Ansible Modules for Dell EMC Unity Product Guide

1.0

June 2020
Rev. 01

DELL EMC

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.

Contents

1

Introduction

This chapter introduces the following topics:

Topics:

Product overview

The Ansible modules for Dell EMC Unity are used to automate and orchestrate the deployment, configuration, and management of Dell EMC Unity Family systems including Unity, Unity XT, and the UnityVSA. The capabilities of Ansible modules are managing volumes (LUNs), consistency groups, storage pools, hosts, snapshots, and snapshot schedules, and obtaining Unity system information. 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 for repetitive tasks, and hence supports fault tolerance.

List of Dell EMC Unity Ansible modules

The following are the list of modules:

- Gather Facts Module
- Volume Module (LUNs)
- Consistency Group Module
- Storage Pool Module
- Host Module
- Snapshot Module
- Snapshot Schedules Module

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

Table 1. Common Unity parameters

Name	Type	Description	Requirement	Choices/Default
`username`	str	Username for Unity Unisphere.	Mandatory	
`password`	str	Password for Unity Unisphere.	Mandatory	
`port`	int	Port at which the Unity Unisphere REST API is hosted. This is the same as the default port for the Unity system for CLI.	Optional	Default: 443
`unispherehost`	str	IP address or FQDN for Unity.	Mandatory	
`verifycert`	bool	Specifies whether or not to verify the SSL certificate for Unity Ansible commands.	Optional	Choices: • *True* (default)- Indicates that the SSL certificate should be verified. • *False* - indicates that the SSL certificate

Table 1. Common Unity parameters(continued)

Name	Type	Description	Requirement	Choices/Default
				should not be verified.

2

Installation

This chapter introduces the following topics:

Topics:

Software prerequisites

This table provides information about the software prerequisites for the Ansible modules for Dell EMC Unity.

Prerequisites

Table 2. Software prerequisites

Ansible Modules	Unity Operating Environment	Python version	Python Library version	Ansible	Red Hat Enterprise Linux
v1.0	5.0.0 or later	Python 2.7.12 or later	Python library for Unity storops 1.2.5 or later	2.7 or later	7.5, 7.6

NOTE: Python library for Unity `storops` can be downloaded from https://github.com/emc-openstack/storops/releases

Install the SDK

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

About this task

Complete the following tasks before you run playbooks on Ansible modules for Dell EMC Unity:

Steps

1. Install *pip*, if it is not present on the Ansible host.
   Run the following command to install:

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

   For more information, refer to the Python documentation.
2. Install the Python SDK named *storops*. It can be installed using pip, based on the appropriate Python version.

Install the modules

Install the specific Ansible for Dell EMC Unity modules.

Prerequisites

After installing the SDK, install the modules.

About this task

Copy the Ansible modules to the appropriate locations in the virtual machine.

Steps

1. 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`
2. Copy *dellemc_ansible_utils.py* from the `/utils` folder to one of the following locations:
   - For Python 2.7 `/usr/lib/python2.7/site-packages/ansible/module_utils/storage/dell`
   - For Python 3.5 `/usr/lib/python3.5/site-packages/ansible/module_utils/storage/dell`
3. Copy *utils/__init__.py* to one of the following:
   - For Python 2.7 `/usr/lib/python2.7/site-packages/ansible/module_utils/storage/dell`
   - For Python 3.5 `/usr/lib/python3.5/site-packages/ansible/module_utils/storage/dell`
4. Copy all the module Python files from the `unity/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/`
5. Copy the *doc_fragments/dellemc_unity.py* from `/doc_fragments` folder 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 the SSL Certificate

Learn how to install the SSL certificate through a certified Certificate Authority.

Prerequisites

These modules are supported through Certificate Authority (CA)-certified certificates only. Self-signed certificates are not supported.

Steps

1. Copy the CA certificate to the `/usr/local/share/ca-certificates` path of the host.
2. Import the SSL certificate to host using the command `sudo update-ca-certificates`.
3. Set the *REQUESTS_CA_BUNDLE* environment variable to the path of the SSL certificate using the command `export REQUESTS_CA_BUNDLE=/usr/local/share/ca-certificates/<<Certificate_Name>>`.

3

Modules

This chapter covers the following modules:

Topics:

Gather Facts Module

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

The Gather Facts module returns information about storage resources and entities such as a list of volumes (LUNs), consistency groups, pools, hosts, initiators, snapshot schedules, and so on, as well as additional details of the Unity system.

This module supports the following functions:

- List of volumes
- List of storage pools
- List of consistency groups
- List of fibre channel (FC) initiators
- List of iSCSI initiators
- List of hosts
- List of snapshot schedules

Table 3. Gather Facts module parameters

Name	Type	Description	Requirement	Choices/Defaults
`gather_subset`	list	List of string variables to specify the storage system entities for which information is required. • *vol* - volumes (LUNs) • *storage_pool* - pools • *cg* - consistency groups • *fc_initiator* - FC initiators • *iscsi_initiator* - iSCSI initiators • *host* - hosts • *snapshot_schedule* - snapshot schedules	Optional	Choices: • *vol* • *storage_pool* • *cg* • *fc_initiator* • *iscsi_initiator* • *host* • *snapshot_schedule*

Get system information

Learn how to get system information for a Unity storage system.

