage_groups:
      - "pie"
      - "bar"
    child_sg_state: "absent-in-group"
```


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

Rename storage group

The user can rename a storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Rename Storage Group
  dellemc_powermax_storagegroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "ansible_sg":
    new_sg_name: "ansible_sg_renamed"
    state: "present"
```

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

Parameters

The following table lists the parameters that must be set before the user runs the playbook for the Storage Group module:

Table 2 Parameters

Parameter	Choices/default	Comments
unispherehost		IP or FQDN of the Unisphere host. This parameter is mandatory.
universion		The version of the Unisphere software. The version is 90. This parameter is mandatory.
verifycert	<ul style="list-style-type: none"> Yes No 	<p>To validate the SSL certificate.</p> <ul style="list-style-type: none"> True - Verify the SSL certificate. False - Do not verify the SSL certificate. <p>This parameter is mandatory.</p>
user		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
password		The password to access the Unisphere server. The

Table 2 Parameters (continued)

Parameter	Choices/default	Comments
		password can be encrypted using Ansible vault. This parameter is mandatory.
<code>serial_no</code>		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
<code>sg_name</code>		The name of the storage group. This parameter is mandatory.
<code>new_sg_name</code>		The new name of the storage group.
<code>service_level</code>		The name of the service level
<code>srp</code>		The name of the storage resource pool. This parameter is ignored if the <code>service_level</code> is not specified. The default value is the name of the default storage resource pool of the array.
<code>compression</code>	<ul style="list-style-type: none"> <code>true</code> <code>false</code> The default value is true.	Defines if the compression is enable for a storage group. This parameter is unavailable if the <code>service_level</code> is not specified.
<code>state</code>	<ul style="list-style-type: none"> <code>absent</code> <code>present</code> 	Defines whether the storage group should exist or not. This parameter is mandatory.
<code>volumes</code>		<p>This parameter lists the volumes present in the storage group. Each volume has four attributes.</p> <ul style="list-style-type: none"> <code>vol_name</code> <code>size</code> <code>cap_unit</code> <code>vol_id</code> <p>Use either volume name or volume ID to identify the volumes. Both cannot be used</p>

Table 2 Parameters (continued)

Parameter	Choices/default	Comments
		simultaneously to identify the volumes. For new volumes that are added to storage group, the name and size must be provided. This parameter is optional.
vol_state	<ul style="list-style-type: none"> present-in-group absent-in-group 	Defines the state of the volumes inside the storage group.
child_storage_groups		Lists the child storage group.
child_sg_state	<ul style="list-style-type: none"> present-in-group absent-in-group 	Describes the state of child storage group in a parent storage group.

Volume module

The Volume module manages the storage volumes on PowerMax arrays.

The volume module has the following capabilities:

- Create volume.
- Expand volume.
- Rename volume.
- Delete volume using the volume ID.
- Move volumes between storage groups.
- Get volume details.

Create volume

The user can create a volume in the storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create volume
  dellemc_powermax_volume:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    vol_name: "{{vol_name}}"
    sg_name: "{{sg_name}}"
    size: 1
    cap_unit: "{{cap_unit}}"
    state: 'present'
```

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

Expand volume

The user can expand the size of the volumes by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Expanding volume size
  dellemc_powermax_volume:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    size: 3
    cap_unit: "{{cap_unit}}"
    vol_id: "0059B"
    state: 'present'
```

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

Rename volume

The user can rename the volume by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Renaming volume
  dellemc_powermax_volume:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    new_name: "Test_GOLD_vol_Renamed"
    vol_id: "0059B"
    state: 'present'
```

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

Delete volume

The user can delete the volume by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Delete volume
  dellemc_powermax_volume:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
```

```

user: "{{user}}"
password: "{{password}}"
serial_no: "{{serial_no}}"
vol_id: "0059B"
state: 'absent'

```

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

Move volumes between storage group

The user can move the volumes from one storage group to another by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```

