Dell EMC OpenManage Ansible Modules

Version 2.0.13 User's Guide

May 2020
Rev. A00

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.

Dell EMC OpenManage Ansible Modules

Version 2.0.13

Contents

1

Overview

Dell EMC OpenManage Ansible Modules allows data center and IT administrators to use RedHat Ansible to automate and orchestrate the configuration, deployment, and update of Dell EMC PowerEdge Servers (12th generation of PowerEdge servers and later) and modular infrastructure by leveraging the management automation capabilities in-built into the Integrated Dell Remote Access Controller (iDRAC) and OpenManage Enterprise (OME) respectively.

With the latest release of Dell EMC OpenManage Ansible Modules, the capabilities have improved with support for OpenManage Enterprise. OpenManage Ansible Modules simplifies and automates provisioning, deployment, and updates of PowerEdge servers and modular infrastructure. It allows system administrators and software developers to introduce the physical infrastructure provisioning into their software provisioning stack, integrate with existing DevOps pipelines and manage their infrastructure using version-controlled playbooks, server configuration profiles, and templates in line with the **Infrastructure-as-Code** (IaC) principles.

This user guide provides information about using **Dell EMC OpenManage Ansible Modules** and its different use cases.

The latest stable version of OpenManage Ansible Modules is available at dell.com/support. In addition to dell.com/support, you can download Ansible modules from https://github.com/dell/dellemc-openmanage-ansible-modules. Dell EMC supports modules that are downloaded from this GitHub location only.

Topics:

Key Features

The key features in OpenManage Ansible Modules are:

- Support for creating, modifying or deleting a user account.
- Perform the supported power state management operations on devices managed by OME.
- Support for creating, modifying or deploying a template.
- Get the list and details of all user accounts or of a specific account.
- Get the list and details of templates or of a specific template.
- Support for firmware update of PowerEdge devices and all its components.
- Support for retrieving job details for a given job ID or the entire job queue.
- Support for retrieving the list of all devices with the exhaustive inventory of each device.
- Export a server configuration profile (SCP) containing either the entire server configuration or component level configuration (iDRAC, BIOS, RAID, NIC) to a local file path on Ansible controller or a remote network share.
- Import an SCP from a local file path on Ansible controller or a remote network share.
- Support for configuration of BIOS, Integrated Dell Remote Access Controller (iDRAC), NIC, and RAID.
- Support for firmware update using a Firmware Repository hosted on a remote network share.
- Support for viewing firmware inventory details.
- Support for Windows, Linux, and ESXi operating system deployments.
- Support for configuring power controls, resetting iDRAC, viewing Lifecycle Controller (LC) job status, deleting LC job, deleting LC job queue, exporting LC logs, and configuring system lockdown mode.
- Retrieve the system inventory details.

NOTE: These features are supported only on iDRAC with enterprise license.

What's new?

The domain user authentication issue using a CIFS share in the dellemc_export_lc_logs module is fixed.

2

Getting Started

How OpenManage Ansible Modules works

- How OpenManage Ansible Modules work with iDRAC
- How OpenManage Ansible Modules work with OME
- How OpenManage Ansible Modules work with Redfish APIs

Running your first Playbook

Playbooks are essentially sets of instructions (plays) that you send to run on a single target or groups of targets (hosts).

To see how to run your first iDRAC and OME playbooks, see the following:

- Running your first iDRAC Playbook
- Running your first OME Playbook

3

Modules for iDRAC

How OpenManage Ansible Modules work with iDRAC

OpenManage Ansible modules uses iDRAC REST APIs based on Redfish standards and Server Configuration Profiles (SCP) for automated configuration, deployment and update of PowerEdge servers. An SCP contains all BIOS, iDRAC, Network and Storage settings of a PowerEdge server. You can apply them to multiple servers, enabling rapid, reliable, and reproducible configuration.

You can perform an SCP operation using any of the following methods:

- Export to or import from a remote network share via CIFS, NFS. Ensure that the remote network share is mounted on the Ansible controller with read-write privileges for user running the Ansible playbooks.
- Export or import via local file streaming (for iDRAC firmware 2.60.60.60 and above).

Setting up a local mount point for a remote network share

Mount the remote network share (CIFS or NFS) locally on the Ansible controller where you want to run the playbook or modules. Local mount point should have read-write privileges in order for OpenManage Ansible modules to write an SCP file to remote network share that will be imported by iDRAC.

NOTE: Refer to Linux man pages for mounting an NFS or CIFS network share on Ansible control machine.

Running your first iDRAC Playbook

Before you run a playbook to manage your iDRACs, you need to have a valid inventory of target PowerEdge servers. For more information on inventory, see Ansible documentation.

1. Install OpenManage Ansible Modules either from the dell.com/support or the https://github.com/dell/dellemc-openmanage-ansible-modules.git repository. For more details, see *Dell EM C OpenManage Ansible Modules Installation Guide*.
2. Create an inventory file containing a list of the iDRACs. In the following inventory example, we are using the inventory variables to store the iDRAC IP addresses and the user credentials. For more information on variables, see Ansible documentation.

```
inventory:

[PowerEdge]
R740.example.com
idrac_ip='192.168.10.10'
idrac_user='root'
idrac_password='idrac_password'
```

3. Define a playbook to fetch the hardware inventory of the servers. Create the playbook in the same directory where you created the inventory. Following is a playbook example:

```
playbook.yml

---
- hosts: PowerEdge
  connection: local
  gather_facts: False

  tasks:
  - name: Get hardware inventory
    dellemc_get_system_inventory:
      idrac_ip:  "{{ idrac_ip }}"
      idrac_user:  "{{ idrac_user }}"
      idrac_password:  "{{ idrac_password }}"
```

4. Now run the playbook. Run the following command from the directory where you created the inventory and the playbook:

```
ansible-playbook playbook.yml -i inventory
```

5. Press **Enter**.

With OpenManage Ansible Modules, you can construct a playbook with a set of modules resulting in an automation workflow for configuration, deployments, and updates of PowerEdge servers.

To view the list of all available iDRAC modules:

1. Run the following command on the Ansible control machine:

```
ansible-doc -l | grep "idrac"
```

2. Press **Enter**.

List of the available iDRAC modules is displayed.

To view the documentation of a module:

1. Run the following command on the Ansible control machine:

```
ansible-doc <module name>
```

2. Press **Enter**.

Updating Firmware

You can maintain up-to-date firmware versions of Dell EMC server components to get better efficiency, security protection and enhanced features. Create update sources to do the firmware update.

Following are the tasks for the firmware update activities:

- View firmware inventory
- Install firmware

View firmware inventory

Command: dellemc_get_firmware_inventory

Synopsis

You can view the firmware inventory of a server using this module. This module displays components of a server and the corresponding firmware versions.

Check_mode support: No

Options

Table 1. dellemc_get_firmware_inventory

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_username	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port

Table 2. Return Values

Name	Description	Returned	Type	Sample
Firmware Inventory	· Components of a server and their firmware versions.	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_get_firmware_inventory.md

Table 2. Return Values

Name	Description	Returned	Type	Sample
	· List of dictionaries, 1 dictionary per firmware.			

Examples

```
-name: Get Installed Firmware Inventory
   dellemc_get_firmware_inventory:
     idrac_ip:   "xx.xx.xx.xx"
     idrac_user: "xxxx"
     idrac_password:  "xxxxxxxx"
```

Install firmware

Module: idrac_firmware

Synopsis: The firmware can be installed from a repository on a network share (CIFS, NFS, HTTP, HTTPS, FTP) to keep the system updated. To install the firmware, connect to a network share that contains a valid repository of Dell Update Packages (DUPs), and a catalog file describing the DUPs.

Check_mode support: No

Options

Table 3. idrac_firmware

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
job_wait	Yes	True	NA	Provides the option to wait for job completion.
catalog_file_name	No	Catalog.xml	NA	Catalog file name relative to the `I(share_name)`.
reboot	No	False	NA	Provides the option to reboot after the updates have been applied.
share_name	Yes	NA	NA	Network share path of update repository. CIFS, NFS, HTP, HTTPS and FTS share types are supported.
share_user	No	NA	NA	User name required to access the network share must be provided as either 'user@domain' or 'domain\user'. This option is mandatory for CIFS network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.

Table 3. idrac_firmware(continued)

Parameter/aliases	Required	Default	Choices	Comments
share_mnt	Yes	NA	NA	Local mount path of the network share with read/write permission for the Ansible user.
ignore_cert_warning	No	True	NA	Specifies if certificate warnings are ignored when HTTPS share is used. If C(True) option is set, then the certificate warnings are ignored.
apply_update	No	True	NA	If I(apply_update) is set to C(True), then packages are applied. If set to C(False), packages are not applied.

Return Values

```
msg:
            type: str
            description: Overall firmware update status.
            returned: always
            sample: "Successfully updated the firmware."
update_status:
            type: dict
            description: Firmware Update job and progress details from the iDRAC.
            returned: success
            sample: {
                   'InstanceID': 'JID_XXXXXXXXXXXX',
                   'JobState': 'Completed',
                   'Message': 'Job completed successfully.',
                   'MessageId': 'REDXXX',
                   'Name': 'Repository Update',
                   'JobStartTime': 'NA',
                   'Status': 'Success',
                   }
```

Example

```
  - name: Update firmware from a repository on a NFS Share.
  idrac_firmware:
       idrac_ip: "192.168.0.1"
       idrac_user: "user_name"
       idrac_password: "user_password"
       share_name: "192.168.0.0:/share"
       reboot: True
       job_wait: True
       apply_update: True
       catalog_file_name: "Catalog.xml"

- name: Update firmware from a repository on a CIFS Share.
  idrac_firmware:
       idrac_ip: "192.168.0.1"
       idrac_user: "user_name"
       idrac_password: "user_password"
       share_name: "\\\\192.168.0.0\\cifs"
       share_user: "share_user"
       share_password: "share_password"
       share_mnt: "/mnt_path"
       reboot: True
       job_wait: True
       apply_update: True
       catalog_file_name: "Catalog.xml"

- name: Update firmware from a repository on a HTTP share.
  idrac_firmware:
```

```
        idrac_ip: "192.168.0.1"
        idrac_user: "user_name"
        idrac_password: "user_password"
        share_name: "http://downloads.dell.com"
        reboot: True
        job_wait: True
        apply_update: True

- name: Update firmware from a repository on a HTTPS share.
  idrac_firmware:
        idrac_ip: "192.168.0.1"
        idrac_user: "user_name"
        idrac_password: "user_password"
        share_name: "https://downloads.dell.com"
        reboot: True
        job_wait: True
        apply_update: True

- name: Update firmware from a repository on a FTP share.
  idrac_firmware:
        idrac_ip: "192.168.0.1"
        idrac_user: "user_name"
        idrac_password: "user_password"
        share_name: "ftp://ftp.dell.com"
        reboot: True
        job_wait: True
        apply_update: True
```

Module: dellemc_idrac_firmware

Synopsis: You can install the firmware from a repository on a network share (CIFS, NFS) to keep the system updated.