You can get general Unity system information by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get the details of Unity array.
  dellemc_unity_gatherfacts:
     unispherehost: "{{unispherehost}}"
     username: "{{username}}"
     password: "{{password}}"
     verifycert: "{{verifycert}}"
```

Get list of storage pools

Learn how to get the list of storage pools for a Unity storage system.

You can get the list of pools by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of Storage Pools on Unity array.
  dellemc_unity_gatherfacts:
     unispherehost: "{{unispherehost}}"
     username: "{{username}}"
     password: "{{password}}"
     verifycert: "{{verifycert}}"
     gather_subset:
       - storage_pool
```

Get list of volumes

Learn how to get the list of volumes (LUNs) for a Unity storage system.

You can get the list of volumes by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of volumes on Unity array
  dellemc_unity_gatherfacts:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    gather_subset:
      - vol
```

Get list of consistency groups

Learn how to get the list of consistency groups for a Unity storage system.

You can get the list of consistency groups by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of Consistency Groups on Unity array.
  dellemc_unity_gatherfacts:
     unispherehost: "{{unispherehost}}"
     username: "{{username}}"
     password: "{{password}}"
     verifycert: "{{verifycert}}"
     gather_subset:
       - cg
```

Get list of FC initiators

Learn how to get a list of FC initiators for a Unity storage system.

You can get the list of FC initiators by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of FC initiators on Unity array.
  dellemc_unity_gatherfacts:
     unispherehost: "{{unispherehost}}"
     username: "{{username}}"
     password: "{{password}}"
     verifycert: "{{verifycert}}"
     gather_subset:
       - fc_initiator
```

Get list of iSCSI initiators

Learn how to get a list of iSCSI initiators for a Unity storage system.

You can get the list of iSCSI initiators by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of ISCSI initiators on Unity array.
  dellemc_unity_gatherfacts:
     unispherehost: "{{unispherehost}}"
     username: "{{username}}"
     password: "{{password}}"
     verifycert: "{{verifycert}}"
     gather_subset:
       - iscsi_initiator
```

Get list of hosts

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

You can get the list of hosts by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of Unity hosts
  dellemc_unity_gatherfacts:
     unispherehost: "{{unispherehost}}"
     username: "{{username}}"
     password: "{{password}}"
     verifycert: "{{verifycert}}"
     gather_subset:
       - host
```

Get list of snapshot schedules

Learn how to get the list of snapshot schedule details for a Unity storage system.

You can get the list of snapshot schedules by running the following playbook.

The syntax of the playbook is as follows:

```
- name: Get list of Snapshot Schedules on Unity array.
  dellemc_unity_gatherfacts:
     unispherehost: "{{unispherehost}}"
     username: "{{username}}"
```

```
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    gather_subset:
      - snapshot_schedule
```

Volume module

Learn about the Volume Module and the supported functions.

The Volume Module is intended to manage the volumes (LUNs) on Unity storage systems. It supports the following functions:

- Create volume
  - In a pool
  - Assign an I/O limit policy
  - Assign a tiering policy
  - Assign a snap schedule
  - Map to a host
- Modify Volume
  - Expand a volume by name or ID
  - Modify volume details including description, compression, and default SP
  - Modify host by volume name or ID
  - Modify host mapping by host name or host ID
- Delete volume
  - Delete a volume by volume name or ID

The parameters of volume module are as follows:

Table 4. Volume Module Parameters

Name	Type	Description	Requirement	Choices/Default
`vol_name`	str	Unique name of the volume (LUN). It can contain a maximum of 63 alphanumeric or printable characters, or fewer printable unicode characters. All other actions for a volume are supported using either volume name or volume ID.	Mandatory for create action only	
`vol_id`	str	Unique, system-assigned ID of the volume automatically generated when a volume is created.	Optional Cannot be used while creating a volume. All other actions are supported using either volume name or volume ID.	
`description`	str	Description of the volume.	Optional	
`pool_name`	str	Name of the pool for the volume.	Mandatory for create action only (unless `pool_id` is used)	
`pool_id`	str	System-assigned ID of the pool for the volume.	Mandatory for create action only (unless `pool_name` is used)	
`size`	int	Size of the volume.	Mandatory for create and expand actions	Size range *1GB-256TB*. Size must be a multiple of 8192.
`cap_unit`	str	Volume size unit of measurement.	Optional	Choices: • *GB* (default)

Table 4. Volume Module Parameters(continued)

Name	Type	Description	Requirement	Choices/Default
				• *TB*
is_thin	bool	Whether or not the volume is a thin LUN.	Optional	Choices: • *true* (default) • *false*
compression	bool	Whether or not data reduction is enabled.	Optional	Choices: • *true* • *false* (default)
sp	str	Default storage processor (SP) for the volume.	Optional	System default SP.
io_limit_policy	str	I/O limit policy for the volume which prioritizes I/O operations for specific hosts.	Optional. Inherited from the system; not applicable to the create action.	No limit.
tiering_policy	str	Tiering policy for how the volume data will be distributed among the tiers available in the pool.	Optional	Choices: • *AUTOTIER_HIGH* • *AUTOTIER* • *HIGHEST* • *LOWEST*
snap_schedule	str	Name of the snapshot schedule for the volume.	Optional.	
host_name	str	Name of the host.	Optional	
host_id	str	System-assigned host ID of the host.	Optional	
hlu	int	Logical unit number for the host volume access. If not specified, the system will automatically assign an HLU number. Optional parameter when mapping a volume to host. HLU modification is not supported. The maximum value support is 255.	Optional	
mapping_state	str	Define whether the volume should be mapped to a host. Only one host can be supplied at a time in a single command.	Optional	Choices: • *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.
state	str	Whether the volume should exist or not on the system.	Mandatory	Choices: • *present* - indicates that the volume should exist on the system, such as in the create action. • *absent* - indicates that the volume should not exist on

Table 4. Volume Module Parameters(continued)

Name	Type	Description	Requirement	Choices/Default
				the system, such as in the delete action.
new_vol_name	str	New name of the volume to be applied during the name rename operations.	Optional	

Create volumes

Learn how to create volumes for the Unity storage system.

You can create volumes by running this playbook. The syntax is as follows:

```
- name: Create Volume
  dellemc_unity_volume:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_name: "{{vol_name}}"
    description: "{{description}}"
    pool_name: "{{pool}}"
    size: 2
    cap_unit: "{{cap_GB}}"
    io_limit_policy: "{{io_limit_policy}}"
    host_name: "{{host_name}}"
    mapping_state: "{{state_mapped}}"
    state: "{{state_present}}"
```

NOTE: To create a new volume, `vol_name` and `size` are required.

Get the details for a volume

Learn how to get the details for a volume on a Unity storage system.

You can obtain the details of a volume on the storage system by running this playbook. The syntax is as follows:

Get details of a volume by volume name

```
- name: Get the details of Volume by volume name
  dellemc_unity_volume:
      unispherehost: "{{unispherehost}}"
      username: "{{username}}"
      password: "{{password}}"
      verifycert: "{{verifycert}}"
      vol_name: "{{vol_name}}"
      state: "{{state_present}}"
```

Get details of a volume by volume ID

```
- name: Get details of volume using id
  dellemc_unity_volume:
      unispherehost: "{{unispherehost}}"
      username: "{{username}}"
      password: "{{password}}"
      verifycert: "{{verifycert}}"
```

```
      vol_id: "{{vol_id}}"
      state: "present"
```

Modify volume attributes

Learn how to modify volume attributes for Unity storage systems, such as name, size, hosts, I/O limit policy, host name, and so on.

You can modify volume size, name, description, and other attributes by running this playbook. The syntax of the playbook is as follows:

```
- name: Modify Volume
  dellemc_unity_volume:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_name: "{{vol_name}}"
    description: "{{description_new}}"
    pool_name: "{{pool}}"
    size: 2
    cap_unit: "{{cap_GB}}"
    io_limit_policy: "{{io_limit_policy}}"
    host_name: "{{host_name}}"
    mapping_state: "{{state_mapped}}"
    state: "{{state_present}}"
```

NOTE: **`size`** **is a required parameter for expanding the volume size.**

Expand volume size

Learn how to expand the volume size for Unity storage systems.

You can expand the size of a volume by running the following playbook. The syntax is as follows:

Expand the volume by volume name

```
- name: Expand Volume
  dellemc_unity_volume:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_name: "{{vol_name}}"
    size: 5
    cap_unit: "{{cap_GB}}"
    state: "{{state_present}}"
```

Expand the volume by volume ID

```
- name: Expand Volume by volume id
  dellemc_unity_volume:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_id:"{{result.volume_details.id}}"
    size: 5
    cap_unit: "{{cap_GB}}"
    state: "{{state_present}}"
```

Modify volume host mapping

Learn how to modify volume hosts for Unity storage systems.

You can modify the host-volume mapping by running the following playbook. The syntax is as follows:

Modify volume-to-host mapping by host name

```
- name: Modify Volume host mapping by host name
  dellemc_unity_volume:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_name: "{{vol_name}}"
    host_name: "{{new_host_name}}"
    mapping_state: "{{state_mapped}}"
    state: "{{state_present}}"
```

Modify volume-to-host mapping by host ID

```
- name: Modify Volume host mapping by host id
  dellemc_unity_volume:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_name: "{{vol_name}}"
    host_id: "{{new_host_name}}"
    mapping_state: "{{state_mapped}}"
    state: "{{state_present}}"
```

Delete volume

Learn how to delete volumes for Unity storage systems.

You can delete volumes by running the following playbook. The syntax is as follows:

Delete a volume using the volume name