- name: Move volume between storage group
  dellemc_powermax_volume:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    vol_name: "{{vol_name}}"
    sg_name: "{{sg_name}}"
    new_sg_name: "{{new_sg_name}}"
    state: 'present'

```

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

Volume module functions

The parameters of the Volume module are listed as follows:

Table 3 Parameters

Parameter	Choices/default	Comments
unispherehost		IP or FQDN of the Unisphere host. This parameter is mandatory.
universion		The version of the Unisphere software. The version is 90. This parameter is mandatory.
verifycert	<ul style="list-style-type: none"> Yes No 	<p>To validate the SSL certificate.</p> <ul style="list-style-type: none"> True - Verify the SSL certificate. False - Do not verify the SSL certificate. <p>This parameter is mandatory.</p>

Table 3 Parameters (continued)

Parameter	Choices/default	Comments
<code>user</code>		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
<code>password</code>		The password to access the Unisphere server. The password can be encrypted using Ansible vault. This parameter is mandatory.
<code>serial_no</code>		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
<code>vol_name</code>		The name of the volume.
<code>new_name</code>		The new identifier for the volume.
<code>sg_name</code>		The name of the current storage group.
<code>new_sg_name</code>		The name of the target storage group for moving volumes from one storage group to another.
<code>size</code>		The desired size of the volume. This parameter is mandatory when you create or expand the volume. Shrinking of volumes is not supported.
<code>cap_unit</code>	<ul style="list-style-type: none"> • MB • GB • TB The default unit is <i>GB</i> .	Volume capacity unit.
<code>vol_id</code>		The native id of the volume. This variable is required when you rename or delete the volume. This parameter is mandatory.

Table 3 Parameters (continued)

Parameter	Choices/default	Comments
state	<ul style="list-style-type: none"> absent present 	Defines whether the volume should exist in the storage group. This parameter is mandatory.

Host module

The host module manages the host within the PowerMax array.

The host module has the following functions:

- Create host with initiators and host flags.
- Add initiators to the host.
- Remove initiators from the host.
- Modify host flag values.
- Rename host.
- Delete host.

Create host

The user can create a host by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create host
  dellemc_powermax_host:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    host_name: "{{host_name}}"
    initiators:
      - 10000090fa7b4e85
    host_flags:
      spc2_protocol_version: true
      consistent_lun: true
      volume_set_addressing: 'unset'
      disable_q_reset_on_ua: false
      openvms: 'unset'
    state: 'present'
    initiator_state: 'present-in-host'
```

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

Get details of the host

The user can get details of the host by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
-name: Get host details
  dellemc_powermax_host:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    host_name: "{{host_name}}"
    state: 'present'
```

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

Add initiator to host

The user can add an initiator to the host by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Adding initiator to host
  dellemc_powermax_host:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    host_name: "{{host_name}}"
    initiators:
      - 10000090fa3d303e
    initiator_state: 'present-in-host'
    state: 'present'
```

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

Remove initiators from host

The user can remove initiators from the host by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Removing initiator from host
  dellemc_powermax_host:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
```



```

host_name: "{{host_name}}"
initiators:
- 10000090fa3d303e
initiator_state: 'absent-in-host'
state: 'present'

```

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

Modify host flags

The user can modify the host flags by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```

- name: Modify flags of host
  dellemc_powermax_host:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    host_name: "{{host_name}}"
    host_flags:
      spc2_protocol_version: unset
      consistent_lun: unset
      volume_set_addressing: true
      disable_q_reset_on_ua: false
      openvms: false
      avoid_reset_broadcast: true
    state: 'present'

```

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

Rename host

The user can rename the host by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```

- name: Rename host
  dellemc_powermax_host:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    host_name: "{{host_name}}"
    new_name: "{{new_host_name}}"
    state: 'present'

```

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

Delete host

The user can delete the host by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Delete host
  dellemc_powermax_host:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    host_name: "{{new_host_name}}"
    state: 'absent'
```

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

Host module functions

The parameters of the Host module are listed as follows:

Table 4 Parameters

Parameter	Choice/default	Comments
unispherehost		IP or FQDN of the Unisphere host. This parameter is mandatory.
universion		The version of the Unisphere software. The version is 90. This parameter is mandatory.
verifycert	<ul style="list-style-type: none"> Yes No 	To validate the SSL certificate. <ul style="list-style-type: none"> True - Verify the SSL certificate. False - Do not verify the SSL certificate. This parameter is mandatory.
user		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
password		The password to access the Unisphere server. The password can be encrypted using Ansible vault. This parameter is mandatory.