To install the firmware:

- Ensure that the network share contains a valid repository of Dell Update Packages (DUPs) and a catalog file that consists the latest DUPs.
- All applicable updates that are contained in the repository are applied to the system.

Check_mode support: No

NOTE: This module is deprecated and replaced with idrac_firmware.

Options

Table 4. dellemc_idrac_firmware

Parameter	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
job_wait	Yes	True	NA	Whether to wait for job completion or not.
catalog_file_name	No	Catalog.xml	NA	Catalog file name relative to the `I(share_name)`.
reboot	No	False	NA	Whether to reboot after applying the updates or not.
share_name	Yes	NA	NA	CIFS or NFS Network share
share_user	No	NA	NA	Network share user in the format 'user@domain' or

Table 4. dellemc_idrac_firmware(continued)

Parameter	Required	Default	Choices	Comments
				'domain\\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	Yes	NA	NA	Local mount path of the network share with read/write permission for ansible user This option is mandatory for Network share.

Return Values

```
msg:
            type: str
            description: Over all firmware update status.
            returned: always
            sample: "Successfully updated the firmware."
update_status:
            type: dict
            description: Firmware Update job and progress details from the iDRAC.
            returned: success
            sample: {
                   'InstanceID': 'JID_XXXXXXXXXXXX',
                   'JobState': 'Completed',
                   'Message': 'Job completed successfully.',
                   'MessageId': 'REDXXX',
                   'Name': 'Repository Update',
                   'JobStartTime': 'NA',
                   'Status': 'Success',
                   }
```

Example

```
- name: Update firmware from repository on a Network Share
    dellemc_idrac_firmware:
      idrac_ip:          "192.168.0.1"
      idrac_user:        "user_name"
      idrac_pwd:         "user_pwd"
      share_name:        "192.168.0.0:/share"
      share_user:        "share_user_name"
      share_pwd:         "share_user_pwd"
      share_mnt:         "/mnt/share"
      reboot:             True
      job_wait:           True
      catalog_file_name: "Catalog.xml"
```

Module: dellemc_install_firmware

Synopsis

You can install the firmware from a repository on a network share (CIFS, NFS) to keep the system updated.

- For 12th and 13th generation of PowerEdge servers, firmware update from a network repository is performed using WS-Man APIs.
- For 14th generation of PowerEdge servers, firmware update from a network repository is performed using the SCP.

To install the firmware:

- Ensure that the network share contains a valid repository of Dell Update Packages (DUPs) and a catalog file that consists the latest DUPs.

- All applicable updates that are contained in the repository are applied to the system.

Check_mode support: No

NOTE: This module is deprecated and replaced with idrac_firmware.

Options

Table 5. dellemc_install_firmware

Parameter	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
job_wait	Yes	True	NA	Whether to wait for job completion or not.
catalog_file_name	No	Catalog.xml	NA	Catalog file name relative to the `I(share_name)`.
reboot	No	False	NA	Whether to reboot after applying the updates or not.
share_name	Yes	NA	NA	CIFS or NFS Network share
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	Yes	NA	NA	Local mount path of the network share with read/write permission for ansible user. This option is mandatory for Network share.

Table 6. Return Values

Name	Description	Returned	Type	Sample
Firmware	Updates firmware from a repository on a network share (CIFS, NFS).	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_install_firmware.md

Example

```
-name: Update firmware from a repository on a Network Share
   dellemc_install_firmware:
     idrac_ip:           "xx.xx.xx.xx"
     idrac_user:         "xxxx"
     idrac_pwd:          "xxxxxxxx"
```

```
      share_name:          "xx.xx.xx.xx:/share"
      share_user:          "xxxx"
      share_pwd:           "xxxxxxxx"
      share_mnt:           "/mnt/share"
      reboot:              "True"
      job_wait:            "True"
      catalog_file_name:   "Catalog.xml"
```

Configuring PowerEdge Servers

Integrated Dell Remote Access Controller (iDRAC) with LC provide the ability to generate a human-readable representation of server configuration using Server Configuration Profile (SCP) feature. This file contains BIOS, iDRAC, LC, network, and RAID configuration settings. You can modify this file as per your need and apply to other servers.

The SCP feature is used in the Ansible module to automate the configuration activity of PowerEdge servers and their components.

Following are the tasks:

- View LC status
- Server Configuration Profile
- Configuring iDRAC
- Configure BIOS
- Configure RAID
- Configure Collect System Inventory on Restart
- Configure syslog

View LC status

Module: dellemc_get_lcstatus

Synopsis

You can view the LC status on a PowerEdge server using this module. You must check the readiness of the LC before carrying out any configuration or update. This module returns the LC readiness as True or False and its status.

Check_mode support: No

Options

Table 7. dellemc_get_lcstatus

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port

Table 8. Return Values

Name	Description	Returned	Type	Sample
LC status	Displays the LC status on a PowerEdge server	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_get_lcstatus.md

Example

```
-name: Get LC Status
   dellemc_get_lcstatus:
     idrac_ip:   "xx.xx.xx.xx"
```

```
    idrac_user: "xxxx"
    idrac_password:  "xxxxxxxx"
```

Server Configuration Profile

Export or Import Server Configuration Profile

Module: idrac_server_config_profile

Synopsis

This module exports the Server Configuration Profile (SCP) from iDRAC. It can also import from a network share or from a local file.

Options

Table 9. idrac_server_config_profile

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
command	No	export	· import · export	· If C(import), will perform SCP import operations. · If C(export), will perform SCP export operations.
job_wait	Yes	NA	NA	Whether to wait for job completion or not.
share_name	Yes	NA	NA	CIFS or NFS Network Share or a local path.
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\\user' if user is a part of a domain, else 'user'. This option is mandatory for CIFS Network Share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network Share.
scp_file	No	NA	NA	SCP file name. This option is mandatory for C(import) state.
scp_components	No	ALL	· ALL · IDRAC · BIOS · NIC · RAID	· If C(ALL), the module imports all components configurations from SCP file. · If C(iDRAC), the module imports iDRAC configuration from SCP file. · If C(BIOS), the module imports BIOS configuration from SCP file. · If C(NIC), the module imports NIC configuration from SCP file. · If C(RAID), the module imports RAID configuration from SCP file.
shutdown_type	No	Graceful	· Graceful · Forced · NoReboot	This option is applicable for C(import) state. · If C(Graceful), it gracefully shuts down the server · If C(Forced), it forcefully shuts down the system

Table 9. idrac_server_config_profile(continued)

Parameter/aliases	Required	Default	Choices	Comments
				• If C(NoReboot), it does not reboot the server
end_host_power_state	No	On	• On • Off	This option is applicable for C(import) state. • If C(On), End host power state is on • If C(Off), End host power state is off
export_format	No	XML	• JSON • XML	Specify the output file format. This option is applicable for C(export) state.
export_use	No	Default	• Default • Clone • Replace	Specify the type of SCP to be exported. This option is applicable for C(export) state.

Return Values

```
msg:
  type: str
  description: status of the import or export SCP job.
  returned: always
  sample: "Successfully imported the Server Configuration Profile"
scp_status:
  type: dict
  description: SCP operation job and progress details from the iDRAC.
  returned: success
  sample:
    {
      "Id": "JID_XXXXXXXXX",
      "JobState": "Completed",
      "JobType": "ImportConfiguration",
      "Message": "Successfully imported and applied Server Configuration Profile.",
      "MessageArgs": [],
      "MessageId": "XXX123",
      "Name": "Import Configuration",
      "PercentComplete": 100,
      "StartTime": "TIME_NOW",
      "Status": "Success",
      "TargetSettingsURI": null,
      "retval": true
    }
```

Examples

```
- name: Import SCP from a network share and wait for this job to get completed.
  dellemc_idrac_server_config_profile:
    idrac_ip:     "192.168.0.1"
    idrac_user:   "user_name"
    idrac_password:"user_password"
    command:      "import"
    share_name:   "192.168.0.2:/share"
    share_user:   "share_user_name"
    share_password:"share_user_password"
    scp_file:     "scp_filename.xml"
    scp_components:"ALL"
    job_wait:      True


- name: Import SCP from a local path and wait for this job to get completed.
  dellemc_idrac_server_config_profile:
    idrac_ip:     "192.168.0.1"
    idrac_user:   "user_name"
    idrac_password:"user_password"
    command:      "import"
    share_name:   "/scp_folder"
    share_user:   "share_user_name"
```

```
    share_password:"share_user_password"
    scp_file:      "scp_filename.xml"
    scp_components:"ALL"
    job_wait:       True


- name: Export SCP to a network share
  dellemc_idrac_server_config_profile:
    idrac_ip:       "192.168.0.1"
    idrac_user:     "user_name"
    idrac_password: "user_password"
    share_name:     "192.168.0.2:/share"
    share_user:     "share_user_name"
    share_password: "share_user_password"
    job_wait:       False


- name: Export SCP to a local path
  dellemc_idrac_server_config_profile:
    idrac_ip:       "192.168.0.1"
    idrac_user:     "user_name"
    idrac_password:"user_password"
    share_name:     "/scp_folder"
    share_user:     "share_user_name"
    share_password:"share_user_password"
    job_wait:      False
```

Module: dellemc_idrac_server_config_profile

Synopsis

This module exports Server Configuration profile (SCP) to a given network share or imports SCP from a network share or a local file.

NOTE: This module is deprecated and replaced with idrac_server_config_profile.

Options

Table 10. dellemc_idrac_server_config_profile

Parameter	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
command	No	export	· import · export	· If C(import), will perform SCP import operations. · If C(export), will perform SCP export operations.
job_wait	Yes	NA	NA	Whether to wait for job completion or not.
share_name	Yes	NA	NA	CIFS or NFS Network Share or a local path.
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\\user' if user is a part of a domain, else 'user'. This option is mandatory for CIFS Network Share.
share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network Share.
scp_file	No	NA	NA	Server Configuration Profile file name. This option is mandatory for C(import) state.
scp_components	No	ALL	· ALL · IDRAC	· If C(ALL), the module imports all components configurations from SCP file.

Table 10. dellemc_idrac_server_config_profile(continued)

Parameter	Required	Default	Choices	Comments
			• BIOS • NIC • RAID	• If C(iDRAC), the module imports iDRAC configuration from SCP file. • If C(BIOS), the module imports BIOS configuration from SCP file. • If C(NIC), the module imports NIC configuration from SCP file. • If C(RAID), the module imports RAID configuration from SCP file.
shutdown_type	No	Graceful	• Graceful • Forced • NoReboot	This option is applicable for C(import) state. • If C(Graceful), it gracefully shuts down the server • If C(Forced), it forcefully shuts down the system • If C(NoReboot), it does not reboot the server
end_host_power_state	No	On	• On • Off	This option is applicable for C(import) state. • If C(On), End host power state is on • If C(Off), End host power state is off
export_format	No	XML	• JSON • XML	Specify the output file format. This option is applicable for C(export) state.
export_use	No	Default	• Default • Clone • Replace	Specify the type of Sever Configuration Profile (SCP) to be exported. This option is applicable for C(export) state.

