

Ansible Modules for Dell EMC Isilon

Product Guide

Version 1.1

Notes, cautions, and warnings

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

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

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

Introduction

Introduces the Ansible modules for Isilon.

Topics:

- [Product overview](#)

Product overview

Ansible Modules for Dell EMC Isilon automate configuration and deployment.

The Ansible Modules for Dell EMC Isilon provide management capabilities.

The following Ansible modules are available:

- [Gather facts](#)
- [User](#)
- [Group](#)
- [Access zone](#)
- [Filesystem](#)
- [NFS export](#)
- [SMB \(CIFS\) share](#)
- [Snapshot](#)
- [Snapshot schedule](#)
- [Smart quotas](#)

Each module performs actions, such as getting, modifying, creating, deleting, and so on. The gather facts module is intended to gather high-level facts about the array.

Installation

Instructions on how to install the SDK and the Ansible modules.

- Supported platforms:
 - Isilon clusters with OneFS version 8.0 or above
 - Ansible 2.7 or above
- Supported (client) operating systems:
 - Red Hat Enterprise Linux 7.6

Prerequisites

Python library for Isilon (version 8.1.1) should be pre-installed on the client.

Topics:

- [Install the Isilon SDK](#)
- [Install Ansible Modules for Dell EMC Isilon](#)

Install the Isilon SDK

Instructions for installing the Isilon SDK.

About this task

Prior to installing the Ansible Modules for Dell EMC Isilon you must install the Isilon SDK, named `isi-sdk-8-1-1`. All Ansible for Isilon modules check for the library. If it is not present, a message appears. The same version works for all OneFS versions from 8.0 through 9.0.

Steps

1. Download the Python library for Isilon from <https://pypi.org/project/isi-sdk-8-1-1/>.
2. Using PIP, the installer package for Python, install the Isilon SDK named `isi-sdk-8-1-1`:

```
pip install isi-sdk-8-1-1
```

Install Ansible Modules for Dell EMC Isilon

Instructions on how to install the Ansible for Isilon modules.

About this task

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

Steps

1. Create necessary folders structure:
 - a. Create the `dell EMC` folder in the following path, if it is not available:


```
/usr/lib/<python_version>/site-packages/ansible/modules/storage
```
 - b. Create the `dell` folder in the following path, if it is not available:


```
/usr/lib/<python_version>/site-packages/ansible/module_utils/storage
```
2. Copy the Ansible modules to the appropriate locations in the virtual machine.
 - a. Copy `dell EMC ansible_isilon_utils.py` and `__init__.py` from the `/utils` directory to the following locations:


```
/usr/lib/<python_version>/site-packages/ansible/module_utils/storage/dell
```
 - b. Copy all the module python files from the `isilon/library` directory to the following:

```
/usr/lib/<python_version>/site-packages/ansible/modules/storage/dellemc
```

- c. Copy the `dellemc_isilon.py` from the `/doc_fragments` directory to the following:

```
/usr/lib/<python_version>/site-packages/ansible/plugins/doc_fragments/
```

Ansible Modules for Isilon

This chapter contains the following topics:

Topics:

- [Module overview](#)
- [Gather facts module](#)
- [User module](#)
- [Group module](#)
- [Access zone module](#)
- [Filesystem module](#)
- [NFS export](#)
- [SMB \(CIFS\) shares](#)
- [Snapshot module](#)
- [Snapshot schedule module](#)
- [Smart quotas module](#)

Module overview

The Ansible modules use common access parameters.

Parameter name	Explanation	Mandatory/ Optional	Default	Comments
onefs_host	IP address or FQDN of the Isilon cluster	Mandatory		
port_no	Port number	Optional	8080	
api_user	Username for accessing the REST API	Mandatory		Credentials can be encrypted using Ansible vault.
api_password	Password for accessing the REST API	Mandatory		Users must have admin access to the access zone on which they want to operate.
verify_ssl	Boolean parameter to indicate secure access	Mandatory		

Gather facts module

The gather facts module is used to gather high-level facts about the array.

This module collects information about the Isilon cluster.

The gather facts module supports the following functions:

- List high-level attributes of the Isilon cluster
- List all the access zones in an Isilon cluster
- List the nodes in an Isilon cluster
- List the authentication providers for an access zone
- List the users and groups of an access zone

Get attributes, access zones, and nodes of the Isilon cluster

The user can get attributes, access zones, and nodes of the Isilon cluster by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Get attributes, access_zones and nodes of the Isilon cluster
  dellemc_isilon_gatherfacts:
    onefs_host: "{{onefs_host}}"
    port_no: "{{isilonport}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    gather_subset:
      - attributes
      - access_zones
      - nodes
```

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

Get the list of authentication providers, users, and groups for an access zone

The user can get a list of authentication providers, users, and groups for an access zone of the Isilon cluster by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Get list of authentication providers, users and groups for an access zone of the
Isilon cluster
  dellemc_isilon_gatherfacts:
    onefs_host: "{{onefs_host}}"
    port_no: "{{isilonport}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    access_zone: "{{access_zone}}"
    gather_subset:
      - providers
      - users
      - groups
```

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

Gather facts parameters

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

Table 1. Parameters

Parameter name	Explanation	Mandatory/Optional	Default
access_zone	The base access zone	Optional	System
gather_subset	<ul style="list-style-type: none">List of string variables to specify the Isilon entities for which information is requiredList of all Isilon entities the module supports<ul style="list-style-type: none">attributes	Mandatory	

Table 1. Parameters(continued)

Parameter name	Explanation	Mandatory/ Optional	Default
	<ul style="list-style-type: none"> ○ access_zones ○ nodes ○ providers ○ users ○ groups 		

User module

The user module is used for managing users on the Isilon.

The user module supports the following functionality:

- Creation of a local user.
- Modification to a local user.
- Addition of roles to users of all provider types in system access zone.
- Removal of roles from users of all provider types in system access zone.
- Getting details of the local, ads, ldap, and file users. To get the details of the user authenticated by active directories `user_name` should include the domain name of the AD server in the specified format `"DOMAIN-NAME\\user_name"` or `'DOMAIN-NAME\\user_name'`.
- Deletion of a local user.