Table 4 Parameters (continued)

Parameter	Choice/default	Comments
<code>serial_no</code>		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
<code>host_name</code>		<p>The name of the host. The following conditions are applicable when naming the host:</p> <ul style="list-style-type: none"> Do not use special characters except "_." Case sensitive for RestAPI calls. <p>This parameter is mandatory.</p>
<code>new_name</code>		<p>The new name of the host when you rename the host. The following conditions must be met when you enter a new name:</p> <ul style="list-style-type: none"> Do not use special characters except "_." Case sensitive for RestAPI calls.
<code>initiators</code>		Lists the initiator <i>WWN</i> or <i>/QN</i> that needs to be added to or removed from the host.
<code>host_flags</code>	<ul style="list-style-type: none"> yes no unset <p>The default parameter is <i>unset</i>.</p>	<p>Enter as in <i>yaml</i> dictionary. The <code>host_flags</code> are optional. All the <code>host_flags</code> are listed below.</p> <ul style="list-style-type: none"> <code>volume_set_addressing</code> <code>disable_q_reset_on_ua</code> <code>environ_set</code> <code>avoid_reset_broadcast</code> <code>openvms</code> <code>scsi_3</code> <code>spc2_protocol_version</code> <code>scsi_support1</code>

Table 4 Parameters (continued)

Parameter	Choice/default	Comments
		<ul style="list-style-type: none"> consistent_lun
state	<ul style="list-style-type: none"> absent present 	<p>Defines whether the host must exist in the system.</p> <ul style="list-style-type: none"> "absent" indicates that the host must not exist in the system. "present" indicates that the host must exist in the system. <p>This parameter is mandatory.</p>
initiator_state	<ul style="list-style-type: none"> present-in-host absent-in-host 	<p>Defines whether the initiator must be available in the host.</p> <ul style="list-style-type: none"> "present-in-host" indicates that the initiator must be present in the host. "absent-in-host" indicates that the initiator must not be present in the host. <p>This parameter is mandatory for the following conditions:</p> <ul style="list-style-type: none"> Create a host with initiators. Add initiators to a host. Remove initiators from a host

Host group module

The host group module manages the host group within the PowerMax array. A host group is a container for multiple hosts and enables simple configuration of clusters.

The functions of the host group module are as follows:

- Create host group with hosts.
- Add hosts to a host group.
- Remove hosts from a host group.
- Rename host group.
- Modify the host flags of a host group.
- Delete host group.

Create host group

The user can create a host group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create host group
  dellemc_powermax_hostgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    hostgroup_name: "{{hostgroup_name}}"
    hosts:
      - Ansible_Testing_host
      state: 'present'
      host_state: 'present-in-group'
      host_flags:
        spc2_protocol_version: true
        consistent_lun: true
        disable_q_reset_on_ua: false
        openvms: 'unset'
```

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

Get details of host group

The user can get the details of the host group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Get host group details
  dellemc_powermax_hostgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    hostgroup_name: "{{hostgroup_name}}"
    state: 'present'
```

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

Add host to host group

The user can add host to a host group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Adding host to host group
  dellemc_powermax_hostgroup:
    unispherehost: "{{unispherehost}}"
```

```

universion: "{{universion}}"
verifycert: "{{verifycert}}"
user: "{{user}}"
password: "{{password}}"
serial_no: "{{serial_no}}"
hostgroup_name: "{{hostgroup_name}}"
hosts:
- Ansible_Testing_host2
state: 'present'
host_state: 'present-in-group'

```

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

Remove host from host group

The user can remove a host from the host group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```

- name: Removing host from host group
  dellemc_powermax_hostgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    hostgroup_name: "{{hostgroup_name}}"
    hosts:
    - Ansible_Testing_host2
    state: 'present'
    host_state: 'absent-in-group'

```

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

Modify host group flags

The user can modify the host flags by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```

- name: Modify flags of host group
  dellemc_powermax_hostgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    hostgroup_name: "{{hostgroup_name}}"
    host_flags:
      spc2_protocol_version: unset
      disable_q_reset_on_ua: false
      openvms: false
      avoid_reset_broadcast: true
    state: 'present'

```

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

Rename host group

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

The syntax of the playbook is shown as follows:

```
- name: Rename host group
  dellemc_powermax_hostgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    hostgroup_name: "{{hostgroup_name}}"
    new_name: "Ansible_Testing_hostgroup2"
    state: 'present'
```

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

Delete host group

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

The syntax of the playbook is shown as follows:

```
- name: Delete host group
  dellemc_powermax_hostgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    hostgroup_name: "Ansible_Testing_hostgroup2"
    state: 'absent'
```

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

Host Group module functions

The parameters of the Host Group module are listed as follows with an example:

Table 5 Parameters

Parameter	Choice/default	Comments
unispherehost		IP or FQDN of the Unisphere host. This parameter is mandatory.

Table 5 Parameters (continued)

Parameter	Choice/default	Comments
<code>universion</code>		The version of the Unisphere software. The version is 90. This parameter is mandatory.
<code>verifycert</code>	<ul style="list-style-type: none"> • Yes • No 	<p>To validate the SSL certificate.</p> <ul style="list-style-type: none"> • True - Verify the SSL certificate. • False - Do not verify the SSL certificate. <p>This parameter is mandatory.</p>
<code>user</code>		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
<code>password</code>		The password to access the Unisphere server. The password can be encrypted using Ansible vault. This parameter is mandatory.
<code>serial_no</code>		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
<code>host_group_name</code>		<p>The name of the host group. The following conditions must be met when you enter the name:</p> <ul style="list-style-type: none"> • Do not use special characters except "_." • Case sensitive for RestAPI calls. <p>This parameter is mandatory.</p>
<code>new_name</code>		The new name of the host group when you rename the host group. The following conditions must be met when you enter a new name:

Table 5 Parameters (continued)

Parameter	Choice/default	Comments
		<ul style="list-style-type: none"> Do not use special characters except "_." Case sensitive for RestAPI calls.
<code>hosts</code>		Lists of the host names that are added to the host group or removed from host group. You can create an empty host group.
<code>host_state</code>	<ul style="list-style-type: none"> <code>present-in-group</code> <code>absent-in-group</code> 	<p>Defines whether the host must be available in the host group.</p> <ul style="list-style-type: none"> "present-in-group" indicates that the host must be present in the host group. "absent-in-group" indicates that the host must not be present in the host group.
<code>host_flags</code>	<ul style="list-style-type: none"> <code>yes</code> <code>no</code> <code>unset</code> (default) 	<p>Enter as in <i>yaml</i> dictionary. All the <code>host_flags</code> are listed below.</p> <ul style="list-style-type: none"> <code>volume_set_addressing</code> <code>disable_q_reset_on_ua</code> <code>environ_set</code> <code>avoid_reset_broadcast</code> <code>openvms</code> <code>scsi_3</code> <code>spc2_protocol_version</code> <code>scsi_support1</code> <code>consistent_lun</code>
<code>state</code>	<ul style="list-style-type: none"> <code>absent</code> <code>present</code> 	<p>Defines whether the host group must be present in the system.</p> <ul style="list-style-type: none"> "absent" - The host must not be present in the system. "present" - The host must be present in the system.

Table 5 Parameters (continued)

Parameter	Choice/default	Comments
		This parameter is mandatory.

Port module

The ports available on the PowerMax array are managed by the Port module. The Port module lists the details of single or multiple ports.

Get details of single/multiple ports

The user can get the details of single or multiple ports by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Get details of single/multiple ports
  dellemc_powermax_port:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{array_id}}"
    ports:
      - director_id: "FA-1D"
        port_id: "5"
      - director_id: "SE-1F"
        port_id: "29"
```

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

Port module functions

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

Table 6 Parameters

Parameter	Choice/default	Comments
unispherehost		IP or FQDN of the Unisphere host. This parameter is mandatory.
universion		The version of the Unisphere software. The version is 90. This parameter is mandatory.
verifycert	<ul style="list-style-type: none"> Yes No 	To validate the SSL certificate. <ul style="list-style-type: none"> True - Verify the SSL certificate.

Table 6 Parameters (continued)

Parameter	Choice/default	Comments
		<ul style="list-style-type: none"> False - Do not verify the SSL certificate. This parameter is mandatory.
<code>user</code>		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
<code>password</code>		The password to access the Unisphere server. The password can be encrypted using Ansible vault. This parameter is mandatory.
<code>serial_no</code>		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
<code>ports</code>		Lists the port director and the port id.

Port Group module

The Port Group module manages the port group in the PowerMax array.