Return Values

```
msg:
  type: str
  description: status of the import or export SCP job.
  returned: always
  sample: "Successfully imported the Server Configuration Profile"
scp_status:
  type: dict
  description: SCP operation job and progress details from the iDRAC.
  returned: success
  sample:
    {
      "Id": "JID_XXXXXXXXX",
      "JobState": "Completed",
      "JobType": "ImportConfiguration",
      "Message": "Successfully imported and applied Server Configuration Profile.",
      "MessageArgs": [],
      "MessageId": "XXX123",
      "Name": "Import Configuration",
      "PercentComplete": 100,
      "StartTime": "TIME_NOW",
      "Status": "Success",
      "TargetSettingsURI": null,
      "retval": true
    }
```

Examples

```
- name: Import Server Configuration Profile from a network share
  dellemc_idrac_server_config_profile:
    idrac_ip: "192.168.0.1"
    idrac_user: "user_name"
    idrac_pwd: "user_pwd"
```

```
    command: "import"
    share_name: "192.168.0.2:/share"
    share_user: "share_user_name"
    share_pwd: "share_user_pwd"
    scp_file:   "scp_filename.xml"
    scp_components: "ALL"
    job_wait: True


- name: Import Server Configuration Profile from a local path
  dellemc_idrac_server_config_profile:
    idrac_ip: "192.168.0.1"
    idrac_user: "user_name"
    idrac_pwd: "user_pwd"
    command: "import"
    share_name: "/scp_folder"
    share_user: "share_user_name"
    share_pwd: "share_user_pwd"
    scp_file:   "scp_filename.xml"
    scp_components: "ALL"
    job_wait: True


- name: Export Server Configuration Profile to a network share
  dellemc_idrac_server_config_profile:
    idrac_ip: "192.168.0.1"
    idrac_user: "user_name"
    idrac_pwd: "user_pwd"
    share_name: "192.168.0.2:/share"
    share_user: "share_user_name"
    share_pwd: "share_user_pwd"
    job_wait: False


- name: Export Server Configuration Profile to a local path
  dellemc_idrac_server_config_profile:
    idrac_ip: "192.168.0.1"
    idrac_user: "user_name"
    idrac_password: "user_password"
    share_name: "/scp_folder"
    share_user: "share_user_name"
    share_pwd: "share_user_pwd"
    job_wait: False
```

Module: dellemc_import_server_config_profile

Synopsis

You can import an SCP file (in an XML or JSON format) exported from a golden PowerEdge server configuration to one or more servers, thus achieving an effortless, consistent, and automated deployment. Importing an SCP file is useful in restoring the configuration of the server to the state stored in the profile.

You can import SCP from a local or a remote share to iDRAC. For a remote share, make sure that a network share path and the file name are available. If there are component configurations (such as BIOS, RAID, NIC, iDRAC, and so on) present in the SCP file that require a server restart, you can use the **I(shutdown_type)** argument to specify whether a **Graceful** or **Forced** shutdown of the server is required.

Check_mode support: No

NOTE: This module is deprecated and replaced with idrac_server_config_profile.

Options

Table 11. dellemc_import_server_config_profile

Parameter	Required	Default	Choices	Comments
end_host_power_state	No	On	• On • Off	• If On, End host power is on • If Off, End host power is off
idrac_ip	Yes	NA	NA	iDRAC IP Address

Table 11. dellemc_import_server_config_profile(continued)

Parameter	Required	Default	Choices	Comments
idrac_user	Yes	NA	NA	iDRAC username
idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
job_wait	Yes	NA	• True • False	• If the value is True, it waits for the SCP import job to finish and returns the job completion status • If the value is False, it returns immediately with a JOB ID after queuing the SCP import job in LC job queue
scp_components	No	ALL	• ALL • iDRAC • BIOS • NIC • RAID	• If ALL, the module imports all components configurations from SCP file • If iDRAC, the module imports iDRAC configuration from SCP file • If BIOS, the module imports BIOS configuration from SCP file • If NIC, the module imports NIC configuration from SCP file • If RAID, the module imports RAID configuration from SCP file
scp_file	Yes	NA	NA	Server Configuration Profile file name
share_name	Yes	NA	NA	Network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
shutdown_type	No	Graceful	• Graceful • Forced • NoReboot	• If Graceful, it gracefully shuts down the server • If Forced, it forcefully shuts down the system • If NoReboot, it does not reboot the server

Table 12. Return Values

Name	Description	Returned	Type	Sample
Import SCP	Imports SCP from a network share or from a local file	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_import_server_config_profile.md

Example

```
-name: Import Server Configuration Profile
   dellemc_import_server_config_profile
     idrac_ip:       "xx.xx.xx.xx"
     idrac_user:     "xxxx"
     idrac_pwd:      "xxxxxxxx"
     share_name:     "xx.xx.xx.xx:/share"
     share_user:     "xxxx"
     share_pwd:      "xxxxxxxx"
     scp_file:       "scp_file.xml"
     scp_components: "ALL"
     job_wait:       "True"
```

Module: dellemc_export_server_config_profile

Synopsis

You can export **Server Configuration Profile (SCP)** with various components such as iDRAC, BIOS, NIC, RAID together or with one of these components. You can export SCP from iDRAC to a local or a network shared location. For shared location, make sure that a network share path is established.

Check_mode support: No

NOTE: This module is deprecated and replaced with idrac_server_config_profile.

Options

Table 13. dellemc_export_server_config_profile

Parameter	Required	Default	Choices	Comments
export_format	No	XML	• JSON • XML	The output file format
export_use	No	Default	• Default • Clone • Replace	• If C(Default), will export the SCP using the Default method • If C(Clone), will export the SCP using the Clone method • If C(Replace), will export the SCP using the Replace method
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
job_wait	Yes	NA	• True • False	• If the value is True, it waits for the SCP export job to finish and returns the job completion status • If the value is False, it returns immediately with a JOB ID after queuing the SCP export job in LC job queue
share_name	Yes	NA	NA	CIFS or NFS network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
scp_components	No	ALL	• ALL • IDRAC • BIOS • NIC • RAID	Specify the hardware components configuration to be exported • If ALL, the module exports all components configurations in SCP file • If IDRAC, the module exports iDRAC configuration in SCP file • If BIOS, the module exports BIOS configuration in SCP file • If NIC, the module exports NIC configuration in SCP file • If RAID, the module exports RAID configuration in SCP file

Table 14. Return Values

Name	Description	Returned	Type	Sample
Export SCP	Exports the SCP to the provided network share or to the local path	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_export_server_config_profile.md

Example

```
-name: Export Server Configuration Profile (SCP)
   dellemc_export_server_config_profile:
     idrac_ip:       "xx.xx.xx.xx"
     idrac_user:     "xxxx"
     idrac_pwd:      "xxxxxxxx"
     share_name:     "xx.xx.xx.xx:/share"
     share_user:     "xxxx"
     share_pwd:      "xxxxxxxx"
     export_format: "XML"
     export_use:     "Default"
     job_wait:       "True"
```

Configuring iDRAC

Following are the modules responsible for configuring specific iDRAC attributes.

- Configure iDRAC users
- Configure iDRAC timezone
- Configure iDRAC eventing
- Configure iDRAC services
- Configure iDRAC network

Configure iDRAC users

Module: dellemc_configure_idrac_users

Synopsis

This module creates, modifies or deletes an iDRAC local user.

Check_mode support: Yes

Options

Table 15. dellemc_configure_idrac_users

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS Network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This

Table 15. dellemc_configure_idrac_users(continued)

Parameter/aliases	Required	Default	Choices	Comments
				option is mandatory for CIFS or NFS Network share.
action	No	create	• create • delete • modify	This value decides whether to create or delete or modify iDRAC user
user_name	No	NA	NA	Provide the username to be created or deleted or modified
user_password	No	NA	NA	Provide the password for the user to be created or modified
privilege_users	No	NA	• NoAccess • Readonly • Operator • Administrator	Privilege user access is configurable
ipmilanprivilege_users	No	NA	• No_Access • Administrator • Operator • User	IPMI Lan Privilege user access is configurable
ipmiserialprivilege_users	No	NA	• No_Access • Administrator • Operator • User	IPMI Serial Privilege user access is configurable **NOTE: This parameter is not supported by PowerEdge Modular servers.**
enable_users	No	NA	• Enabled • Disabled	Enabling or Disabling the new iDRAC user
solenable_users	No	NA	• Enabled • Disabled	Enabling or Disabling SOL for iDRAC user
protocolenable_users	No	NA	• Enabled • Disabled	Enabling or Disabling protocol for iDRAC user
authenticationprotocol_users	No	NA	• T_None • SHA • MD5	Configuring authentication protocol for iDRAC user
privacyprotocol_users	No	NA	• T_None • DES • AES	Configuring privacy protocol for iDRAC user

Table 16. Return Values

Name	Description	Returned	Type	Sample
iDRAC users	Configures the iDRAC users attributes	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_configure_idrac_users.md

Example

```
-name: Configure the iDRAC users attributes
   dellemc_configure_idrac_users:
     idrac_ip:                    "xx.xx.xx.xx"
```

```
    idrac_user:                     "xxxx"
    idrac_password:                 "xxxxxxxx"
    share_name:                     "xx.xx.xx.xx:/share"
    share_password:                 "xxxxxxxx"
    share_user:                     "xxxx"
    share_mnt:                      "/mnt/share"
    action:                         "create"
    user_name:                      "username"
    user_password:                  "xxxxxxxx"
    privilege_users:                "Administrator"
    ipmilanprivilege_users:         "Administrator"
    ipmiserialprivilege_users:      "Administrator"
    enable_users:                   "Enabled"
    solenable_users:                "Enabled"
    protocolenable_users:           "Enabled"
    authenticationprotocol_users:   "SHA"
    privacyprotocol_users:          "AES"
```

Configure iDRAC timezone

Module: dellemc_configure_idrac_timezone

Synopsis

This module configures the iDRAC timezone related attributes.

Check_mode support: Yes

Options

Table 17. dellemc_configure_idrac_timezone

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS Network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This option is mandatory for CIFS or NFS Network share.
setup_idrac_timezone	No	NA	NA	Configuring the timezone for iDRAC
enable_ntp	No	NA	NA	Whether to Enable or Disable NTP for iDRAC
ntp_server_1	No	NA	NA	NTP configuration for iDRAC
ntp_server_2	No	NA	NA	NTP configuration for iDRAC
ntp_server_3	No	NA	NA	NTP configuration for iDRAC

Table 18. Return Values

Name	Description	Returned	Type	Sample
iDRAC Timezone	Configures the iDRAC timezone attributes	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_configure_idrac_timezone.md

Example

```
-name: Configure the iDRAC timezone attributes
   dellemc_configure_idrac_timezone:
     idrac_ip:              "xx.xx.xx.xx"
     idrac_user:            "xxxx"
     idrac_password:        "xxxxxxxx"
     share_name:            "xx.xx.xx.xx:/share"
     share_password:        "xxxxxxxx"
     share_user:            "xxxx"
     share_mnt:             "/mnt/share"
     setup_idrac_timezone: "UTC"
     enable_ntp:            "Enabled"
     ntp_server_1:          "x.x.x.x"
     ntp_server_2:          "x.x.x.x"
     ntp_server_3:          "x.x.x.x"
```

Configure iDRAC eventing

Module: dellemc_configure_idrac_eventing

Synopsis

This module configures iDRAC eventing related attributes.