Create a user with a role

A user can be created with an assigned role for the system access zone by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create a user with a role
  dellemc_isilon user:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    user_name: "test-user"
    password: "asdf"
    state: "present"
    role_name: "AuditAdmin"
    role_state: "present-for-user"
```

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

Create a user in a non-system access zone with an enabled account

A user can be created in a non-system access zone with an enabled account by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create a user with enabled account
  dellemc_isilon user:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    access_zone: "sample-zone"
    user_name: "test-user"
    password: "asdf"
```



```
enabled: "True"
state: "present"
```

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

Get the details of a user

The details of the user can be fetched by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Get details of a user
  dellemc_isilon_user:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    access_zone: "sample-zone"
    user_name: "test-user"
    state: "present"
```

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

Remove a role assigned to a user

A role that is assigned to a user can be removed by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Remove a role from an existing user
  dellemc_isilon_user:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    user_name: "test-user"
    state: "present"
    role_name: "AuditAdmin"
    role_state: "absent-for-user"
```

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

Update user details

User details can be modified using `user_id` or `user_name` by running the appropriate playbook.

This is an example of the syntax of a playbook for adding the email and full name to the existing user using `user_id`.

```
- name: Update user details
  dellemc_isilon_user:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    user_id: "2002"
    email: "test-user@dell.com"
    full_name: "Sample Test User"
    state: "present"
```

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

Add a role to a user

A role can be added to a user of the system access zone by listing either `user_name` or `user_id`.

This is an example syntax of a playbook where a role `AuditAdmin` is added to a user using a `user_name`.

```
- name: Add role to existing user
  dellemc_isilon_user:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    user_name: "test-user"
    state: "present"
    role_name: "AuditAdmin"
    role_state: "present-for-user"
```

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

Delete a user

A user can be deleted by mentioning the state as `absent` for a given `user_id` or `user_name`.

This is an example of the syntax of a playbook where a user is deleted using `user_name`.

```
- name: Delete a user
  dellemc_isilon_user:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    access_zone: "sample-zone"
    user_name: "test-user"
    state: "absent"
```

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

User module parameters

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

Table 2. Parameters

Parameter	Explanation	Mandatory/ Optional	Default	Comments
<code>user_name</code>	The name of the user account	Optional		The name of the user account is required for user creation. For all other operations, either <code>user_name</code> or <code>user_id</code> is required.
<code>user_id</code>	The ID of the user account	Optional		The <code>user_id</code> is autogenerated during creation. For all other operations, either <code>user_id</code> or <code>user_name</code> is required.
<code>password</code>	The password of the user account	Optional		<ul style="list-style-type: none">• Mandatory for creation of user.• If given for any other operation, it is ignored.
<code>access_zone</code>	The zone in which the user account exists	Optional	system	<ul style="list-style-type: none">• For all operations, <code>access_zone</code> is optional.• If it is not mentioned, the operation is performed in the system access zone.

Table 2. Parameters(continued)

Parameter	Explanation	Mandatory/ Optional	Default	Comments
provider_type	The authentication type that is configured to allow users to authenticate	Optional	local	<ul style="list-style-type: none"> The provider_type specifies the authentication provider to authenticate the user. Supported authentication providers are ldap, ads, local, and file. The user creation, modification, and deletion are allowed only if the provider is local. The details of users who are authenticated by any provider type can be fetched. If it is not mentioned, then the operation is performed assuming the user is authenticated locally.
home_directory	The directory which is treated as home for the user	Optional		<ul style="list-style-type: none"> Used in the creation and modification of the user account details. If not given during creation, [access_zone base directory] /home/name is assigned as home directory.
primary_group	The group to which the user account belongs.	Optional		Used in the creation and modification of the user account details
enabled	Enables the users to the access rights	Optional		<ul style="list-style-type: none"> Optional parameter for creation and modification. By default, Ansible creates a user with enabled as False.
full_name	The full name of the user	Optional		Can be given during creating and updating the user account.
email	The email address of the user	Optional		Can be given during creating and updating the user account.
shell	Specifies the path to the shell for the user	Optional		Can be given during creating and updating the user account.
state	Mentions the function that is performed	Mandatory		<ul style="list-style-type: none"> During creation and getting the user, the state is present. During deletion, the state is absent.
role_name	The name of the role that the user can perform	Optional		<ul style="list-style-type: none"> While creating a user, it is optional. It is mandatory for adding and removing a user from a role.
role_state	State that mentions adding or removing the user from a role	Optional		<ul style="list-style-type: none"> During create and update the field is optional. It is mandatory for adding and removing a user from a role. For adding a user to role, it is present-for-user. For removing a user from a role, it is absent-for-user.

Group module

The group module manages group permissions and memberships in Isilon.

The creation, deletion, and modification of a group is allowed only for the local provider type.

The group details of any provider type can be fetched.

Only local users can be added and removed from the local groups.

The group module has the following functions:

- Create a local group
- Modify a local group
- Add local users to a local group
- Remove local users from a local group
- Get the details of a group
- Delete a local group

Create a local group in a system access zone

The user can create a local group in a system access zone by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create a Group
  dellemc_isilon_group:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    access_zone: "{{access_zone}}"
    provider_type: "{{provider_type}}"
    group_name: "{{group_name}}"
    state: "present"
```

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

Create a local group in non-system access zone

The user can create a local group in a non-system access zone with a user member added to the group by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create Group with Users
  dellemc_isilon_group:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    provider_type: "local"
    access_zone: "sample-zone"
    group_name: "{{group_name}}"
    users:
      - user_id: "2012"
      - user_name: "test_user_1"
      - user_id: "2014"
    user_state: "present-in-group"
    state: "present"
```

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

Get the details of a group

The user can get the details of the group using `group_name` or `group_id` by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Get Details of the Group using Group Id
  dellemc_isilon_group:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    provider_type: "{{provider_type}}"
    access_zone: "{{access_zone}}"
    group_id: "{{group_id}}"
    state: "present"
```

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

Add a user to a local group

The user can be added to the local group using the `user_name` or `user_id` by running the appropriate playbook.

This is an example of the syntax of a playbook where multiple users are added to a group:

```
- name: Add users to a Group.
  dellemc_isilon_group:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    provider_type: "{{provider_type}}"
    access_zone: "{{access_zone}}"
    group_id: "{{group_id}}"
    users:
      - user_id: "2012"
      - user_name: "test_user_1"
      - user_id: "2014"
    user_state: "present-in-group"
    state: "present"
```

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

Remove a user from a group

The user can be removed from the local group using the `user_name` with `user_id` by running the appropriate playbook.

This is an example of the syntax of a playbook for removing multiple users from a group:

```
- name: Remove users from a Group.
  dellemc_isilon_group:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    provider_type: "{{provider_type}}"
    access_zone: "{{access_zone}}"
    group_id: "{{group_id}}"
    users:
      - user_id: "2012"
      - user_name: "test_user_1"
      - user_id: "2014"
    user_state: "absent-in-group"
    state: "present"
```

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

Delete a group

The user can delete a local group using the `group-name` or `group-id` by running the appropriate playbook.

This is an example of the syntax of a playbook where a group is delete using `group_name`:

```
- name: Delete the Group using Group Name
  dellemc_isilon_group:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    provider_type: "{{provider_type}}"
    access_zone: "{{access_zone}}"
    group_name: "{{group_name}}"
    state: "absent"
```

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

Group module parameters

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

Table 3. Parameters

Parameter name	Explanation	Mandatory/ Optional	Default	Comments
group_name	The name of the group	Optional		The group name is required for creation of the group. For all other operations, either <code>group_name</code> or <code>group_id</code> is required.
group_id	The ID of the group	Optional		In the creation of the group, <code>group_id</code> is not required (auto-generated). For all other operations, either <code>group_name</code> or <code>group_id</code> is required.
users	Multiple users can be specified either by the <code>user_name</code> or by the <code>user_id</code>	Optional		In the users, multiple users can be mentioned using the <code>user_name</code> or <code>user_id</code> . The users section can be mentioned during creating, adding, and removing users from the group. <ul style="list-style-type: none">• <code>users</code>• <code>user_name: sample_user</code>• <code>user_id: 2007</code>
access_zone	The zone in which the group exists	Optional	system	The zone in which the group exists. At the time of creation, it acts as a parameter. For all other operations, it acts as a filter.
provider_type	The authentication type for the group	Optional	local	At the time of creation, <code>provider_type</code> acts as a parameter. For all other operations, it acts as a filter.

Table 3. Parameters(continued)

Parameter name	Explanation	Mandatory/Optional	Default	Comments
user_state	The state of the users in the group Choices: [present-in-group, absent-in-group]	Optional		To add the users to the group, the user_state is present-in-group. To remove the users from the group the user_state is absent-in-group.
state	The state of the group in the Isilon system choices: [present, absent]	Mandatory		The state is present for all the operations except deletion. For deletion, the state is absent.

Access zone module

The access zone module allows the user to get details and modify the settings of an access zone.

Creation and deletion of access zones is not allowed through Ansible.

The access zone module has the following functions:

- Get access zone details.
- Modify the default SMB settings of an access zone.
- Modify the default NFS settings of an access zone.

Get access zone details

The user can get access zone details by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Get details of access zone including smb and nfs settings
  dellemc_isilon_accesszone:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    az_name: "{{access zone}}"
    state: "present"
```

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

Modify the default SMB settings of an access zone

The user can modify the default SMB settings of an access zone by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify smb settings of access zone
  dellemc_isilon_accesszone:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    az_name: "{{access zone}}"
    state: "present"
    smb:
      create_permissions: 'default acl'
      directory_create_mask: '777'
      directory_create_mode: '700'
```

```

file_create_mask: '700'
file_create_mode: '100'
access_based_enumeration: true
access_based_enumeration_root_only: false
ntfs_acl_support: true
oplocks: true

```

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

Modify the default NFS settings of an access zone

The user can modify the default NFS settings of an access zone by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Modify nfs settings of access zone
  dellemc_isilon_accesszone:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    az_name: "{{access zone}}"
    state: "present"
    nfs:
      commit_asynchronous: false
      nfsv4_allow_numeric_ids: false
      nfsv4_domain: 'localhost'
      nfsv4_no_domain: false
      nfsv4_no_domain_uids: false
      nfsv4_no_names: false

```

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

Access zone module parameters

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

Table 4. Parameters

Parameter name	Explanation	Mandatory/ Optional	Default	Comments
az_name	Name of the access zone	Mandatory		Not case sensitive
state	State of the access zone choices: [absent, present]	Mandatory		
smb	Default SMB settings of the access zone	Optional		Sub-options to include: <ul style="list-style-type: none"> create_permissions (choices: default acl, Inherit mode bits, Use create mask and mode, default value: default acl) directory_create_mask (Type: str, Default Value=700 (octal)) directory_create_mode (Type:str, Default Value=None) file_create_mask (Type:str, Default Value=700 (octal)) file_create_mode (Type:str, Default Value='100' (octal)) access_based_enumeration (Type:bool, Default Value:false)

Table 4. Parameters(continued)

Parameter name	Explanation	Mandatory/ Optional	Default	Comments
				<ul style="list-style-type: none"> access_based_enumeration_root_only (Type:bool, Default Value:false) ntfs_acl_support (Type:bool, Default Value:true) oplocks (Type:bool, Default Value:true)
nfs	Default NFS settings of the access zone	Optional		Sub-options to include: <ul style="list-style-type: none"> commit_asynchronous (Type:bool, Default Value=false) nfsv4_allow_numeric_ids (Type:bool, Default Value=true) nfsv4_domain (Type:str, Default Value=localhost) nfsv4_no_domain (Type:bool, Default Value=false) nfsv4_no_domain_uids (Type:bool, Default Value=true) nfsv4_no_names (Type:bool, Default Value=false)

Filesystem module

The user can create, delete, modify, and get details of a filesystem.

The owner and group can be local, file, ldap, or ads.

The quota can be increased as well as reduced. Once a quota has been assigned to a filesystem, it can be removed by specifying its state as absent.

The filesystem module has the following functions:

- Create filesystem with quota in given access zone.
- Create filesystem without quota.
- Get filesystem details.
- Modify filesystem.
- Delete the filesystem.