The port group management module has the following functions:

- Create a port group with ports.
- Create empty port group.
- Add ports to a port group.
- Remove ports from a port group.
- Rename a port group.
- Remove a port group.

Create port group without ports

The user can create a port group without ports by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create port group without ports
```

```
dellemc_powermax_portgroup:
  unispherehost: "{{unispherehost}}"
  universion: "{{universion}}"
  verifycert: "{{verifycert}}"
  user: "{{user}}"
  password: "{{password}}"
  serial_no: "{{array_id}}"
  portgroup_name: "{{portgroup_name}}"
  state: "present"
```

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

Create port group with ports

The user can create a port group with ports by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create port group with ports
  dellemc_powermax_portgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{array_id}}"
    portgroup_name: "{{portgroup_name}}"
    state: "present"
    ports:
      - director_id: "FA-1D"
        port_id: "5"
      - director_id: "FA-2D"
        port_id: "5"
    port_state: "present-in-group"
```

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

Add ports to port group

The user can add ports to a port group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Add ports to port group
  dellemc_powermax_portgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{array_id}}"
    portgroup_name: "{{portgroup_name}}"
    state: "present"
    ports:
      - director_id: "FA-2D"
        port_id: "8"
      - director_id: "FA-2D"
```

```
port_id: "9"
port_state: "present-in-group"
```

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

Remove ports from port group

The user can remove ports from the port group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Remove ports from port group
  dellemc_powermax_portgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{array_id}}"
    portgroup_name: "{{portgroup_name}}"
    state: "present"
    ports:
      - director_id: "FA-2D"
        port_id: "8"
      - director_id: "FA-2D"
        port_id: "9"
    port_state: "absent-in-group"
```

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

Rename port group

The user can rename the port group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Modify port group
  dellemc_powermax_portgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{array_id}}"
    portgroup_name: "{{portgroup_name}}"
    state: "present"
    new_name: "{{new_name}}"
```

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

Delete port group

The user can delete a port group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Delete port group
  dellemc_powermax_portgroup:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{array_id}}"
    portgroup_name: "{{portgroup_name}}"
    state: "absent"
```

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

Port Group module functions

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

Table 7 Parameters

Parameter	Choice/default	Comments
unispherehost		IP or FQDN of the Unisphere host. This parameter is mandatory.
universion		The version of the Unisphere software. The version is 90. This parameter is mandatory.
verifycert	<ul style="list-style-type: none"> Yes No 	To validate the SSL certificate. <ul style="list-style-type: none"> True - Verify the SSL certificate. False - Do not verify the SSL certificate. This parameter is mandatory.
user		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
password		The password to access the Unisphere server. The password can be encrypted using Ansible vault. This parameter is mandatory.

Table 7 Parameters (continued)

Parameter	Choice/default	Comments
<code>serial_no</code>		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
<code>portgroup_name</code>		<p>The name of the port group. The following conditions must be met when you enter the name:</p> <ul style="list-style-type: none"> Do not use special characters except "_." Case sensitive for RestAPI calls. <p>This parameter is mandatory.</p>
<code>state</code>	<ul style="list-style-type: none"> absent present 	<p>Defines whether the port group must be present in the system.</p> <ul style="list-style-type: none"> "absent" - The port group must not be present in the system. "present" - The port group must be present in the system. <p>This parameter is mandatory.</p>
<code>ports</code>		Lists the port director and ports that are added to the port group or removed from port group.
<code>port_state</code>	<ul style="list-style-type: none"> present-in-group absent-in-group 	<p>Defines whether the port must be available in the port group.</p> <ul style="list-style-type: none"> "present-in-group" indicates that the port must be present in the port group. "absent-in-group" indicates that the port must not be present in the port group.

Table 7 Parameters (continued)

Parameter	Choice/default	Comments
<code>new_name</code>		<p>The new name of the port group when you rename the port group. The following conditions must be met when you enter a new name:</p> <ul style="list-style-type: none"> Do not use special characters except "_." Case sensitive for RestAPI calls.

Masking View module

The Masking View module manages the masking views on the PowerMax array.

The masking view module has the following functions:

- Create masking view with a port group, storage group, host, or host group.
- Modify masking view.
- Delete masking view.