```
- name: Delete Volume by vol name
  dellemc_unity_volume:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_name: "{{vol_name}}"
    state: "{{state_absent}}"
```

Delete a volume using the volume ID

```
- name: Delete Volume by vol id
  dellemc_unity_volume:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_id: "{{vol_id}}"
    state: "{{state_absent}}"
```

NOTE:

- **This action does not delete the associated snapshots for the volume.**

- **A volume that is attached to a host or is part of a consistency group cannot be deleted.**

Storage Pools module

Learn about the Storage Pools module and its supported functions.

The Storage Pools module allows you to manage pools for your Unity storage systems. It supports the following functions:

- Get storage pool details using a pool name or pool ID
- Modify storage pool attributes using the pool name or pool ID

Table 5. Storage Pool Module Parameters

Name	Type	Description	Requirement	Choices/Default
pool_name	str	Unique pool name in the storage system.	Optional	
pool_id	str	Unique, system-assigned identifier of the pool.	Optional	
new_pool_name	str	New unique name of the storage pool in the rename operation.	Optional	
pool_description	str	Description of the storage pool.	Optional	
fast_cache	str	Indicates whether the FAST Cache is enabled for the pool.	Optional	Choices: • *enabled* • *disabled*
fast_vp	str	Indicates whether to enable scheduled data relocations (FAST VP) for the pool.	Optional	Choices: • *enabled* • *disabled*
state	str	State of the storage pool.	Mandatory	Choices: • *present* - indicates that the pool should exist on the system. • *absent* - indicates that the pool should not exist on the system.

Get the details for a storage pool

Learn how to get the details for a Unity storage pool.

You can obtain details for a pool by running this playbook. The syntax is as follows:

Get storage pool details using the pool name

```
- name: Get the details of Storage pool by name
  dellemc_unity_storagepool:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    pool_name: "{{pool_name}}"
    state: "present"
```

Get storage pool details using the pool ID

```
- name: Get the details of Storage pool by pool id
  dellemc_unity_storagepool:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    pool_id: "{{pool_id}}"
    state: "present"
```

Modify storage pool attributes

Learn how to modify the attributes for a Unity pool, such as pool name, description, FAST cache status, and FAST VP status.

You can modify the attributes of a pool by running the following playbook. The syntax is as follows:

Modify pool attributes using the pool name

```
- name: Modify Storage pool attributes using pool_name
  dellemc_unity_storagepool:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    pool_name: "{{pool_name}}"
    new_pool_name: "{{new_pool_name}}"
    pool_description: "{{pool_description}}"
    fast_cache: "{{fast_cache_enabled}}"
    fast_vp: "{{fast_vp_enabled}}"
    state: "present"
```

Modify pool attributes using the pool ID

```
- name: Modify Storage pool attributes using pool_id
  dellemc_unity_storagepool:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    pool_id: "{{pool_id}}"
    new_pool_name: "{{new_pool_name}}"
    pool_description: "{{pool_description}}"
    fast_cache: "{{fast_cache_enabled}}"
    fast_vp: "{{fast_vp_enabled}}"
    state: "present"
```

Consistency Group module

Learn about the consistency group (LUN group) module and its supported functions.

The consistency group (CG) module allows you to manage consistency groups on your Unity storage system. It supports the following functions:

- Create a consistency group
- Get details for a consistency group by consistency group name or ID
- Add volumes to a consistency group
- Modify consistency group details
- Rename a consistency group
- Remove volumes (LUNs) from a consistency group

- Delete a consistency group

Table 6. Consistency group module parameters

Name	Type	Description	Requirement	Choices/Default
cg_name	str	Unique name of the consistency group on the storage system.	Optional	
cg_id	str	Unique ID of the consistency group on the storage system.	Optional	
description	str	Description of the consistency group.	Optional	
volumes	list	List of volumes to be added to or removed from the consistency group.	Optional	
snap_schedule	str	Snapshot schedule assigned to the consistency group.	Optional	
new_cg_name	str	New name for the consistency group for the rename operation.	Optional	
vol_state	str	State of the volume in a consistency group. Indicates whether the volume is part of the CG.	Optional	Choices: • *present-in-group*--exists in the consistency group. • *absent-in-group*is not included in the consistency group.
state	str	State of the consistency group.	Mandatory	Choices: • *present* - indicates that the CG should exist on the system. • *absent* - indicates that the CG should not exist on the system.
tiering_policy	str	Tiering policy choices for how the CG data will be distributed among the tiers available in the pool.	Optional	Choices: • *AUTOTIER_HIGH* • *AUTOTIER* • *HIGHEST* • *LOWEST*

Create consistency groups

Learn how to create consistency groups (LUN groups) on the Unity storage system.

You can create consistency groups on a Unity system by running this playbook. They syntax is as follows:

```
- name: Create consistency group with existing volumes
  dellemc_unity_consistencygroup:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    cg_name: "{{cg_name}}"
    description: "{{description}}"
    tiering_policy: "{{policy1}}"
    volumes:
     - vol_id: "sv_21540"
     - vol_name: "Ansible_vol_Test"
    vol_state: "present-in-group"
```

```
    snap_schedule: "Ansible_Test"
    state: "present"
```

Get the details for a consistency group

Learn how to obtain details for a consistency group on a Unity storage system.

You can obtain details for a consistency group by running this playbook. The syntax is as follows:

Get details of a consistency group by CG name

```
- name: Get details of consistency group
  dellemc_unity_consistencygroup:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    cg_name: "{{cg_name}}"
    tate: "present"
```

Get details of a consistency group by CG ID

```
- name: Get details of consistency group
  dellemc_unity_consistencygroup:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    cg_id: "{{cg_id}}"
    state: "present"
```

Rename a consistency group

Learn how to rename an existing consistency group on the Unity system.

You can rename a consistency group by running this playbook. The syntax is as follows:

```
- name: Rename consistency group
  dellemc_unity_consistencygroup:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    cg_name: "{{cg_name}}"
    new_cg_name: "{{new_cg_name}}"
    state: "present"
```

Add volumes (LUNs) to a consistency group

Learn how to add volumes to an existing consistency group.

You can add volumes to an existing consistency group by running this playbook. The syntax is as follows:

```
  - name: Add volumes to consistency group
    dellemc_unity_consistencygroup:
      unispherehost: "{{unispherehost}}"
      username: "{{username}}"
      password: "{{password}}"
      verifycert: "{{verifycert}}"
      cg_id: "{{cg_id}}"
```

```
    volumes:
      - vol_name: "Ansible_cg-1"
      - vol_id: "sv_1878"
    vol_state: "{{vol_state_present}}"
    state: "{{state_present}}"
```

Remove volumes from a consistency group

Learn how to remove volumes (LUNs) from a consistency group on Unity storage systems.

You can remove volumes (LUNs) from a consistency group by running this playbook. The syntax is as follows:

```
- name: Remove volumes from consistency group
  dellemc_unity_consistencygroup:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    cg_name: "{{new_cg_name}}"
    volumes:
      - vol_id: "sv_21540"
      - vol_name: "Ansible_vol_Test"
    vol_state: "absent-in-group"
    state: "present"
```

Delete a consistency group

Learn how to delete a consistency group from a Unity storage system.

You can delete a consistency group from the Unity storage system by running this playbook. The syntax is as follows:

```
- name: Delete consistency group
  dellemc_unity_consistencygroup:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    cg_id: "{{cg_id}}"
    state: "absent"
```

Host Module

Learn about the host module and its supported functions.

The Host module is intended to manage hosts on Unity storage systems.

Host module supports the following functions:

- Create a host
- Create a host with FC or iSCSI initiators
- Add FC or iSCSI initiators to a host
- Get details of a host
- Modify host attributes
- Rename a host
- Remove FC and iSCSI initiators from a host
- Delete a host

Table 7. Host Module parameters

Name	Type	Description	Requirement	Choices/Default
host_name	str	Name of the host.	Mandatory when creating the host.	

Table 7. Host Module parameters(continued)

Name	Type	Description	Requirement	Choices/Default
host_id	str	Unique, system-assigned identifier for the host. You can use either host_id or host_name for modify and delete tasks. Cannot be used for the create host action.	Optional	
host_os	str	Operating system of the host.	Optional	Choices: • *Windows Client* • *Windows Server* • *Citrix XenServer* • *IBM VIOS* • *Linux* • *Mac OS* • *VMware ESXi* • *HP-UX* • *AIX* • *Solaris*
description	str	Host description.	Optional	
initiators	list	List of initiators to be added or removed from the host. Specify the initiator WWN or IQN.	Optional. Only for adding or removing initiators to a host.	
new_host_name	str	The new name for the host in the rename operation.	Optional	
state	str	State of the host.	Mandatory	Choices: • *present* • *absent*
initiator_state	str	Whether the initiators should be present or absent on the host.	Optional. Required when adding or removing initiators from the host.	Choices: • *present-in-host* - indicates that the initiators should exist on host. • *absent-in-host* - indicates that the initiators should not exist on host.

Create hosts

Learn how to add hosts to the Unity storage system.

You can add hosts with or without initiators to a Unity storage system by running this playbook. The syntax is as follows:

Create host

```
- name: Create a host
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_name: "ansible_test_host_1"
    description: "Host created for testing"
```

```
    host_os: "Linux"
    state: "present"
```

Create host with FC or iSCSI initiators