Create filesystem with quota in given access zone

The user can create a filesystem with quota in a given access zone by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create a filesystem with Quota and ACL in POSIX mode
  dellemc_isilon_filesystem:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{path_ansible}}"
    access_zone: "{{access_zone}}"
    owner:
      name: 'ldap_test_user_1'
      provider_type: 'ldap'
```

```

group:
  name: 'sample_ldap_group_2'
  provider_type: 'ldap'
quota:
  include_snap_data: False
  include_data_protection_overhead: False
  soft_limit_size: 5
  hard_limit_size: 10
  cap_unit: "GB"
  quota_state: "present"
access_control: "{{access_control}}"
recursive: "{{recursive}}"
state: "{{state_present}}"

```

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

Create filesystem without quota

The user can create a filesystem without quota in a default system access zone by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Create a filesystem in System Access Zone
  dellemc_isilon_filesystem:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{new_path_system}}"
    owner:
      name: 'ldap_test_user_1'
      provider_type: 'ldap'
    group:
      name: 'sample_ldap_group_2'
      provider_type: 'ldap'
    access_control: "{{access_control}}"
    recursive: "{{recursive}}"
    state: "{{state_present}}"

```

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

Get filesystem details

The user can get filesystem details by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Get filesystem details
  dellemc_isilon_filesystem:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{path}}"
    state: "{{state_present}}"

```

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

Modify filesystem hard quota

The user can modify a filesystem hard quota by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Modify Filesystem Hard Quota

```

```
dellemc_isilon_filesystem:
  onefs_host: "{{isilonhost}}"
  port_no: "{{isilonport}}"
  verify_ssl: "{{verify_ssl}}"
  username: "{{user}}"
  password: "{{password}}"
  path: "{{path}}"
  access_zone: "{{access_zone}}"
  quota:
    hard_limit_size: 15
    cap_unit: "GB"
    quota_state: "present"
  state: "{{state_present}}"
```

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

Modify filesystem owner, group, and ACL

The user can modify the filesystem owner, group, and ACL by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify Filesystem Owner, Group and ACL
  dellemc_isilon_filesystem:
    onefs_host: "{{isilonhost}}"
    port_no: "{{isilonport}}"
    verify_ssl: "{{verify_ssl}}"
    username: "{{user}}"
    password: "{{password}}"
    path: "{{path}}"
    access_zone: "{{access_zone}}"
    owner:
      name: 'ansible_user'
      provider_type: 'ldap'
    group:
      name: 'ansible_group'
      provider_type: 'ldap'
    access_control: "{{new_access_control}}"
    state: "{{state_present}}"
```

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

Delete the filesystem

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

This is an example of the syntax of a playbook:

```
- name: Delete a filesystem
  dellemc_isilon_filesystem:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{path_system}}"
    state: "{{state_absent}}"
```

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

Filesystem parameters

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

Table 5. Parameters

Parameter name	Explanation	Mandatory/Optional	Default	Comments
path	The directory path	Mandatory	Nil	For non-system access zones, this path is a relative path from the base of the access zone.
access_zone	The base access zone	Optional	Nil	If no access zone is specified, the filesystem is in the system access zone.
owner	User permissions	Optional	Nil	Name is mandatory. Provider_type is optional with a default value of local. Owner is mandatory only for creating a filesystem.
group	Group permissions	Optional	Nil	Name is mandatory. Provider_type is optional with a default value of local.
access_control	The ACL value for the directory	Optional	Nil	User can either provide input, such as private_read, private, public_read, public_read_write, public or in POSIX format (that is, 0700).
recursive	Creates intermediate folders recursively when set to true.	Optional	Nil	If recursive is false and an intermediate path is missing, the error is propagated back from Isilon.
quota		Optional	Nil	<ul style="list-style-type: none"> The following sub-options are supported: <ul style="list-style-type: none"> include_snap_data (boolean) include_data_protection_overhead (boolean) advisory_limit_size (int) soft_limit_size (int) hard_limit_size (int) cap_unit (MB, GB, or TB) quota_state (present or absent) The default grace period is 7 days. Modification of grace period is not supported. Modification of include_snap_data is not supported. The default capacity unit is GB.
list_snapshots	If set to True, filesystem snapshot details are returned.	Optional	Nil	
state	The state of the filesystem choices: [present, absent]	Mandatory	Nil	The state is present for all the operations except deletion. For deletion, the state is absent.

NFS export

Managing NFS exports on an Isilon system includes creating NFS export for a directory in an access zone, adding or removing clients, modifying different parameters of the export and deleting export.

The NFS export module has the following functions:

- Create an NFS export.
- Get NFS export details.
- Assign/remove access of clients (clients, root-clients, read-write clients, and read-only clients).
- Set parameters, such as `read_only` flag and `sub_directories_mountable` flag.
- Delete NFS export.

Create an NFS export

The user can create an NFS export by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create an NFS export
  dellemc_isilon_nfs:
    onefs_host: "{{onefs_host}}"
    api_user: "{{user}}"
    api_password: "{{password}}"
    path: "{{path}}"
    access_zone: {{access_zone}}
    read_write_clients:
      - client1
    sub_directories_mountable: True
    client_state: 'present-in-export'
    description: 'description'
    state: 'present'
```

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

Get NFS export details

The user can get NFS export details by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Get NFS Export details
  dellemc_isilon_nfs:
    onefs_host: "{{onefs_host}}"
    api_user: "{{user}}"
    api_password: "{{password}}"
    verify_ssl: "{{verify_ssl}}"
    path: "{{path}}"
    access_zone: {{access_zone}}
    state: 'present'
```

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

Add a root client

The user can add a client to NFS export by running the appropriate playbook.

This is an example of the syntax of a playbook for adding a root client.

```
- name: Add a root client
  dellemc_isilon_nfs:
    onefs_host: "{{onefs_host}}"
    api_user: "{{user}}"
```

```
api_password: "{{password}}"
verify_ssl: "{{verify_ssl}}"
path: "{{path}}"
access_zone: {{access_zone}}
root_clients:
- {{client4}}
client_state: 'present-in-export'
state: 'present'
```

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

Set sub_directories_mountable flag

The user can set `sub_directories_mountable` flag by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Set sub_directories_mountable flag to True
  dellemc_isilon_nfs:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    path: "{{path}}"
    access_zone: {{access_zone}}
    sub_directories_mountable: True
    state: 'present'
```

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

Remove a root client

The user can remove a client by running the appropriate playbook.