Check_mode support: Yes

Options

Table 19. dellemc_configure_idrac_eventing

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS Network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This option is mandatory for CIFS or NFS Network share.
destination_number	No	None	NA	Destination number for SNMP Trap
destination	No	None	NA	Destination for SNMP Trap

Table 19. dellemc_configure_idrac_eventing(continued)

Parameter/aliases	Required	Default	Choices	Comments
snmp_v3_username	No	NA	NA	SNMP v3 username for SNMP Trap
snmp_trap_state	No	NA	· Enabled · Disabled	Whether to Enable or Disable SNMP alert
email_alert_state	No	NA	· Enabled · Disabled	Whether to Enable or Disable Email alert
alert_number	No	None	NA	Alert number for Email configuration
address	No	NA	NA	Email address for SNMP Trap
custom_message	No	NA	NA	Custom message for SNMP Trap reference
enable_alerts	No	NA	· Enabled · Disabled	Whether to Enable or Disable iDRAC alerts
authentication	No	NA	· Enabled · Disabled	Simple Mail Transfer Protocol Authentication
smtp_ip_address	No	NA	NA	SMTP IP address for communication
smtp_port	No	None	NA	SMTP Port number for access
username	No	None	NA	Username for SMTP authentication
password	No	None	NA	Password for SMTP authentication

Table 20. Return Values

Name	Description	Returned	Type	Sample
iDRAC eventing	Configures the iDRAC eventing attributes	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_configure_idrac_eventing.md

Example

```
-name: Configure the iDRAC eventing attributes
   dellemc_configure_idrac_eventing:
     idrac_ip:            "xx.xx.xx.xx"
     idrac_user:          "xxxx"
     idrac_password:      "xxxxxxxx"
     share_name:          "xx.xx.xx.xx:/share"
     share_password:      "xxxxxxxx"
     share_user:          "xxxx"
     share_mnt:           "/mnt/share"
     destination_number: "xxxx"
     destination:         "xxxx"
     snmp_v3_username:    "xxxx"
     snmp_trap_state:     "xxxx"
     email_alert_state:   "xxxx"
     alert_number:        "xxxx"
     address:             "xxxxxxxxxx"
     custom_message:      "xxxx"
     enable_alerts:       "xxxxxx"
     authentication:      "xxxxxx"
     smtp_ip_address:     "x.x.x.x"
     smtp_port:           "xxxx"
     username:            "xxxx"
     password:            "xxxxxxxx"
```

Configure iDRAC services

Module: dellemc_configure_idrac_services

Synopsis

This module configures the iDRAC services related attributes.

Check_mode support: Yes

Options

Table 21. dellemc_configure_idrac_services

Parameter	Required	Default	Choices	Comments
idrac_ip	True	NA	NA	iDRAC IP Address
idrac_user	True	NA	NA	iDRAC username
idrac_password/ idrac_pwd	True	NA	NA	iDRAC user password
idrac_port	False	443	NA	iDRAC port
share_name	True	NA	NA	CIFS or NFS Network share or a local path
share_user	False	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	False	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	False	NA	NA	Local mount path of the network share with read/write permission for Ansible user. This option is mandatory for CIFS or NFS Network share.
enable_web_server	False	NA	• Enabled • Disabled	Whether to Enable or Disable web server configuration for iDRAC
ssl_encryption	False	NA	• Auto_Negotiate • T_128_Bit_or_higher • T_168_Bit_or_higher • T_256_Bit_or_higher	Secure Socket Layer encryption for web server
tls_protocol	False	NA	• TLS_1_0_and_Higher • TLS_1_1_and_Higher • TLS_1_2_Only	Transport Layer Security for web server
https_port	False	NA	NA	HTTPS access port
http_port	False	NA	NA	HTTP access port
timeout	False	NA	NA	Timeout value
snmp_enable	False	NA	• Enabled • Disabled	Whether to Enable or Disable SNMP protocol for iDRAC
snmp_protocol	False	NA	• All • SNMPv3	Type of the SNMP protocol
community_name	False	test	NA	SNMP community name for iDRAC

Table 21. dellemc_configure_idrac_services(continued)

Parameter	Required	Default	Choices	Comments
alert_port	False	None	NA	SNMP alert port for iDRAC
discovery_port	False	162	NA	SNMP discovery port for iDRAC
trap_format	False	None	SNMPv1, SNMPv2 or SNMPv3	SNMP trap format for iDRAC

Table 22. Return Values

Name	Description	Returned	Type	Sample
iDRAC services	Configures the iDRAC services attributes	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_configure_idrac_services.md

Example

```
-name: Configure the iDRAC services attributes
   dellemc_configure_idrac_services:
     idrac_ip:          "xx.xx.xx.xx"
     idrac_user:        "xxxx"
     idrac_password:    "xxxxxxxx"
     share_name:        "xx.xx.xx.xx:/share"
     share_password:    "xxxxxxxx"
     share_user:        "xxxx"
     share_mnt:         "/mnt/share"
     enable_web_server: "Enabled"
     http_port:         "80"
     https_port:        "443"
     ssl_encryption:    "Auto_Negotiate"
     tls_protocol:      "TLS_1_2_Only"
     timeout:           "1800"
     snmp_enable:       "Enabled"
     snmp_protocol:     "SNMPv3"
     community_name:    "test"
     alert_port:        "None"
     discovery_port:    "162"
     trap_format:       "None"
```

Configure iDRAC network

Module: dellemc_configure_idrac_network

Synopsis

This module configures the iDRAC networking attributes.

Check_mode support: Yes

Options

Table 23. dellemc_configure_idrac_network

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS Network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part

Table 23. dellemc_configure_idrac_network(continued)

Parameter/aliases	Required	Default	Choices	Comments
				of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This option is mandatory for CIFS or NFS Network share.
setup_idrac_nic_vlan	No	NA	NA	Configuring the VLAN-related setting for iDRAC
register_idrac_on_dns	No	NA	• Enabled • Disabled	Registering Domain Name System for iDRAC
dns_idrac_name	No	NA	NA	DNS Name for iDRAC
auto_config	No	NA	• Enabled • Disabled	Automatically creates the records for DNS
static_dns	No	NA	NA	Static configuration for DNS
vlan_id	No	None	NA	Configuring the VLAN ID for iDRAC
vlan_priority	No	None	NA	Configuring the VLAN priority for iDRAC
enable_nic	No	NA	• Enabled • Disabled	Whether to Enable or Disable Network Interface Controller for iDRAC
nic_selection	No	NA	• Dedicated • LOM1 • LOM2 • LOM3 • LOM4	Selecting Network Interface Controller types for iDRAC
failover_network	No	NA	• ALL • LOM1 • LOM2 • LOM3 • LOM4 • T_None	Failover Network Interface Controller types for iDRAC
auto_detect	No	NA	• Enabled • Disabled	Auto detect Network Interface Controller types for iDRAC
auto_negotiation	No	NA	• Enabled • Disabled	Auto negotiation of Network Interface Controller for iDRAC
network_speed	No	NA	• T_10 • T_100 • T_1000	Network speed for Network Interface Controller types for iDRAC
duplex_mode	No	NA	• Full • Half	Transmission of data Network Interface Controller types for iDRAC
nic_mtu	No	None	NA	NIC Maximum Transmission Unit
ip_address	No	NA	NA	IP Address needs to be defined

Table 23. dellemc_configure_idrac_network(continued)

Parameter/aliases	Required	Default	Choices	Comments
enable_dhcp	No	NA	NA	Whether to Enable or Disable DHCP Protocol for iDRAC
dns_from_dhcp	No	NA	• Enabled • Disabled	Specifying Domain Name System from Dynamic Host Configuration Protocol
enable_ipv4	No	NA	• Enabled • Disabled	Whether to Enable or Disable IPv4 configuration
static_dns_1	No	NA	NA	Specify Domain Name System Configuration
static_dns_2	No	NA	NA	Specify Domain Name System Configuration
static_gateway	No	None	NA	Interfacing the network with another protocol
static_net_mask	No	None	NA	Determine whether IP address belongs to host

Table 24. Return Values

Name	Description	Returned	Type	Sample
iDRAC network	Configures the iDRAC network attributes	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_configure_idrac_network.md

Example

```
-name: Configure the iDRAC network attributes
   dellemc_configure_idrac_network:
   idrac_ip:               "xx.xx.xx.xx"
   idrac_user:             "xxxx"
   idrac_password:         "xxxxxxxx"
   share_name:             "xx.xx.xx.xx:/share"
   share_password:         "xxxxxxxx"
   share_user:             "xxxx"
   share_mnt:              "/mnt/share"
   register_idrac_on_dns: "Enabled'
   dns_idrac_name:         "None"
   auto_config:            "None"
   static_dns:             "None"
   setup_idrac_nic_vlan:  "Enabled"
   vlan_id:                "0"
   vlan_priority:          "1"
   enable_nic:             "Enabled"
   nic_selection:          "Dedicated"
   failover_network:       "T_None"
   auto_detect:            "Disabled"
   auto_negotiation:       "Enabled"
   network_speed:          "T_1000"
   duplex_mode:            "Full"
   nic_mtu:                "1500"
   ip_address:             "x.x.x.x"
   enable_dhcp:            "Enabled"
   dns_from_dhcp:          "Enabled"
   enable_ipv4:            "Enabled"
   static_dns_1:           "x.x.x.x"
   static_dns_2:           "x.x.x.x"
   static_gateway:         "None"
   static_net_mask:        "None"
```

Configure BIOS

Module: dellemc_configure_bios

Synopsis

This module configures the BIOS attributes for PowerEdge servers.

Check_mode support: Yes

Options

Table 25. dellemc_configure_bios

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	No	NA	NA	CIFS or NFS network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This option is mandatory for CIFS or NFS Network share.
boot_mode	No	NA	· Bios · Uefi	(deprecated) Configures the boot mode to BIOS or UEFI. **NOTE: This option has been deprecated, and will be removed in the later version. Please use the I(attributes) for BIOS attributes configuration instead.** **NOTE: I(boot_mode) is mutually exclusive with I(boot_sources).**
boot_sequence	No	NA	NA	(deprecated) Boot devices' FQDDs in the sequential order for BIOS or UEFI Boot Sequence. Provide the `I(boot_mode)` option to determine the appropriate boot sequence to be applied. **NOTE: This option has been deprecated, and will be removed in the later version. Please use the I(attributes) or I(boot_sources) for Boot Sequence modification instead.** **NOTE: I(boot_sequence) is mutually exclusive with I(boot_sources).**
nvme_mode	No	NA	· NonRaid · Raid	(deprecated) Configures the NVME mode in the 14th generation of PowerEdge servers.