```
- name: Create a host with FC initiators
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_name: "ansible_test_host_2"
    description: "Host created for testing"
    host_os: "Linux"
    initiators:
      - "20:00:00:90:FA:13:82:34:10:00:00:90:FA:13:82:34"
      - "20:00:00:90:FA:13:81:8C:10:00:00:90:FA:13:81:8C"
    initiator_state: "present-in-host"
    state: "present"
```

Get the details for a host

Learn how to get details for a host on the Unity storage system

You can obtain details for a host by running this playbook. Syntax is as follows:

Get host details using the host name

```
- name: Get host details
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_name: "ansible_test_host_2"
    state: "present"
```

Get host details using the host ID

```
- name: Get host details
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_id: "Host_1111"
    state: "present"
```

Modify host attributes

Learn how to modify the attributes of a host on a Unity storage system.

You can modify the attributes for a host on a Unity storage system, including the host name, description, and host OS, by running this playbook. The syntax is as follows:

```
- name: Modify host attributes
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
```

```
    verifycert: "{{verifycert}}"
    host_name: "ansible_test_host_1"
    new_host_name: "ansible_test_host_modified"
    description: "Host modified"
    host_os: "Windows Server"
    state: "present"
```

Add initiators to a host

Learn how to add FC initiators, iSCSI initiators, or both to an existing host on the Unity storage system.

You can add FC initiators, iSCSI initiators, or both to an existing host by running this playbook. The syntax is as follows:

```
- name: Add FC and iSCSI initiators to a host
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_name: "ansible_test_host_1"
    initiators:
      - "20:00:00:90:FA:13:82:35:10:00:00:90:FA:13:82:35"
      - "iqn.1994-05.com.redhat:854b3d9fd168"
    initiator_state: "present-in-host"
    state: "present"
```

Remove initiators from a host

Learn how to remove initiators from an existing host on the Unity storage system.

You can remove initiators from a host by running this playbook. The syntax is as follows:

Remove initiators from a host using the host name

```
- name: Remove initiator from a host
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_name: "ansible_test_host_1"
    initiators:
      - "iqn.1994-05.com.redhat:854b3d9fd168"
    initiator_state: "absent-in-host"
    state: "present"
```

Remove initiators from a host using the host ID

```
- name: Remove initiator from a host
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_id: "Host_1111"
    initiators:
      - "20:00:00:90:FA:13:82:35:10:00:00:90:FA:13:82:35"
    initiator_state: "absent-in-host"
    state: "present"
```

NOTE: Host initiators can only be removed when they are in a logged-off state (when no active initiators paths are associated with the initiator).

Delete hosts

Learn how to delete a host from the Unity storage system.

You can delete a host by running this playbook. The syntax is as follows:

Delete a host by host name

```
- name: Delete a host
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_name: "ansible_test_host_2"
    state: "absent"
```

Delete a host by host ID

```
- name: Delete a host
  dellemc_unity_host:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    host_id: "Host_1111"
    state: "absent"
```

Snapshot module

Learn about Snapshot module and the supported functions.

Snapshot Module is intended to manage snapshots on Unity storage systems.

The following functions are supported by snapshot module:

- Create a snapshot for a consistency group
- Create a snapshot for a volume (LUN)
- Get snapshot details
- Map a host to a snapshot
- Unmap a host from a snapshot
- Modify attributes of a snapshot for a volume (LUN)
- Delete a snapshot

Table 8. Snapshot module parameters

Name	Type	Description	Requirement	Choices/Default
snapshot_name	str	The name of the snapshot.	Mandatory during create operation. For all other operations, specify either snapshot_name or snapshot_ID.	
snapshot_id	str	Unique, system-assigned ID of the snapshot.	Not applicable to create operation. For all other	

Table 8. Snapshot module parameters(continued)

Name	Type	Description	Requirement	Choices/Default
			operations, specify either `snapshot_name` or `snapshot_ID`.	
`description`	str	Description of the snapshot.	Optional	
`state`	str	Define whether the snapshot should exist or not.	Mandatory	Choices: • *present* - indicates that the snap should exist on the system. • *absent* - indicates that the snap should not exist on the system.
`vol_name`	str	Name of the volume.	Mandatory only for the volume snapshot creation operation, otherwise optional.	
`cg_name`	str	Name of the consistency group.	Mandatory only for a consistency group snapshot creation operation, otherwise optional.	
`new_snapshot_name`	str	The new name of the snapshot for the rename operation.	Optional	
`auto_delete`	bool	Specify whether the snapshot should automatically be deleted according to the pool automatic snapshot deletion policy.	Optional	Choices: • *true*-Snapshot will be automatically deleted in accordance with the pool automatic snapshot deletion policy. • *false* - Snapshot will not be automatically deleted in accordance with the pool automatic snapshot deletion policy.
`expiry_time`	str	The timestamp (in UTC) after which the snapshot will expire. **NOTE: If specified, `autodelete` must be *false*.**	Optional	UTC in the format "%m/%d/%Y %H:%M" For example: "04/15/2020 14:30"
`host_name`	str	The name of the host to be mapped to or unmapped from the snapshot.	Optional	
`host_id`	str	The ID of the host to be mapped to or unmapped from the snapshot.	Optional	

Table 8. Snapshot module parameters(continued)

Name	Type	Description	Requirement	Choices/Default
host_state	str	The state of the host.	Optional	Choices: • *mapped* - indicates that the snapshot should be mapped to the host or host group. • *unmapped* - indicates that the snapshot should not be mapped to the host or host group.

Create a snapshot of a volume

Learn how to create a snapshot of a volume (LUN) on a Unity storage system.

You can create a snapshot of a volume (LUN) by running this playbook. The syntax is as follows:

```
- name: Create snapshot of volume
  dellemc_unity_snapshot:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_name: "{{vol_name}}"
    snapshot_name: "{{snapshot_name}}"
    description: "{{description}}"
    expiry_time: "04/15/20 14:30"
    host_name: "{{host_name}}"
    host_state: "mapped"
    state: "present"
```

Create a snapshot of a consistency group

Learn how to create a snapshot of a consistency group on a Unity storage system.

You can create a snapshot of a consistence group by running this playbook. The syntax is as follows:

```
- name: Create snapshot of CG
  dellemc_unity_snapshot:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    cg_name: "{{cg_name}}"
    snapshot_name: "{{snapshot_name}}"
    description: "{{description}}"
    auto_delete: True
    state: "present"
```

Map a host to a volume snapshot

Learn how to map a host to a snapshot on Unity storage systems.

You can map a host to a snapshot by running this playbook. The syntax is as follows:

```
- name: Create snapshot of volume mapped to host
  dellemc_unity_snapshot:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
```

```
    verifycert: "{{verifycert}}"
    snapshot_name: "{{snapshot_name}}"
    host_name: "{{host_name}}"
    host_state: "mapped"
    state: "present"
```

NOTE: Only one host can be mapped to a volume snapshot at a time.

Get the details for a snapshot

Get details for a snapshot on a Unity storage system.

You can get details for a snapshot by running this playbook. The syntax is as follows:

```
- name: Get the details of snapshot
  dellemc_unity_snapshot:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    snapshot_name: "{{snapshot_name}}"
    state: "present"
```

Modify snapshot attributes

Learn how to modify attributes of a snapshot on Unity storage systems.

You can modify the details of a snapshot, such as expiration time, name, description, or state, by running this playbook. The syntax is as follows:

```
- name: Modify snapshot of volume
  dellemc_unity_snapshot:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    vol_name: "{{volume_name}}"
    snapshot_id: "{{snapshot_id}}"
    new_snapshot_name: "{{new_snapshot_name}}"
    description: "{{new_description}}"
    expiry_time: "04/15/2022 14:30"
    state: "present"
```

Unmap a host from a volume snapshot

Learn how to unmap a host snapshot on a Unity storage system.

You can unmap a snapshot from a host using this playbook. The syntax is as follows:

```
- name: Unmap the snapshot from host
  dellemc_unity_snapshot:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    snapshot_name: "{{snapshot_name}}"
    host_name: "{{host_name}}"
    host_state: "unmapped"
    state: "present"
```

Delete a snapshot

Learn how to delete a snapshot on a Unity storage system.

You can delete a snapshot from the system by running this playbook. The syntax is as follows:

```
- name: Delete a snapshot
  dellemc_unity_snapshot:
    unispherehost: "{{unispherehost}}"
    username: "{{username}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    snapshot_id: "{{snapshot_id}}"
    state: "absent"
```

NOTE: If a snapshot is currently attached to a host, you cannot delete it. First unmap the snapshot, then execute the delete action.

Snapshot Schedule module

Learn about the Snapshot schedule module and its supported functions.

The Snapshot schedule module is intended to provide protection to a volume or consistency group on Unity storage systems by taking snapshots of the resource on a regularly scheduled basis.

The following functions are supported in this module:

- Create a snapshot schedule with an hourly rule
- Create a snapshot schedule with a daily rule
- Create a snapshot schedule with an every n days rule
- Create a snapshot schedule with a weekly rule
- Create a snapshot schedule with a monthly rule
- Get details of a specific snapshot schedule
- Modify the attributes of a snapshot schedule
- Delete snapshot schedule

Table 9. Snapshot schedule module parameters

Name	Type	Description	Requirement	Choices/Default
name	str	Name of the snapshot schedule.	Mandatory during the create operation, otherwise optional.	
id	str	Unique, system-assigned ID of the snapshot schedule.	Not applicable to the create operation. All other operations require either the name or ID.	
type	str	The type of snapshot rule of the snapshot schedule.	Mandatory for the create and modify operations.	Choices: • *every_n_hours* • *every_day* • *every_n_days* • *every_week* • *every_month*
interval	int	Number of hours between snaps. Applicable only when the rule is *every_n_hours*.	Optional	
hours_of_day	list	Hours of the day when the snapshot will be taken.	Optional	

Table 9. Snapshot schedule module parameters(continued)

Name	Type	Description	Requirement	Choices/Default
		Applicable only when the rule is *every_day*.		
day_interval	int	Number of days between snaps. Applicable only when the rule is *every_n_days*.	Optional	
days_of_week	list	Day of the week for which the snapshot rule applies. Applicable only when the rule is *every_week*.	Optional	Choices: • *MONDAY* • *TUESDAY* • *WEDNESDAY* • *THURSDAY* • *FRIDAY* • *SATURDAY* • *SUNDAY*
day_of_month	int	Days of the month for which the snapshot rule applies. Applicable only when the rule is *every_month*.	Optional	Values must be between 1 and 31.
hour	int	The hour of the day when the snapshot will be taken. Applicable only when the rule is *every_n_days*, *every_week*, and *every_month*.	Optional	Value must be between 0 and 23.
minute	int	The minute offset from the hour when the snapshot will be taken.	Optional	Value must be between 0 and 59.
desired_retention	int	The number of days or hours for which the snapshot will be retained. **NOTE: Cannot be specified when `auto_delete` is *true*.**	Optional	
retention_unit	str	The unit of measurement for the retention interval of the snapshot.	Optional	Choices: • *hours* (default) • *days*
auto_delete	bool	Indicates whether the system will automatically delete snapshots.	Optional	Choices: • *True* • *False*
state	str	Defines whether the snapshot schedule should exist on the storage system.	Mandatory	Choices: • *present*--Indicates that the snapshot schedule will be present on the system for create, modify, or get details operations. • *absent* --Indicates that the snapshot schedule will not be present on the system for the delete operation.

Create a snapshot schedule

Learn how to create a snapshot schedule on the Unity storage system.

You can create a snapshot schedule using this playbook. Syntax examples are as follows:

Create a snapshot schedule that takes snapshots every n hours