This is an example of the syntax of a playbook for removing a root client.

```
- name: Remove a root client
  dellemc_isilon_nfs:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    path: "{{path}}"
    access_zone: {{access_zone}}
    root_clients:
    - {{client4}}
    client_state: 'absent-in-export'
    state: 'present'
```

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

Set read_only flag to False

The user can modify the NFS export in system access zone to read-only by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Set read_only flag to False
  dellemc_isilon_nfs:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    path: "{{path}}"
```

```

access_zone: "{{access_zone}}"
read_only: False
state: 'present'

```

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

Modify the description

The user can modify the description by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Modify description
  dellemc_isilon_nfs:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    path: "{{path}}"
    access_zone: {{access_zone}}
    description: "new description"
    state: 'present'

```

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

Delete NFS export

The user can delete the NFS export by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Delete NFS Export
  dellemc_isilon_nfs:
    onefs_host: "{{onefs_host}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    verify_ssl: "{{verify_ssl}}"
    path: "{{path}}"
    access_zone: "{{access_zone}}"
    state: 'absent'

```

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

NFS export module parameters

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

Table 6. Parameters

Parameter name	Explanation	Mandatory/ Optional	Format	Default	Comments
path	<p>Directory path to be exported. For non-system access zones, this path is a relative path from the base of the access zone.</p> <p>The path must exist. The NFS</p>	Mandatory	str		<p>Ansible module only supports one export for a given path.</p> <p>If there multiple exports present with the same path, operations of such exports fail.</p>

Table 6. Parameters(continued)

Parameter name	Explanation	Mandatory/ Optional	Format	Default	Comments
	module does not create the path.				
access_zone	Specifies the zone in which the export is valid.	Optional	str	system	<zone> is used in the documentation.
clients	Specifies the clients to the export. The type of access to clients in this list is determined by the read_only parameter.	Optional	list[str]		This list can be changed anytime during the lifetime of the NFS export.
root_clients	Specifies the clients with root access to the export.	Optional	list[str]		This list can be changed anytime during the lifetime of the NFS export.
read_only_clients	Specifies the clients with read-only access to the export even when the export is read/write.	Optional	list[str]		This list can be changed anytime during the lifetime of the NFS export.
read_write_clients	Specifies the clients with both read and write access to the export even when the export is set to read-only.	Optional	list[str]		This list can be changed anytime during the lifetime of the NFS export.
read_only	Specifies whether the export is read-only or read-write. This parameter only has effect on the clients list and not the other three lists.	Optional	bool		<read_only> is used in the documentation. This setting can be modified any time. If it is not set at the time of creation, the export is of type read/write.
sub_directories_mountable	True if all directories under the specified paths are mountable. If not set, sub-directories are not mountable.	Optional	bool		<all_dirs> is used in the documentation. This setting can be modified any time. If it is not set at the time of creation, the sub-directories are not mountable.
state	[present, absent]	Mandatory	str		
client_state	[present-in-export, absent-in-export]	Mandatory when adding or removing clients from the export			Define whether the clients can access the NFS export. present-in-export indicates that the clients can access the NFS export. absent-in-export indicates that the client cannot

Table 6. Parameters(continued)

Parameter name	Explanation	Mandatory/ Optional	Format	Default	Comments
					access the NFS export. Required when adding or removing access of clients from the export. While removing clients, only the specified clients are removed from the export. Others remain as is.
description		Optional, can be modified too.	str		Can be modified by passing a new value.

SMB (CIFS) shares

The SMB shares module allows the user to create, modify, and delete an SMB share.

The SMB shares module has the following functions:

- Create SMB share for a system or non-system access zone.
- Modify various supported attributes of an SMB share.
- Add, remove, or modify permissions for users, groups, and wellknown.
- Get SMB share details.
- Delete an SMB share.

Create an SMB share for a system access zone

The user can create an SMB share for a system access zone by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create SMB share for system access zone
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    share_name: "{{name}}"
    path: "{{system_az_path}}"
    description: "{{description}}"
    permissions:
      - user_name: "{{system_az_user}}"
        permission: "full"
        permission_type: "allow"
      - group_name: "{{system_az_group}}"
        permission: "read"
        permission_type: "allow"
      - wellknown: "everyone"
        permission: "read"
        permission_type: "allow"
    state: "{{state_present}}"
```

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

Create SMB share for non-system access zone

The user can create an SMB share for a non-system access zone by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create SMB share for non system access zone
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    share_name: "{{name}}"
    path: "{{non_system_az_path}}"
    access_zone: "{{non_system_access_zone}}"
    description: "{{description}}"
    permissions:
      - user_name: "{{non_system_az_user}}"
        permission: "read"
        permission_type: "allow"
      - group_name: "{{non_system_az_group}}"
        permission: "read"
        permission_type: "allow"
      - wellknown: "everyone"
        permission: "read"
        permission_type: "allow"
    state: "{{state_present}}"
```

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

Get SMB share details

The user can get SMB share details by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Get SMB share details
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    share_name: "{{name}}"
    state: "{{state_present}}"
```

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

Modify the name of an existing SMB share

The user can modify the name of an existing SMB share by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify name for an existing SMB share
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    share_name: "{{name}}"
    new_share_name: "{{new_name}}"
    access_zone: "{{non_system_access_zone}}"
    description: "new description"
    state: "{{state_present}}"
```

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

Modify an SMB share to add user permission

The user can modify an SMB share to add user permission by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify user permission for SMB share
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    share_name: "{{name}}"
    path: "{{system_az_path}}"
    description: "{{description}}"
    permissions:
      - user_name: "{{system_az_user}}"
        permission: "full"
        permission_type: "allow"
      - group_name: "{{system_az_group}}"
        permission: "write"
        permission_type: "allow"
      - wellknown: "everyone"
        permission: "write"
        permission_type: "deny"
    state: "{{state_present}}"
```

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

Modify user permission for an SMB share

The user can modify user permission for an SMB share by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify user permission for smb share
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    share_name: "{{name}}"
    path: "{{non_system_az_path}}"
    access_zone: "{{non_system_access_zone}}"
    description: "{{description}}"
    permissions:
      - user_name: "{{non_system_az_user}}"
        permission: "full"
        permission_type: "allow"
    state: "{{state_present}}"
```

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

Modify an SMB share to remove group permission

The user can modify an SMB share to remove group permission by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify SMB share to remove group permission
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
```

```

api_user: "{{api_user}}"
api_password: "{{api_password}}"
share_name: "{{name}}"
access_zone: "{{non_system_access_zone}}"
permissions:
  - group_name: "{{group1}}"
    permission: "write"
    permission_type: "deny"
    provider_type: "ads"
state: "{{state_present}}"

```

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

Modify group permission for an SMB share

The user can modify group permission for an SMB share by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Modify group permission for smb share
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    share_name: "{{name}}"
    path: "{{path}}"
    access_zone: "{{non_system_access_zone}}"
    permissions:
      - group_name: "{{group1}}"
        permission: "write"
        permission_type: "allow"
        provider_type: "ads"
    description: "smb description"
    state: "{{state_present}}"

```

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

Modify the description for an existing SMB share

The user can modify the description for an existing SMB share by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Modify description for an existing SMB share
  dellemc_isilon_smb:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    share_name: "{{name}}"
    access_zone: "{{non_system_access_zone}}"
    description: "new description"
    state: "{{state_present}}"

```

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

Delete an SMB share

The user can delete an existing SMB share by running the appropriate playbook.