Table 25. dellemc_configure_bios(continued)

Parameter/aliases	Required	Default	Choices	Comments
				NOTE: This option has been deprecated, and will be removed in the later version. Please use the l(attributes) for BIOS attributes configuration instead. **NOTE: l(nvme_mode) is mutually exclusive with l(boot_sources).**
secure_boot_mode	No	NA	• AuditMode, • DeployedMode • SetupMode • UserMode	(deprecated) Configures how the BIOS uses the Secure Boot Policy Objects in the 14th generation of PowerEdge servers. **NOTE: This option has been deprecated, and will be removed in the later version. Please use the l(attributes) for BIOS attributes configuration instead.** **NOTE: l(secure_boot_mode) is mutually exclusive with l(boot_sources).**
onetime_boot_mode	No	NA	• Disabled • OneTimeBootSeq • OneTimeCustomBootSeqStr • OneTimeCustomHddSeqStr • OneTimeCustomUefiBootSeqStr • OneTimeHddSeq • OneTimeUefiBootSeq	(deprecated) Configures the one time boot mode setting. **NOTE: This option has been deprecated, and will be removed in the later version. Please use the l(attributes) for BIOS attributes configuration instead.** **NOTE: l(onetime_boot_mode) is mutually exclusive with l(boot_sources).**
attributes	No	NA	NA	Dictionary of BIOS attributes and value pair. Attributes should be part of the Redfish Dell BIOS Attribute Registry. Redfish URI to view BIOS attributes: (https://l(idrac_ip)/redfish/v1/Systems/System.Embedded.1/Bios). If deprecated options are given and the same are repeated in l(attributes) then values in l(attributes) will take precedence. **NOTE: l(attributes) is mutually exclusive with l(boot_sources).**
boot_sources	No	NA	NA	List of boot devices to set the boot sources settings. Boot devices are dictionary. While applying boot sequence, **Index** of at least one boot device should be 0. **NOTE: l(boot_sources) is mutually exclusive with l(attributes), l(boot_sequence), l(onetime_boot_mode),**

Table 25. dellemc_configure_bios(continued)

Parameter/aliases	Required	Default	Choices	Comments
				l(secure_boot_mode), l(nvme_mode), and l(boot_mode). **NOTE:** When user does not provide `Index` or `Enabled` value in boot_sources option, dellemc_configure_bios module uses the current `Index` or `Enabled` value from the target server for the specified boot source while applying boot sources. **NOTE:** In case the selected `Index` or `Enabled` value from the target server conflicts with any of the boot_sources option values to be applied, dellemc_configure_bios module may fail to apply with appropriate error message.

Table 26. Return Values

Name	Description	Returned	Type	Sample
BIOS	Configures the BIOS configuration attributes	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_configure_bios.md

Examples

```
-name: Configure BIOS Generic attributes
   dellemc_configure_bios:
     idrac_ip:           "xx.xx.xx.xx"
     idrac_user:         "xxxx"
     idrac_password:     "xxxxxxxx"
     attributes:
       BootMode :        "Bios"
       OneTimeBootMode: "Enabled"
       BootSeqRetry:     "Enabled"
```

```
- name: Configure PXE Generic Attributes
    dellemc_configure_bios:
      idrac_ip:              "xx.xx.xx.xx"
      idrac_user:            "xxxx"
      idrac_password:        "xxxxxxxx"
      attributes:
        PxeDev1EnDis:        "Enabled"
        PxeDev1Protocol:     "IPV4"
        PxeDev1VlanEnDis:    "Enabled"
        PxeDev1VlanId:        x
        PxeDev1Interface:    "NIC.Embedded.x-x-x"
        PxeDev1VlanPriority: x
```

```
- name: Configure Boot Sources
    dellemc_configure_bios:
      idrac_ip:                "xx.xx.xx.xx"
      idrac_user:              "xxxx"
      idrac_password: "xxxxxxxx"
      boot_sources:
        - Name :    "NIC.Integrated.x-x-x"
```

```
          Enabled : True
          Index :   0
```

```
- name: Configure Boot Sources
   dellemc_configure_bios:
     idrac_ip:        "xx.xx.xx.xx"
     idrac_user:      "xxxx"
     idrac_password:  "xxxxxxxx"
     boot_sources:
       - Name : "NIC.Integrated.x-x-x"
         Enabled : True
         Index : 0
       - Name : "NIC.Integrated.x-x-x"
         Enabled : true
         Index : 1
       - Name : "NIC.Integrated.x-x-x"
         Enabled : true
         Index : 2
```

```
- name: Configure Boot Sources - Enabled
    dellemc_configure_bios:
      idrac_ip:                        "xx.xx.xx.xx"
      idrac_user:                      "xxxx"
      idrac_password:   "xxxxxxxx"
      boot_sources:
        - Name :   "NIC.Integrated.x-x-x"
          Enabled : True
```

```
- name: Configure Boot Sources - Index
    dellemc_configure_bios:
      idrac_ip:                       "xx.xx.xx.xx"
      idrac_user:                     "xxxx"
      idrac_password:  "xxxxxxxx"
      boot_sources:
        - Name : "NIC.Integrated.x-x-x"
          Index : 0
```

Configure RAID

Module: dellemc_configure_raid

Synopsis

This module hosts the RAID configuration related attributes.

NOTE: This module is deprecated and replaced with dellemc_idrac_storage_volume.

Options

Table 27. dellemc_configure_raid

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS Network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'.

Table 27. dellemc_configure_raid(continued)

Parameter/aliases	Required	Default	Choices	Comments
				This option is mandatory for CIFS Network share.
share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This option is mandatory for Network share.
vd_name	No	NA	NA	Virtual disk name • Optional, if we perform create operations • Mandatory, if we perform remove operations
span_depth	No	1	NA	Span Depth
span_length	No	2	NA	Span Length
number_dedicated_hot_spare	No	0	NA	Number of Dedicated Hot Spare
number_global_hot_spare	No	0	NA	Number of Global Hot Spare
raid_level	No	RAID 0	• RAID 0 • RAID 1 • RAID 5 • RAID 6 • RAID 10 • RAID 50 • RAID 60	Provide the required RAID level
disk_cache_policy	No	Default	• Default • Enabled • Disabled	Disk Cache Policy
write_cache_policy	No	WriteThrough	• WriteThrough • WriteBack • WriteBackForce	Write cache policy
read_cache_policy	No	NoReadAhead	• NoReadAhead • ReadAhead • Adaptive	Read cache policy
stripe_size	No	65536	NA	Provide stripe size value in multiples of 64 * 1024
controller_fqdd	Yes	NA	NA	Fully Qualified Device Descriptor (FQDD) of the storage controller, for e.g. `RAID.Integrated.1-1`
media_type	No	HDD	• HDD • SSD	Media type
bus_protocol	No	SATA	• SAS • SATA	Bus protocol

Table 27. dellemc_configure_raid(continued)

Parameter/aliases	Required	Default	Choices	Comments
state	Yes	NA	• present • absent	• If the value is 'present', the module will perform 'create' operations • If the value is 'absent', the module will perform 'remove' operations

Table 28. Return Values

Name	Description	Returned	Type	Sample
RAID configuration	Configures the RAID configuration attributes	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_configure_raid.md

Example

```
-name: Configure the RAID attributes
   dellemc_configure_raid:
     idrac_ip:         "xx.xx.xx.xx"
     idrac_user:       "xxxx"
     idrac_pwd:        "xxxxxxxx"
     share_name:       "xx.xx.xx.xx:/share"
     share_pwd:        "xxxxxxxx"
     share_user:       "xxxx"
     share_mnt:        "xxxxxx"
     controller_fqdd: "xxxxxxxx"
     vd_name:          "xxxxxx"
```

Configure storage volume

Module: dellemc_idrac_storage_volume

Synopsis

This module hosts the RAID configuration related attributes.

Check_mode support: Yes

Options

Table 29. dellemc_idrac_storage_volume

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
span_depth	No	1	NA	Span Depth
span_length	No	1	NA	Span Length
number_dedicated_hot_spare	No	0	NA	Number of Dedicated Hot Spare
volume_type	No	RAID 0	• RAID 0 • RAID 1 • RAID 5 • RAID 6	Provide the required RAID level

Table 29. dellemc_idrac_storage_volume(continued)

Parameter/aliases	Required	Default	Choices	Comments
			• RAID 10 • RAID 50 • RAID 60	
disk_cache_policy	No	Default	• Default • Enabled • Disabled	Disk Cache Policy
write_cache_policy	No	WriteThrough	• WriteThrough • WriteBack • WriteBackForce	Write Cache Policy
read_cache_policy	No	NoReadAhead	• NoReadAhead • ReadAhead • AdaptiveReadAhead	Read Cache Policy
stripe_size	No	65536	NA	Provide stripe size value in multiples of 64 * 1024
controller_id	No	NA	NA	Fully Qualified Device Descriptor (FQDD) of the storage controller, for example: `RAID.Integrated.1-1` **NOTE: Controller FQDD is required for C(create) RAID configuration.**
volume_id	No	NA	NA	Fully Qualified Device Descriptor (FQDD) of the virtual disk, for example: `Disk.virtual.0:RAID.Slot.1-1` **NOTE: This option is used to get the virtual disk information.**
media_type	No	None	• HDD • SDD	Media type
protocol	No	None	• SAS • SATA	Bus protocol
state	Yes	view	• create • delete • view	• C(create) performs **create** volume operations. • C(delete) performs **remove** volume operations. • C(view) returns the storage view.
volumes	No	NA	NA	A list of virtual disk-specific iDRAC attributes. This is applicable for C(create) and C(delete) operations. • For C(create) operation, name and drives are applicable options, other volume options can also be specified. **NOTE: The drives is a required option for C(create) operation and accepts either location (list of drive slot) or id (list of drive fqdd).** • For C(delete) operation, only name option is applicable.

Table 29. dellemc_idrac_storage_volume(continued)

Parameter/aliases	Required	Default	Choices	Comments
capacity	No	NA	NA	Virtual disk size in GB
raid_reset_config	No	NA	NA	This option represents whether a Reset Config operation needs to be performed on the RAID controller. Reset Config operation deletes all the virtual disks present on the RAID controller.
raid_init_operation	No	None	• None • Fast	This option represents Initialization Configuration operation to be performed on the virtual disk.