For creating a masking view, the port groups, storage groups, hosts, and host groups must be present on the array. Once the masking view is created, only the name of the masking view can be changed. The entities such as port group, storage group, hosts, or host group cannot be changed. The masking view can be created either for a host or a host group, but not for both. The host name or a host group name must be provided to create masking view.

Create masking view with host group

The user can create a masking view with a host group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create MV with hostgroup
  dellemc_powermax_maskingview:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    mv_name: "{{mv_name}}"
    portgroup_name: "Ansible_Testing_portgroup"
    hostgroup_name: "Ansible_Testing_hostgroup"
    sg_name: "Ansible_Testing_SG"
    state: "present"
```

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

Create masking view with host

The user can create a masking view with host by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create MV with host
  dellemc_powermax_maskingview:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    mv_name: "{{mv_name}}"
    portgroup_name: "Ansible_Testing_portgroup"
    host_name: "Ansible_Testing_host"
    sg_name: "Ansible_Testing_SG"
    state: "present"
```

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

Rename masking view

The user can rename the masking view by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Rename host masking view
  dellemc_powermax_maskingview:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    mv_name: "{{mv_name}}"
    new_mv_name: "Ansible_Testing_mv_renamed"
    state: "present"
```

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

Delete masking view

The user can delete the masking view by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Delete host masking view
  dellemc_powermax_maskingview:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
```

```
serial_no: "{{serial_no}}"
mv_name: "Ansible_Testing_mv_renamed"
state: "absent"
```

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

Masking View module functions

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

Table 8 Parameters

Parameter	Choice/default	Comments
unisphrehost		IP or FQDN of the Unisphere host. This parameter is mandatory.
universion		The version of the Unisphere software. The version is 90. This parameter is mandatory.
verifycert	<ul style="list-style-type: none"> Yes No 	<p>To validate the SSL certificate.</p> <ul style="list-style-type: none"> True - Verify the SSL certificate. False - Do not verify the SSL certificate. <p>This parameter is mandatory.</p>
user		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
password		The password to access the Unisphere server. The password can be encrypted using Ansible vault. This parameter is mandatory.
serial_no		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
mv_name		The name of the masking view. The following conditions

Table 8 Parameters (continued)

Parameter	Choice/default	Comments
		<p>must be met when you enter the name:</p> <ul style="list-style-type: none"> Do not use special characters except "_." Case sensitive for RestAPI calls. <p>This parameter is mandatory.</p>
<code>new_mv_name</code>		<p>The new name of the masking view when you rename the existing masking view. The following conditions must be met when you provide the new name for masking view:</p> <ul style="list-style-type: none"> Do not use special characters except "_." Case sensitive for REST API calls.
<code>portgroup_name</code>		The name of the existing port group.
<code>hostgroup_name</code>		The name of the existing host group. Use this parameter to create cluster export.
<code>host_name</code>		The name of the existing host. Use this parameter to create an exclusive host or to export to a host.
<code>sg_name</code>		The name of the existing storage group.
<code>state</code>	<ul style="list-style-type: none"> absent present 	Defines whether the masking view must be present on the system. This parameter is mandatory.

Snapshot module

The Snapshot module manages the snapshots available on the PowerMax array.

The snapshot module has the following functions:

- Create a storage group snapshot.
- Get details of a storage group snapshot.
- Rename a storage group snapshot.
- Change storage group snapshot link status.
- Delete storage group snapshot.

Create snapshot for a storage group

The user can create a snapshot for a storage group by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Create a Snapshot for a Storage Group
  dellemc_powermax_snapshot:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "ansible_sg"
    snapshot_name: "ansible_sg_snap"
    ttl: "2"
    ttl_unit: "days"
    state: "present"
```

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

Get details of storage group snapshot

The user can get details of a storage group snapshot by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Get Storage Group Snapshot details
  dellemc_powermax_snapshot:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "ansible_sg"
    snapshot_name: "ansible_sg_snap"
    state: "present"
```

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

Delete storage group snapshot

The user can delete a storage group snapshot by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Delete Storage Group Snapshot
  dellemc_powermax_snapshot:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
```