This is an example of the syntax of a playbook:

```

- name: Delete SMB share
  dellemc_isilon_smb:

```

```
onefs_host: "{{onefs_host}}"
verify_ssl: "{{verify_ssl}}"
api_user: "{{api_user}}"
api_password: "{{api_password}}"
share_name: "{{name}}"
state: "{{state_absent}}"
```

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

SMB shares parameters

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

Table 7. Parameters

Parameter name	Type	Explanation	Mandatory/Optional	Default	Comments
share_name	str	Name of the SMB share	Mandatory		All the operations are supported through this parameter.
path	str	Path of the SMB share	Optional		This parameter is mandatory only for the create operation.
access_zone	str	Access zone that contains this share	Optional	system	If not specified, it is system access zone. For non-system access zone, the effective path where the SMB is created is determined by the base path of the access zone and the path provided by the user in the playbook. For system access zone, the effective path is the absolute path provided by the user in the playbook.
description	str	Description about the SMB share	Optional		
permissions	list[dict]	Specifies permission for specific user, group, or trustee. Valid options read, write, and full. This is a list of dictionaries. Each dictionary entry has 3 mandatory values: <ol style="list-style-type: none"> user_name/ group_name/ wellknown can have actual name of the trustee like user/ group/wellknown Permissions can be read/write/full permission_type can be allow/deny 	Optional		Permission: - user_name: "{{system_az_user}}" permission: "full" permission_type: "allow" provider_type: "ads" - group_name: "{{system_az_group}}" permission: "read" permission_type: "allow" provider_type: "file" - wellknown: "everyone" permission: "read"

Table 7. Parameters(continued)

Parameter name	Type	Explanation	Mandatory/ Optional	Default	Comments
		<p>4. provider_type can be local, filesystem, ads, or ldap.</p> <p>The fourth entry provider_type is optional (default is local) and applicable to user and group only.</p>			permission_type: "allow"
state	str	State of the SMB share. Options are: [present, absent]	Mandatory		
new_share_name	str	New name of the SMB share			
access_based_enumeration	bool	Only enumerate files and folders the requesting user has access to.	Optional	False	
access_based_enumeration_root_only	bool	Access-based enumeration on only the root directory of the share.	Optional	False	
browsable	bool	Share is visible in net view and the browse list.	Optional	False	
ntfs_acl_support	bool	Support NTFS ACLs on files and directories.	Optional	True	
directory_create_mask	str	Directory create mask bits.	Optional	System AZ: 700 (octal) Non-system AZ: 777 (octal)	Octal value for owner, group, others vs read, write, execute
directory_create_mode	str	Directory create mode bits.	Optional	System AZ: 0 Non-system AZ: 777 (octal)	Octal value for owner, group, others vs read, write, execute
file_create_mask	str	File create mask bits.	Optional	System AZ: 700 (octal) Non-system AZ: 700 (octal)	Octal value for owner, group, others vs read, write, execute
file_create_mode	str	File create mode bits.	Optional	System AZ: 100 (octal) Non-system AZ: 100 (octal)	Octal value for owner, group, others vs read, write, execute

Snapshot module

The snapshot module manages the snapshots available in Isilon.

The snapshot module supports the following functions:

- Create a filesystem snapshot.
- Get details of a filesystem snapshot.
- Modify a filesystem snapshot.
- Delete filesystem snapshot.

Create a snapshot for a Isilon filesystem

The user can create a snapshot of a filesystem by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create a filesystem snap on Isilon
  dellemc_isilon_snapshot:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{new_path_ansible_1}}"
    access_zone: "{{access_zone}}"
    snapshot_name: "{{snapshot_name}}"
    desired_retention: "{{desired_retention}}"
    retention_unit: "{{retention_unit_days}}"
    alias: "{{ansible_snap_alias}}"
    state: "{{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 a filesystem snapshot

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

This is an example of the syntax of a playbook:

```
- name: Get details of a filesystem snapshot
  dellemc_isilon_snapshot:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{new_path_ansible_1}}"
    access_zone: "{{access_zone}}"
    snapshot_name: "{{snapshot_name}}"
    state: "{{state_present}}"
```

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

Modify filesystem snapshot desired retention

The user can modify a snapshot desired retention by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify filesystem snapshot desired retention
  dellemc_isilon_snapshot:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
```

```
api_password: "{{api_password}}"
path: "{{new_path_ansible_1}}"
access_zone: "{{access_zone}}"
snapshot_name: "{{snapshot_name}}"
desired_retention: "{{desired_retention_new}}"
retention_unit: "{{retention_unit_days}}"
state: "{{state_present}}"
```

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

Modify expiration timestamp of the snapshot

The user can modify a snapshot expiration timestamp by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify filesystem snapshot expiration timestamp
  dellemc_isilon_snapshot:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    snapshot_name: "{{snapshot_name}}"
    expiration_timestamp: "{{expiration_timestamp_new}}"
    state: "{{present}}"
```

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

Modify filesystem snapshot alias

The user can modify a filesystem snapshot alias by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify filesystem snapshot alias
  dellemc_isilon_snapshot:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    snapshot_name: "{{snapshot_name}}"
    alias: "{{ansible_snap_alias_new}}"
    state: "{{state_present}}"
```

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

Delete snapshot alias

The user can delete a filesystem snapshot alias by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Delete snapshot alias
  dellemc_isilon_snapshot:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{new_path_ansible_1}}"
    access_zone: "{{access_zone}}"
    snapshot_name: "{{snapshot_name}}"
    alias: ""
    state: "{{state_present}}"
```


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

Rename filesystem snapshot

The user can rename a filesystem snapshot by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Rename filesystem snapshot
  dellemc_isilon_snapshot:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{new_path_ansible_1}}"
    access_zone: "{{access_zone}}"
    snapshot_name: "{{snapshot_name}}"
    new_snapshot_name: "{{new_snapshot_name}}"
    state: "{{state_present}}"
```

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

Delete filesystem snapshot

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

This is an example of the syntax of a playbook:

```
- name: Delete filesystem snapshot
  dellemc_isilon_snapshot:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{new_path_ansible_1}}"
    access_zone: "{{access_zone}}"
    snapshot_name: "{{new_snapshot_name}}"
    state: "{{state_absent}}"
```

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

Snapshot module parameters

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

Table 8. Parameters

Parameter name	Explanation	Mandatory/Optional	Default	Comments
snapshot_name	Name of the snapshot	Mandatory	Nil	
path	The path on which the snapshot will be taken	Optional	Nil	For non-system access zones, this path is relative to the base path of the access zone. For system access zones, it is the absolute path.
access_zone	the access zone	Optional	System	The effective path where the snapshot is created is determined by the base path of the access zone and the path provided by the user in the playbook.

Table 8. Parameters(continued)

Parameter name	Explanation	Mandatory/Optional	Default	Comments
new_snapshot_name	The new name of the snapshot	Optional	Nil	This parameter is for renaming the snapshot.
expiration_timestamp	The timestamp on which the snapshot will expire (UNIX epoch format)	Optional	Nil	Either this or desired retention can be specified but not both.
desired_retention	The number of days for which the snapshot can be retained.	Optional	Nil	Either this or expiration timestamp can be specified but not both.
retention_unit	The retention unit for the snapshot.	Optional	Nil	Hours is the default. Choices : [hours, days]
alias	The alias for the snapshot.	Optional	Nil	User can remove the alias by specifying empty string ("").
state	The state of the snapshot on Isilon.	Mandatory	Nil	Choices: [present, absent]

Snapshot schedule module

The user can create, modify, delete, and get details of snapshot schedules.

To create a snapshot schedule, name, path, pattern, and schedule parameters are compulsory.

If desired_retention is not specified while creating snapshot schedule, the snapshot created with that schedule never expires.

Modification of path is not allowed through the Ansible module.

For a system access zone, the path has to be absolute. For non-system access zone, the path is relative.

The snapshot schedule module has the following functions:

- Create a snapshot schedule.
- Get the details of the snapshot schedule.
- Modify the snapshot schedule.
- Delete a snapshot schedule.

Create snapshot schedule

The user can create a snapshot schedule by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Create snapshot schedule
  dell EMC isilon_snapshotschedule:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    name: "{{name}}"
    access_zone: '{{access_zone}}'
    path: '{{path1}}'
    alias: "{{alias1}}"
    desired_retention: "{{desired_retention1}}"
    pattern: "{{pattern1}}"
    schedule: "{{schedule1}}"
    state: "{{state_present}}"
```

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

Get the details of the snapshot schedule

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

This is an example of the syntax of a playbook:

```
- name: Get details of snapshot schedule
  dellemc_isilon_snapshotschedule:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    name: "{{name}}"
    state: "{{state_present}}"
```

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

Modify the pattern of the snapshot schedule

The user can modify the pattern of a snapshot schedule by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify pattern of snapshot schedule on Isilon
  dellemc_isilon_snapshotschedule:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    name: "{{name}}"
    pattern: "{{pattern2}}"
    state: "{{state_present}}"
```

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

Modify the schedule of the snapshot schedule

The user can modify the schedule of a snapshot schedule by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify schedule of snapshot schedule
  dellemc_isilon_snapshotschedule:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    name: "{{new_name}}"
    schedule: "{{schedule2}}"
    state: "{{state_present}}"
```

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

Rename a snapshot schedule

The user can rename a snapshot schedule by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Rename snapshot schedule on Isilon
  dellemc_isilon_snapshotschedule:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
```

```
api_user: "{{api_user}}"
api_password: "{{api_password}}"
name: "{{name}}"
new_name: "{{new_name}}"
state: "{{state_present}}"
```

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

Modify the alias of a snapshot schedule

The user can modify the alias of a snapshot schedule by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify alias of snapshot schedule on Isilon
  dellemc_isilon_snapshotschedule:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    name: "{{new_name}}"
    alias: "{{alias2}}"
    state: "{{state_present}}"
```

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

Modify the retention of a snapshot schedule

The user can modify the retention of a snapshot schedule by running the appropriate playbook.

This is an example of the syntax of a playbook:

```
- name: Modify retention of snapshot schedule
  dellemc_isilon_snapshotschedule:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    name: "{{new_name}}"
    desired_retention: 2
    retention_unit: "{{retention_unit_days}}"
    state: "{{state_present}}"
```

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

Delete a snapshot schedule

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

This is an example of the syntax of a playbook:

```
- name: Delete snapshot schedule on Isilon
  dellemc_isilon_snapshotschedule:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    name: "{{new_name}}"
    state: "{{state_absent}}"
```

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

Snapshot schedule module parameters

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

Table 9. Parameters

Parameter name	Explanation	Mandatory/Optional	Default	Comments
name	Name of the snapshot schedule	Mandatory		
path	The path on which the snapshot is taken	Optional		This path is relative to the base path of the access zone.
access_zone	The access zone	Optional	System	The effective path for the snapshot schedule is determined by the base path of the access zone and the path provided by the user in the playbook. The effective path where the snapshot is created is determined by the base path of the access zone and the path provided by the user in the playbook.
new_name	New name of the snapshot schedule.	Optional		Used for rename operation.
pattern	Pattern expanded with strftime to create snapshot names	Optional		Mandatory while creating the snapshot schedule. For example: If the pattern is "Demo_%Y-%m-%d_%H:%M", the snapshot that is generated is named "Demo_2020-01-17_12:00".
schedule	The isodate compatible natural language description of the schedule It specifies the frequency of the schedule.	Optional		Mandatory while creating the snapshot schedule. <ul style="list-style-type: none">• To run schedule daily at 1.00 AM, the value of schedule is every 1 day at 1:00 AM.• To run schedule daily every 4 hours, the value of schedule is every day every 4 hours between 12:00 AM and 11:59 PM.
desired_retention	The number of days/ hours for which snapshot is retained	Optional		
retention_unit	The retention unit for the snapshot	Optional	Hours	Hours is the default. Choices: [hours, days]
alias	The alias for the snapshot	Optional		
state	State of the snapshot schedule	Mandatory		Choices: [present, absent]

Smart quotas module

Managing smart quotas on an Isilon system includes create, get, modify and delete quotas for user, group and directory.

The smart quotas module has the following functions:

- Create a quota
- Modify quota attributes (such as include_overheads, soft_limit_size, cap_unit)
- Get details of a quota
- Delete a quota

 **NOTE:** There are two quotas of each type per directory — one with snapshots included, and one without snapshots included.

Create a smart quota of user type

The user can create a smart quota of user type by running the appropriate playbook.