Return Values

```
msg:
  type: str
  description: Overall status of the storage configuration operation.
  returned: always
  sample: "Successfully completed the view storage volume operation"
storage_status:
  type: dict
  description: Storage configuration job and progress details from the iDRAC.
  returned: success
  sample:
    {
      "Id": "JID_XXXXXXXXX",
      "JobState": "Completed",
      "JobType": "ImportConfiguration",
      "Message": "Successfully imported and applied Server Configuration Profile.",
      "MessageId": "XXX123",
      "Name": "Import Configuration",
      "PercentComplete": 100,
      "StartTime": "TIME_NOW",
      "Status": "Success",
      "TargetSettingsURI": null,
      "retval": true
    }
```

Examples

```
-name: Create single volume
   dellemc_idrac_storage_volume:
     idrac_ip:        "192.168.0.1"
     idrac_user:      "username"
     idrac_password:  "password"
     controller_id:   "RAID.Slot.1-1"
     state:           "create"
     volumes:
        - drives:
          location:    [5]
```

```
-name: Create multiple volume
   dellemc_idrac_storage_volume:
     idrac_ip:                   "192.168.0.1"
     idrac_user:                 "username"
     idrac_password:             "password"
     raid_reset_config:          "True"
     state:                      "create"
     controller_id:              "RAID.Slot.1-1"
     volume_type:                 "RAID 1"
     span_depth:                   1
     span_length:                  2
     number_dedicated_hot_spare:  1
     disk_cache_policy:          "Enabled"
     write_cache_policy:         "WriteBackForce"
     read_cache_policy:          "ReadAhead"
```

```
    stripe_size:                 65536
    capacity:                    100
    raid_init_operation:        "Fast"
    volumes:
      - name:                   "volume_1"
        drives:
            id:                 ["Disk.Bay.1:Enclosure.Internal.0-1:RAID.Slot.1-1",
                                 "Disk.Bay.2:Enclosure.Internal.0-1:RAID.Slot.1-1"]
      - name:                   "volume_2"
        volume_type:            "RAID 5"
        span_length:             3
        span_depth:              1
        drives:
            location:           [7,3,5]
        disk_cache_policy:      "Disabled"
        write_cache_policy:     "WriteBack"
        read_cache_policy:      "NoReadAhead"
        stripe_size:             131072
        capacity:                200
        raid_init_operation:    "None"
```

```
-name: View all volume details
   dellemc_idrac_storage_volume:
     idrac_ip:                       "192.168.0.1"
     idrac_user:                     "username"
     idrac_password:  "password"
     state:                          "view"
```

```
-name: View specific volume details
   dellemc_idrac_storage_volume:
     idrac_ip:         "192.168.0.1"
     idrac_user:       "username"
     idrac_password:   "password"
     state:            "view"
     controller_id:    "RAID.Slot.1-1"
     volume_id:        "Disk.Virtual.0:RAID.Slot.1-1"
```

```
-name: Delete single volume
   dellemc_idrac_storage_volume:
     idrac_ip:                "192.168.0.1"
     idrac_user:              "username"
     idrac_password: "password"
     state:                   "delete"
     volumes:
       - name:    "volume_1"
```

```
-name: Delete multiple volume
   dellemc_idrac_storage_volume:
     idrac_ip:                "192.168.0.1"
     idrac_user:              "username"
     idrac_password: "password"
     state:                   "delete"
     volumes:
       - name: "volume_1"
       - name: "volume_2"
```

Configure Collect System Inventory on Restart

Module: dellemc_idrac_lc_attributes

Synopsis

This module is responsible for enabling or disabling of **Collect System Inventory on Restart (CSIOR)** property for all iDRAC or LC jobs. When you enable the **CSIOR** property, hardware inventory and part configuration information are discovered and compared with previous system inventory information on every system restart.

Check_mode support: Yes

Options

Table 30. dellemc_idrac_lc_attributes

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This option is mandatory for CIFS or NFS Network share.
csior	Yes	NA	· Enabled · Disabled	Whether to Enable or Disable Collect System Inventory on Restart (CSIOR) property for all iDRAC or LC jobs

Table 31. Return Values

Name	Description	Returned	Type	Sample
iDRAC CSIOR	Configures CSIOR property for all iDRAC or LC jobs	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_idrac_lc_attributes.md

Example

```
-name: Set up iDRAC LC Attributes
   dellemc_idrac_lc_attributes:
     idrac_ip:   "xx.xx.xx.xx"
     idrac_user: "xxxx"
     idrac_password:  "xxxxxxxx"
     share_name: "xx.xx.xx.xx:/share"
     share_user: "xxxxx"
     share_password:  "xxxxxxxx"
     share_mnt:  "/mnt/share"
     csior:      "xxxxxxx"
```

Configure syslog

Module: dellemc_setup_idrac_syslog

Synopsis

This module enables or disables syslog parameters for iDRAC.

Check_mode support: Yes

Options

Table 32. dellemc_setup_idrac_syslog

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS Network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This option is mandatory for CIFS or NFS Network share.
syslog	Yes	NA	• Enabled • Disabled	Whether to Enable or Disable iDRAC syslog

Table 33. Return Values

Nam	Description	Returned	Type	Sample
iDRAC Syslog	Configures iDRAC Syslog parameters	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_setup_idrac_syslog.md

Example

```
-name: Configure iDRAC Syslog Parameters
   dellemc_setup_idrac_syslog:
     idrac_ip:      "xx.xx.xx.xx"
     idrac_user:    "xxxx"
     idrac_password:"xxxxxxxx"
     share_name:    "xx.xx.xx.xx:/share"
     share_user:    "xxxx"
     share_password:"xxxxxxxx"
     share_mnt:     "/mnt/share"
     syslog:        "xxxxxxx"
```

Deploying operating system

To provision a bare metal server, it is essential to deploy the required operating system in the device before you start using it. This section describes the process of deploying the operating system on the PowerEdge servers using Ansible.

To automate the process of operating system deployment in an unattended manner using Ansible, the iDRAC's capability is utilized to transfer the customized ISO to iDRAC for boot.

To perform OS deployment, ensure:

- Operating system image is injected with required Dell drivers, and unattended response file.

- iDRAC is enabled, configured, and reachable.
- RAID is configured.

Boot to a network ISO image

Module: idrac_os_deployment

Synopsis

This module facilitates the operating system deployment. You can run this module to boot the target system to a bootable ISO image on a CIFS or NFS share. This module looks for the customized ISO in the configured share location and transfers the image to iDRAC to load it. On the system reboot, the operating system deployment begins.

Check_mode support: No

Options

Table 34. idrac_os_deployment

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC password
idrac_port	No	443	NA	iDRAC port
iso_image	Yes	NA	NA	Network ISO name
share_name	Yes	NA	NA	CIFS or NFS Network share
share_user	No	NA	NA	User name required to access the network share must be provided as either 'user@domain' or 'domain\user'. This option is mandatory for CIFS network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network shares.
expose_duration	No	1080	NA	It is the time taken in minutes for the ISO image file to be exposed as a local CD-ROM device to the host server. When the specified time expires, the ISO image gets automatically detached.

Table 35. Return Values

Name	Description	Returned	Type	Sample
Boot to Network ISO	Boots to a network ISO Image	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_boot_to_network_iso.md

Example

```
-name: Boot to Network ISO
           idrac_os_deployment:
     idrac_ip:       "192.168.0.1"
     idrac_user:     "user_name"
     idrac_password: "user_password"
     share_name:     "192.168.0.0:/nfsfileshare"
     share_user:     "share_user_name"
     share_password: "share_user_pwd"
     iso_image:      "unattended_os_image.iso"
     expose_duration: 180
```

Return

```
msg:
                description: Details of the boot to network ISO image operation.
                returned: always
                type: dict
                sample: {
                    "DeleteOnCompletion": "false",
                    "InstanceID": "DCIM_OSDConcreteJob:1",
                    "JobName": "BootToNetworkISO",
                    "JobStatus": "Success",
                    "Message": "The command was successful.",
                    "MessageID": "OSD1",
                    "Name": "BootToNetworkISO",
                    "Status": "Success",
                    "file": "192.168.0.0:/nfsfileshare/unattended_os_image.iso",
                    "retval": true
                }
            '''
```

Module: dellemc_boot_to_network_iso

Synopsis

This module facilitates the operating system deployment. You can run this module to boot the target system to a bootable ISO image on a CIFS or NFS share. This module looks for the customized ISO in the configured share location and transfers the image to iDRAC to load it. On the system reboot, the OS deployment begins.

Check_mode support: No

NOTE: This module is deprecated and replaced with idrac_os_deployment.

Options

Table 36. dellemc_boot_to_network_iso

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC password
idrac_port	No	443	NA	iDRAC port
iso_image	Yes	NA	NA	Network ISO name
share_name	Yes	NA	NA	CIFS or NFS Network share
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.

Table 37. Return Values

Name	Description	Returned	Type	Sample
Boot to Network ISO	Boots to a network ISO Image	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_boot_to_network_iso.md

Example

```
-name: Boot to Network ISO
   dellemc_boot_to_network_iso:
```

```
    idrac_ip:        "xx.xx.xx.xx"
    idrac_user:      "xxxx"
    idrac_password:  "xxxxxxxx"
    share_name:      "xx.xx.xx.xx:/share"
    share_user:      "xxxx"
    share_password:  "xxxxxxxx"
    iso_image:       "uninterrupted_os_installation_image.iso"
```

Server Inventory

This section describes the process of retrieving the server inventory of the PowerEdge Servers using Ansible Modules.

View the system inventory

Module: dellemc_get_system_inventory

Synopsis

System inventory provides basic and component level detailed inventory information. You can run this module when you want to verify the asset, configured state, inventory, and health-related information for the system and its component.

Check_mode support: No

Options

Table 38. dellemc_get_system_inventory

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port

Table 39. Return Values

Name	Description	Returned	Type	Sample
System Inventory	Displays the PowerEdge Server System Inventory	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_get_system_inventory.md

Example

```
-name: Get System Inventory
   dellemc_get_system_inventory:
     idrac_ip:       "xx.xx.xx.xx"
     idrac_user:     "xxxx"
     idrac_password:"xxxxxxxx"
```

Server administration tasks

This section describes the tasks that you can run using OpenManage Ansible Modules. Following are the tasks:

- Configure the power state on the PowerEdge servers
- Reset iDRAC
- View LC job status
- Export LC logs
- Delete LC job
- Delete LC job queue
- Configure System Lockdown Mode

Configure the power state on the PowerEdge servers

Module: dellemc_change_power_state

Synopsis

This module configures the power control options on a PowerEdge server. You can run this module:

- To turn on the server.
- To turn off the server.
- To reboot the server.
- For hard reset of the server.

Check_mode support: Yes

Options

Table 40. dellemc_change_power_state

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
change_power	Yes	NA	• On • ForceOff • GracefulRestart • GracefulShutdown • PushPowerButton • Nmi	Desired power state

Table 41. Return Values

Name	Description	Returned	Type	Sample
Power state of a server	Configures the power control options on a PowerEdge server	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_change_power_state.md

Example

```
-name: Change Power State
   dellemc_change_power_state:
     idrac_ip:       "xx.xx.xx.xx"
     idrac_user:     "xxxx"
     idrac_password:"xxxxxxxx"
     change_power:   "xxxxxxx"
```

Reset iDRAC

Module: dellemc_idrac_reset

Synopsis

You can reset the iDRAC using this module.