```

verifycert: "{{verifycert}}"
user: "{{user}}"
password: "{{password}}"
serial_no: "{{serial_no}}"
sg_name: "ansible_sg"
snapshot_name: "ansible_sg_snap"
generation: 1
state: "absent"

```

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

Rename storage group snapshot

The user can rename the storage group snapshot by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```

- name: Rename Storage Group Snapshot
  dell EMC PowerMax Snapshot:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "ansible_sg"
    snapshot_name: "ansible_sg_snap"
    new_snapshot_name: "ansible_snap_new"
    generation: 0
    state: "present"

```

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

Change snapshot link status to linked

The user can change the link status of the snapshot to *linked* by running the appropriate playbook

The syntax of the playbook is shown as follows:

```

- name: Change Snapshot Link Status to Linked
  dell EMC PowerMax Snapshot:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "ansible_sg"
    snapshot_name: "ansible_snap_new"
    generation: 1
    target_sg_name: "ansible_sg_target"
    link_status: "linked"
    state: "present"

```

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

Change snapshot link status to unlinked

The user can change the link status of the snapshot to *unlinked* by running the appropriate playbook.

The syntax of the playbook is shown as follows:

```
- name: Change Snapshot Link Status to UnLinked
  dellemc_powermax_snapshot:
    unispherehost: "{{unispherehost}}"
    universion: "{{universion}}"
    verifycert: "{{verifycert}}"
    user: "{{user}}"
    password: "{{password}}"
    serial_no: "{{serial_no}}"
    sg_name: "ansible_sg"
    snapshot_name: "ansible_snap_new"
    generation: 1
    target_sg_name: "ansible_sg_target"
    link_status: "unlinked"
    state: "present"
```

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

Snapshot module functions

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

Table 9 Parameters

Parameter	Choice/default	Comments
unispherehost		IP or FQDN of the Unisphere host. This parameter is mandatory.
universion		The version of the Unisphere software. The version is 90. This parameter is mandatory.
verifycert	<ul style="list-style-type: none"> Yes No 	To validate the SSL certificate. <ul style="list-style-type: none"> True - Verify the SSL certificate. False - Do not verify the SSL certificate. This parameter is mandatory.
user		The user name to access the Unisphere server. The user name can be encrypted using Ansible vault. This parameter is mandatory.
password		The password to access the Unisphere server. The

Table 9 Parameters (continued)

Parameter	Choice/default	Comments
		password can be encrypted using Ansible vault. This parameter is mandatory.
serial_no		The serial number of the PowerMax array. It is a required parameter for getting information of specific entities. If this parameter is absent, the script lists all the serial numbers of the registered array on the specified Unisphere host.
sg_name		The name of the storage group. This parameter is mandatory.
snapshot_name		The name of the snapshot. This parameter is mandatory.
new_snapshot_name		The new name of the snapshot.
generation		The generation number of the Snapshot. Generation is mandatory for link, unlink, rename, and delete operations. Optional for <i>Get snapshot details</i> operation. Create snapshot creates a snapshot with generation number 0. Rename is supported only for generation number 0.
target_sg_name		The target storage group name.
link_status	<ul style="list-style-type: none"> linked unlinked 	Defines the link status of the snapshot.
ttl		The Time To Live (TTL) value for the Snapshot. If the TTL value is not specified, the storage group snapshot details would be returned. However, to create a storage group snapshot, the TTL must be specified. If the storage group snapshot does not have a TTL value, specify the TTL value as "None."

Table 9 Parameters (continued)

Parameter	Choice/default	Comments
<code>t1l_unit</code>	<ul style="list-style-type: none">• hours• days	The unit for TTL. If no <code>t1l_unit</code> is specified, 'days' is taken as default value.
<code>state</code>	<ul style="list-style-type: none">• absent• present	Defines whether the snapshot must exist on the storage array. This parameter is mandatory.

CHAPTER 4

Troubleshooting

This chapter presents the following topics:

- [Troubleshooting Ansible modules for Dell EMC PowerMax](#)..... 50

Troubleshooting Ansible modules for Dell EMC PowerMax

This section describes the known problems and limitations for Ansible modules for Dell EMC PowerMax

Table 10 Troubleshooting

Symptoms	Prevention, resolution, or workaround
Error logs.	See the log file <i>dell_ansible_provisioning.log</i> for detailed error description.
Verbose mode.	Use <code>-vvv</code> to get more details about the playbook.
Ansible module entities.	The user can verify the entities created with Ansible modules in Unisphere UI.