This is an example of the syntax of a playbook where a smart quota is created of user type

```
- name: Create a Smart Quota
  dellemc_isilon_smartquota:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{path}}"
    quota_type: "user"
    user_name: "{{user_name}}"
    access_zone: "{{access_zone}}"
    provider_type: "{{provider_type}}"
    quota:
      include_snapshots: True
      include_overheads: False
      advisory_limit_size: 3
      soft_limit_size: 5
      soft_grace_period: 5
      period_unit: 'weeks'
      hard_limit_size: 6
      cap_unit: 'TB'
      state: "present"
```

This will create a user quota with grace period of 5 weeks, with snapshots included in the quota.

The parameters must be set before running the playbook. See the [Parameter Table](#) for more information about the parameters.

Create a smart quota of a directory

The user can create a smart quota of a directory type by running the appropriate playbook.

This is an example of the syntax of a playbook where a smart quota is created of a directory.

```
- name: Create a Quota for a Directory
  dellemc_isilon_smartquota:
    onefs_host: "{{onefs_host}}"
    verify_ssl: "{{verify_ssl}}"
    api_user: "{{api_user}}"
    api_password: "{{api_password}}"
    path: "{{path}}"
    quota_type: "directory"
    access_zone: "{{access_zone}}"
    quota:
      include_snapshots: False
      include_overheads: False
      advisory_limit_size: 3
      soft_limit_size: 5
      soft_grace_period: 5
      period_unit: 'weeks'
      hard_limit_size: 6
      cap_unit: 'TB'
      state: "present"
```

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

Modify a quota

The user can modify all parameters mentioned in the quota dictionary (except `include_snapshots`) by running the appropriate playbook.

This is an example syntax of a playbook where the soft limit and grace period of a quota is modified for a local user of a system access zone:

```
- name: Modify a grace period of Local User Quota of System Access Zone
dellemc_isilon_smartquota:
  onefs_host: "{{onefs_host}}"
  verify_ssl: "{{verify_ssl}}"
  api_user: "{{api_user}}"
  api_password: "{{api_password}}"
  path: "{{path}}"
  quota_type: "user"
  user_name: "{{user_name}}"
  quota:
    include_snapshots: True
    soft_limit_size: 5
    soft_grace_period: 3
    period_unit: 'days'
    cap_unit: 'TB'
  state: "present"
```

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

Get details of a quota

User can get the details of all types of quotas by running the appropriate playbook.

This is an example syntax of a playbook where the details of a quota for a local group of a system access zone is requested:

```
- name: Get details of a Local Group Quota of System Access Zone
dellemc_isilon_smartquota:
  onefs_host: "{{onefs_host}}"
  verify_ssl: "{{verify_ssl}}"
  api_user: "{{api_user}}"
  api_password: "{{api_password}}"
  path: "{{path}}"
  quota_type: "group"
  group_name: "{{group_name}}"
  state: "present"
```

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

Delete a quota

The user can delete all types of quotas by running the appropriate playbook.

This is an example syntax of a playbook where directory quota is deleted:

```
- name: Delete Quota for a directory
dellemc_isilon_smartquota:
  onefs_host: "{{onefs_host}}"
  verify_ssl: "{{verify_ssl}}"
  api_user: "{{api_user}}"
  api_password: "{{api_password}}"
  path: "{{path}}"
  quota_type: "directory"
  quota:
    include_snapshots: True
  state: "absent"
```

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

Smart Quotas module parameters

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

Table 10. Parameters

Parameter name	Explanation	Mandatory/Optional	Format	Default	Comments
path	Path for which a quota exists.	Mandatory	str		For System Access Zone the Absolute Path must be mentioned. For Non System Access Zone the path relative to the access zone's base path has to be mentioned
quota_type	Type of a quota. choices: ['directory', 'user', 'group']	Mandatory	str		
user_name	The name of the user for which the quota exists.	Optional	str		For performing operations on a user quota, user_name is mandatory.
group_name	The name of the group for which the quota exists.	Optional	str		For performing operations on a group quota, group_name is mandatory.
access_zone	Access Zone in which user/group/directory exists	Optional	str	" system "	If not mentioned then the access_zone will be assumed as " system ".
provider_type	Type of authentication used to authenticate user/group. choices: ['local', 'file', 'ldap', 'ads']	Optional	str	" local "	If not mentioned then the access_zone will be assumed as " local ".
quota: include_snapshots include_overheads advisory_limit_size soft_limit_size soft_grace_period period_unit hard_limit_size cap_unit	The quota dictionary includes the following attributes: <ul style="list-style-type: none"> include_snapshots: <ul style="list-style-type: none"> Whether to include snapshot's storage space in the quota calculations or not. default_value: False Type: bool include_overheads: <ul style="list-style-type: none"> Whether to include the metadata and protection overhead in the quota or not During quota creation if not passed then it will be taken as false. 	Optional	dictionary		<ul style="list-style-type: none"> include_snapshots if not passed then all operations will be performed taking it as False. The soft_limit_size and soft_grace_period should be mentioned together in a task. advisory_limit_size, soft_limit_size and hard_limit_size can be assigned together or separately. Cap_unit has to be passed if any of the advisory, soft or hard threshold provided.

Table 10. Parameters(continued)

Parameter name	Explanation	Mandatory/ Optional	Form at	Default	Comments
	<ul style="list-style-type: none"> ◦ type: bool • advisory_limit_size: <ul style="list-style-type: none"> ◦ The threshold value after which the advisory notification will be sent. ◦ type: int • soft_limit_size: <ul style="list-style-type: none"> ◦ threshold value after which soft_limit exceeded notification will be sent and soft_grace period will start. ◦ type: int • soft_grace_period: <ul style="list-style-type: none"> ◦ time period after which the write access for the quota will be restricted. ◦ type: int • period_unit: <ul style="list-style-type: none"> ◦ unit of the time period for soft_grace_period ◦ type: str ◦ choices: ['days', 'weeks', 'months'] • hard_limit_size: <ul style="list-style-type: none"> ◦ threshold value after which hard_limit exceeded notification will be sent. ◦ type: int • cap_unit: <ul style="list-style-type: none"> ◦ unit of storage for the hard, soft and advisory limits ◦ type: str ◦ choices: ['GB', 'TB'] 				
state	<p>The state of the quota after the task is performed.</p> <p>Choices: ['present', 'absent']</p>	Mandatory	str		<p>To delete the quota, state must be 'absent'.</p> <p>For all other operations, state must be 'present'.</p>