Check_mode support: Yes

Options

Table 42. dellemc_idrac_reset

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port

Table 43. Return Values

Name	Description	Returned	Type	Sample
Reset iDRAC	Resets the iDRAC	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_idrac_reset.md

Example

```
-name: Reset iDRAC
   dellemc_idrac_reset:
     idrac_ip:       "xx.xx.xx.xx"
     idrac_user:     "xxxx"
     idrac_password:"xxxxxxxx"
     idrac_port:     "xxx"
```

View LC job status

Module: dellemc_get_lc_job_status

Synopsis

You can view the iDRAC or LC job status using this module. To view information about a job status, a job id is required. After a job is initiated, the system stages the job request information and sends a job id back to the system. You can query the progress and status of the job by using the job id.

Check_mode support: No

Options

Table 44. dellemc_get_lc_job_status

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
job_id	Yes	NA	NA	JOB ID in the format "JID_123456789012"

Table 45. Return Values

Name	Description	Returned	Type	Sample
LC Job Status	Displays the status of an LC job	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_get_lc_job_status.md

Example

```
-name: Get LC Job Status
   dellemc_get_lc_job_status
     idrac_ip:       "xx.xx.xx.xx"
     idrac_user:     "xxxx"
     idrac_password: "xxxxxxxx"
     job_id:         "JID_1234567890"
```

Export LC logs

Module: dellemc_export_lc_logs

Synopsis

LC logs provide records of past activities on a managed system. These log files are useful for the server administrators since they provide detailed information about recommended actions and some other technical information that is useful for troubleshooting purposes.

The various types of information available in LC logs are alerts-related, configuration changes on the system hardware components, firmware changes due to an upgrade or downgrade, replaced parts, temperature warnings, detailed timestamps of when the activity has started, severity of the activity, and so on.

Check_mode support: No

Options

Table 46. dellemc_export_lc_logs

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS Network share
share_user	No	NA	NA	Network share user in the format 'user@domain' or 'domain\user' if user is part of a domain else 'user'. This option is mandatory for CIFS Network share.
share_password/ share_pwd	No	NA	NA	Network share user password. This option is mandatory for CIFS Network share.
job_wait	Yes	NA	• True • False	• If the value is True, it waits for the job to complete and returns the job completion status • If the value is False, it returns immediately with a JOB ID after queuing the job in LC job queue

Table 47. Return Values

Name	Description	Returned	Type	Sample
LC logs	Exports the LC logs to the given network share	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_export_lc_logs.md

Example

```
-name: Export Lifecycle Controller Logs
   dellemc_export_lc_logs:
     idrac_ip:       "xx.xx.xx.xx"
     idrac_user:     "xxxx"
     idrac_password: "xxxxxxxx"
     idrac_port:     "xxx"
     share_name:     "xx.xx.xx.xx:/share"
```

```
    share_user:     "xxxx"
    share_password: "xxxxxxxx"
    job_wait:       "True"
```

Delete LC job

Module: dellemc_delete_lc_job

Synopsis

This module deletes an LC job for a given valid JOB ID from the job queue.

You can delete an LC job:

- after the job is completed.
- if you do not want to perform the job or if it is taking long to execute.

Check_mode support: Yes

Options

Table 48. dellemc_delete_lc_job

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
job_id	Yes	NA	NA	JOB ID in the format "JID_XXXXXXXXX"

Table 49. Return Values

Name	Description	Returned	Type	Sample
Delete LC job	Deletes an LC job for a given a JOB ID	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_delete_lc_job.md

Examples

```
-name: Delete LC Job
   dellemc_delete_lc_job:
     idrac_ip:        "xx.xx.xx.xx"
     idrac_user:      "xxxx"
     idrac_password:  "xxxxx"
     idrac_port:      "xxx"
     job_id:          "JID_XXXXXXXX"
```

Delete LC job queue

Module: dellemc_delete_lc_job_queue

Synopsis

You can delete all the jobs in the LC job queue using this module. All the jobs in the job queue are terminated when you delete a job queue.

Check_mode support: No

Options

Table 50. dellemc_delete_lc_job_queue

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port

Table 51. Return Values

Name	Description	Returned	Type	Sample
LC Job Queue	Deletes the LC job queue	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_delete_lc_job_queue.md

Example

```
-name: Delete LC Job Queue
   dellemc_delete_lc_job_queue:
     idrac_ip:       "xx.xx.xx.xx"
     idrac_user:     "xxxx"
     idrac_password:"xxxxx"
     idrac_port:     "xxx"
```

Configure System Lockdown Mode

Module: dellemc_system_lockdown_mode

Synopsis

System Lockdown Mode provides a mechanism to protect configuration from any unintentional or accidental changes after the system is provisioned to a certain level.

This module is responsible for enabling or disabling the lockdown mode of a system. When System Lockdown Mode is enabled, the system's configuration is locked and system cannot be configured or updated until the lockdown mode is disabled.

Check_mode support: No

Options

Table 52. dellemc_system_lockdown_mode

Parameter/aliases	Required	Default	Choices	Comments
idrac_ip	Yes	NA	NA	iDRAC IP Address
idrac_user	Yes	NA	NA	iDRAC username
idrac_password/ idrac_pwd	Yes	NA	NA	iDRAC user password
idrac_port	No	443	NA	iDRAC port
share_name	Yes	NA	NA	CIFS or NFS network share or a local path
share_user	No	NA	NA	Network share user in the format 'user@domain' or user\domain if user is part of a domain else 'user'. This field is mandatory for CIFS Network Share.
share_password/ share_pwd	No	NA	NA	Network share user password. This field is mandatory for CIFS Network Share.
share_mnt	No	NA	NA	Local mount path of the network share with read-write permission for Ansible user. This option is mandatory for CIFS or NFS Network share.

Table 52. dellemc_system_lockdown_mode(continued)

Parameter/aliases	Required	Default	Choices	Comments
lockdown_mode	Yes	NA	• Enabled • Disabled	Whether to Enable or Disable system lockdown mode

Table 53. Return Values

Name	Description	Returned	Type	Sample
System Lockdown Mode	Configures lockdown mode of the system	Success	String	https://github.com/dell/Dell-EMC-Ansible-Modules-for-iDRAC/blob/master/samples/dellemc_system_lockdown_mode.md

Example

```
-name: Configure System Lockdown Mode
   dellemc_system_lockdown_mode:
     idrac_ip:      "xx.xx.xx.xx"
     idrac_user:    "xxxx"
     idrac_password:"xxxxxxxx"
     share_name:    "xx.xx.xx.xx:/share"
     share_user:    "xxxx"
     share_password:"xxxxxxxx"
     share_mnt:     "/mnt/share"
     lockdown_mode: "xxxxxxx"
```

Storage controller

This section describes the process of configuring the storage controller settings of the PowerEdge servers using Ansible modules.

Configure storage controller settings

Module: idrac_redfish_storage_controller

Synopsis

This module configures the storage controller settings using Redfish APIs.

Options

Table 54. idrac_redfish-storage-controller

Parameter	Required	Default	Choices	Comments
baseuri	True	NA	NA	IP address of the target iDRAC. For example- <ipaddress>:<port>
username	True	NA	NA	Username of the target iDRAC.
password	True	NA	NA	Password of the target iDRAC.
command	False	AssignSpare	ResetConfig, AssignSpare, SetControllerKey, RemoveControllerKey,or ReKey.	Set of actions to configure the storage controller settings. • C(ResetConfig) - Deletes all the virtual disks and unassigns all hot spares on physical disks. • C(AssignSpare) - Assigns a physical disk as a dedicated or global hot spare for a virtual disk. • C(SetControllerKey) - Sets the key on controllers, which is

Table 54. idrac_redfish-storage-controller(continued)

Parameter	Required	Default	Choices	Comments
				used to encrypt the drives in Local key Management(LKM). • C(RemoveControllerKey) - Erases the encryption key on the controller. • C(ReKey) - Resets the key on the controller.
target	False	NA	NA	• Fully Qualified Device Descriptor (FQDD) of the target physical drive that is assigned as a spare. • This [option] is mandatory when I(command) is C(AssignSpare) • If I(volume_id) is not specified or empty, this physical drive will be assigned as a global hot spare.
volume_id	False	NA	NA	• FQDD of the volumes to which a hot spare is assigned. • [This option is] Applicable if I(command) is C(AssignSpare). • To know the number of volumes to which a hot spare can be assigned, refer iDRAC Redfish API guide.
controller_id	False	NA	NA	• FQDD of the storage controller. For example- 'RAID.Slot.1-1'. • This option is mandatory when I(command) is C(ResetConfig), C(SetControllerKey), C(RemoveControllerKey) and C(ReKey).
key	False	NA	NA	• A new security key passphrase that the encryption-capable controller uses to create the encryption key. The controller uses the encryption key to lock or unlock access to the Self Encryption Disk(SED). • Only one encryption key can be created for each controller. • This option is mandatory when I(command) is C(SetControllerKey) or C(ReKey), and when I(mode) is C(LKM).
key_id	False	NA	NA	• This is a user supplied text label associated with the passphrase. • This option is mandatory when I(command) is C(SetControllerKey) or

Table 54. idrac_redfish-storage-controller(continued)

Parameter	Required	Default	Choices	Comments
				C(ReKey), and when I(mode) is C(LKM).
old_key	False	NA	NA	• Security key passphrase used by the encryption-capable controller. • This option is mandatory when I(command) is C(ReKey) and I(mode) is C(LKM). .
mode	False	NA	LKM or SEKM	• Encryption mode of the encryption-capable controller: 1 - Local Key Management (LKM), 2 - Security Enterprise Key Manager(SEKM). • This option is applicable only when I(command) is C(ReKey). • C(SEKM) requires secure enterprise key manager license on the iDRAC.