```
- name: Create snapshot schedule with "every_n_hours" rule
  dellemc_unity_snapshotschedule:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    name: "{{schedule_name1}}"
    type: "every_n_hours"
    interval: 6
    minute: 10
    desired_retention: 24
    retention_unit: "hours"
    state: "present"
```

Create a snapshot schedule that takes snapshots every day

```
- name: Create snapshot schedule with "every_day" rule
  dellemc_unity_snapshotschedule:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    name: "{{schedule_name2}}"
    type: "every_day"
    hours_of_day:
      - 8
      - 14
    auto_delete: True
    state: "present"
```

Create a snapshot schedule that takes snapshots every n days

```
- name: Create snapshot schedule with "every_n_days" rule
  dellemc_unity_snapshotschedule:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    name: "{{schedule_name3}}"
    type: "every_n_days"
    day_interval: 2
    hour: 13
    minute: 29
    auto_delete: True
    state: "present"
```

Create a snapshot schedule that takes snapshots every week

```
- name: Create snapshot schedule with "every_week" rule
  dellemc_unity_snapshotschedule:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
```

```
    name: "{{schedule_name4}}"
    type: "every_week"
    days_of_week:
      - SUNDAY
      - THURSDAY
    hour: 8
    auto_delete: True
    state: "present"
```

Create a snapshot schedule that takes snapshots every month

```
- name: Create snapshot schedule with "every_month" rule
  dellemc_unity_snapshotschedule:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    name: "{{schedule_name5}}"
    type: "every_month"
    day_of_month: 15
    auto_delete: True
    state: "present"
```

Get the details of a snapshot schedule

Learn how to get the details of a snapshot schedule on Unity systems.

You can get the details of a specified snapshot schedule using this playbook. The syntax is as follows:

Get the details of a snapshot schedule using the schedule name

```
- name: Get details of snapshot schedule using name
  dellemc_unity_snapshotschedule:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    name: "{{schedule_name1}}"
    state: "present"
```

Get the details of a snapshot schedule using the schedule ID

```
- name: Get details of snapshot schedule using id
  dellemc_unity_snapshotschedule:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    id: "{{schedule_id1}}"
    state: "present"
```

Modify snapshot schedule attributes

Learn how to modify the attributes of a snapshot schedule.

You can modify the attributes of a snapshot schedule, including type, interval, retention, and retention unit using this playbook. The syntax is as follows.

```
- name: Modify the attributes of snapshot schedule using id
  dellemc_unity_snapshotschedule:
```

```
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    id: "{{schedule_id1}}"
    type: "every_n_hours"
    interval: "{{interval2}}"
    desired_retention: "{{desired_retention2}}"
    retention_unit: "{{retention_unit_days}}"
    state: "present"
```

Delete a snapshot schedule

Learn how to delete a snapshot schedule from a Unity storage system.

You can delete a snapshot schedule from the storage system using this playbook. The syntax is as follows:

```
- name: Delete the snapshot schedule
  dellemc_unity_snapshotschedule:
    unispherehost: "{{unispherehost}}"
    username: "{{user}}"
    password: "{{password}}"
    verifycert: "{{verifycert}}"
    name: "{{schedule_name1}}"
    state: "absent"
```