Return values

```
msg:
  type: str
  description: Overall status of the storage controller configuration operation.
  returned: always
  sample: "Successfully submitted the job that performs AssignSpare operation"
task:
  type: dict
  description: ID and URI resource of the created job.
  returned: success
  sample: {
    "id": "JID_XXXXXXXXXXXX",
    "uri": "/redfish/v1/Managers/iDRAC.Embedded.1/Jobs/JID_XXXXXXXXXXXX"
  }
error_info:
  type: dict
  description: Details of a http error.
  returned: on http error
  sample:  {
    "error": {
      "@Message.ExtendedInfo": [
        {
          "Message": "Cannot run the method because the requested HTTP method is not
allowed.",
          "MessageArgs": [],
          "MessageArgs@odata.count": 0,
          "MessageId": "iDRAC.1.6.SYS402",
          "RelatedProperties": [],
          "RelatedProperties@odata.count": 0,
          "Resolution": "Enter a valid HTTP method and retry the operation. For information
about
          valid methods, see the Redfish Users Guide available on the support site.",
          "Severity": "Informational"
        }
      ],
      "code": "Base.1.0.GeneralError",
      "message": "A general error has occurred. See ExtendedInfo for more information"
    }
  }
```

Examples

```
- name: Assign [a] dedicated hot spare.
  idrac_redfish_storage_controller:
    baseuri: "192.168.0.1:443"
    username: "user_name"
    password: "user_password"
    volume_id: "Disk.Virtual.0:RAID.Slot.1-1"
    target: "Disk.Bay.0:Enclosure.Internal.0-1:RAID.Slot.1-1"
  tags:
    - assign_dedicated_hot_spare

- name: Assign [a] global hot spare.
  idrac_redfish_storage_controller:
    baseuri: "192.168.0.1:443"
    username: "user_name"
    password: "user_password"
    target: "Disk.Bay.0:Enclosure.Internal.0-1:RAID.Slot.1-1"
  tags:
    - assign_global_hot_spare

- name: Set [a] controller encryption key.
  idrac_redfish_storage_controller:
    baseuri: "192.168.0.1:443"
    username: "user_name"
    password: "user_password"
    command: "SetControllerKey"
    controller_id: "RAID.Slot.1-1"
    key: "PassPhrase@123"
    key_id: "mykeyid123"
  tags:
    - set_controller_key

- name: Rekey in LKM mode.
  idrac_redfish_storage_controller:
    baseuri: "192.168.0.1:443"
    username: "user_name"
    password: "user_password"
    command: "ReKey"
    controller_id: "RAID.Slot.1-1"
    key: "PassPhrase@123"
    key_id: "mykeyid123"
    old_key: "OldPassPhrase@123"
  tags:
    - rekey_lkm

- name: Rekey in SEKM mode.
  idrac_redfish_storage_controller:
    baseuri: "192.168.0.1:443"
    username: "user_name"
    password: "user_password"
    command: "ReKey"
    controller_id: "RAID.Slot.1-1"
    mode: "SEKM"
  tags:
    - rekey_sekm

- name: Remove [the] controller key.
  idrac_redfish_storage_controller:
    baseuri: "192.168.0.1:443"
    username: "user_name"
    password: "user_password"
    command: "RemoveControllerKey"
    controller_id: "RAID.Slot.1-1"
  tags:
    - remove_controller_key

- name: Reset configuration.
  idrac_redfish_storage_controller:
    baseuri: "192.168.0.1:443"
    username: "user_name"
    password: "user_password"
    command: "ResetConfig"
```

```
    controller_id: "RAID.Slot.1-1"
  tags:
    - reset_config
```

4

Modules for OpenManage Enterprise (OME)

How OpenManage Ansible Modules work with OME

OpenManage Enterprise (OME) is a system management and monitoring application that provides rich sets of features to manage the Dell EMC servers, chassis, storage, and network switches in an enterprise data center or IT environment. Using the comprehensive set of REST APIs provided by OME, system administrators and software developers can discover, configure, provision, update, and manage their entire Dell EMC infrastructure.

OpenManage Ansible modules for OME simplifies and automates the PowerEdge server and modular infrastructure provisioning, deployment, and updates supported by OME. Leveraging the repeatable template configuration and deployment feature provided by OME, administrators can automatically deploy the changes, ensure consistency and thereby significantly improve productivity by reducing manual interactions and errors.

For information on which user roles in OME have the required privileges to run modules, refer roles and associated privileges for OME.

Running your first OME Playbook

Before you run a playbook to manage your iDRACs using OME, you need to have an inventory file that contains the target OME server details. For more information on inventory, see Ansible documentation

1. Install OpenManage Ansible Modules either from the dell.com/support or the https://github.com/dell/dellemc-openmanage-ansible-modules.git repository. For more details, see *Dell EM C OpenManage Ansible Modules Installation Guide*.
2. Create an inventory file containing a list of the OMEs. In the following inventory example, we are using the inventory variables to store the OME IP addresses and the user credentials. For more information on variables, see Ansible documentation.

```
inventory:

[PowerEdge]
ome.example.com
ome_ipaddress= '192.168.1.1'
ome_username='ome_user'
ome_password='ome_password'
```

3. Define a playbook to fetch the server inventory managed by the OME. Create the playbook in the same directory where you created the inventory. Following is a playbook example:

```
playbook.yml

---
- hosts: PowerEdge
  connection: local
  gather_facts: False

  tasks:
  - name: Get server inventory
    dellemc_ome_device_facts:
      hostname:  "{{ ome_ipaddress }}"
      username:  "{{ ome_username }}"
      password:  "{{ ome_password }}"
    system_query_options:
      filter:  "Type eq 1000"
```

4. Now run the playbook. Run the following command from the directory where you created the inventory and the playbook:

```
ansible-playbook playbook.yml -i inventory
```

5. Press **Enter**.

With OpenManage Ansible Modules, you can construct a playbook with a set of modules resulting in an automation workflow for configuration, deployments, and updates of PowerEdge and modular servers.

To view the list of all available OME modules:

1. Run the following command on the Ansible control machine:

```
ansible-doc -l | grep "ome"
```

2. Press **Enter**.

List of the available OME modules is displayed.

To view the documentation of a module:

1. Run the following command on the Ansible control machine:

```
ansible-doc <module name>
```

2. Press **Enter**.

View device information

Module: ome_device_info

Synopsis

This module retrieves the list of devices in the inventory of OpenManage Enterprise along with the details of each device.

Options

Table 55. ome_device_info

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
fact_subset	No	basic_inventory	· basic_inventory · detailed_inventory · subsystem_health	· C(basic_inventory) returns the list of the devices. · C(detailed_inventory) returns the inventory details of specified devices. · C(subsystem_health) returns the health status of specified devices.
system_query_options	No	NA	· device_id: A list of unique identifier is applicable for C(detailed_inventory) and C(subsystem_health). · device_service_tag: A list of service tags is applicable for C(detailed_inventory) and C(subsystem_health). · inventory_type: For C(detailed_inventory), it returns details of the specified inventory type. · filter: For C(basic_inventory), it filters the collection of devices. I(filter) query format should be aligned with OData standards.	· I(system_query_options) is applicable for the choices of the fact_subset. · Either I(device_id) or I(device_service_tag) is mandatory for C(detailed_inventory) and C(subsystem_health). · I(device_id) or I(device_service_tag)

Table 55. ome_device_info(continued)

Parameter	Required	Default	Choices	Comments
				can be used individually or together.

Return Values

```
msg:
  type: str
  description: Overall device information status.
  returned: on error
  sample: "Failed to fetch the device information"
device_info:
  type: dict
  description: Returns the information collected from the device.
  returned: success
  sample: {
        "value": [
            {
                "Actions": null,
                "AssetTag": null,
                "ChassisServiceTag": null,
                "ConnectionState": true,
                "DeviceManagement": [
                    {
                        "DnsName": "dnsname.host.com",
                        "InstrumentationName": "MX-12345",
                        "MacAddress": "11:10:11:10:11:10"
                        "ManagementId": 12345,
                        "ManagementProfile": [
                            {
                                "HasCreds": 0,
                                "ManagementId": 12345,
                                "ManagementProfileId": 12345,
                                "ManagementURL": "https://192.168.0.1:443",
                                "Status": 1000,
                                "StatusDateTime": "2019-01-21 06:30:08.501"
                            }
                        ],
                       "ManagementType": 2,
                       "NetworkAddress": "192.168.0.1"
                    }
                ],
                "DeviceName": "MX-0003I",
                "DeviceServiceTag": "MXL1234",
                "DeviceSubscription": null,
                "LastInventoryTime": "2019-01-21 06:30:08.501",
                "LastStatusTime": "2019-01-21 06:30:02.492",
                "ManagedState": 3000,
                "Model": "PowerEdge MX7000",
                "PowerState": 17,
                "SlotConfiguration": {},
                "Status": 4000,
                "SystemId": 2031,
                "Type": 2000
            }
        ]
    }
```

Examples

```
- name: Retrieve basic inventory of all devices.
  ome_device_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"


- name: Retrieve basic inventory for devices identified by IDs 33333 or 11111 using filtering.
```

```
  ome_device_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    fact_subset: "basic_inventory"
    system_query_options:
      filter: "Id eq 33333 or Id eq 11111"


- name: Retrieve inventory details of specified devices identified by IDs 11111 and 22222.
  ome_device_info:
     hostname:  "192.168.0.1"
     username: "username"
     password: "password"
     fact_subset: "detailed_inventory"
     system_query_options:
       device_id:
        - 11111
        - 22222


- name: Retrieve inventory details of specified devices identified by service tags MXL1234
and MXL4567.
  ome_device_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    fact_subset: "detailed_inventory"
    system_query_options:
      device_service_tag:
        - MXL1234
        - MXL4567


- name: Retrieve details of specified inventory type of specified devices identified by ID
and service tags.
  ome_device_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    fact_subset: "detailed_inventory"
    system_query_options:
      device_id:
        - 11111
      device_service_tag:
        - MXL1234
        - MXL4567
      inventory_type: "serverDeviceCards"


- name: Retrieve subsystem health of specified devices identified by service tags.
  ome_device_info:
    hostname:  "192.168.0.1"
    username:  "username"
    password:  "password"
    fact_subset: "subsystem_health"
    system_query_options:
      device_service_tag:
        - MXL1234
        - MXL4567
```

View device inventory

Module: dellemc_ome_device_facts

Synopsis

This module retrieves the list of all devices with the exhaustive inventory of each device discovered using OpenManage Enterprise.

NOTE: This module is deprecated and replaced with ome_device_info

Options

Table 56. dellemc_ome_device_facts

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
fact_subset	No	basic_inventory	· basic_inventory · detailed_inventory · subsystem_health	· C(basic_inventory) returns the list of the devices. · C(detailed_inventory) returns the inventory details of specified devices. · C(subsystem_health) returns the health status of specified devices.
system_query_options	No	NA	· device_id: A list of unique identifier is applicable for C(detailed_inventory) and C(subsystem_health). · device_service_tag: A list of service tags is applicable for C(detailed_inventory) and C(subsystem_health). · inventory_type: For C(detailed_inventory), it returns details of the specified inventory type. · filter: For C(basic_inventory), it filters the collection of devices. I(filter) query format should be aligned with OData standards.	I(system_query_options) is applicable for the choices of the fact_subset. Either I(device_id) or I(device_service_tag) is mandatory for C(detailed_inventory) and C(subsystem_health) or both can be applicable.

Return Values

```
msg:
  type: str
  description: Over all device_facts status.
  returned: on error
  sample: "Failed to fetch the device facts"
ansible_facts:
  type: dict
  description: Device inventory details.
  returned: success
  sample: {
        "value": [
            {
                "Actions": null,
                "AssetTag": null,
                "ChassisServiceTag": null,
                "ConnectionState": true,
                "DeviceManagement": [
                    {
                        "DnsName": "dnsname.host.com",
                        "InstrumentationName": "MX-12345",
                        "MacAddress": "11:10:11:10:11:10"
                        "ManagementId": 12345,
                        "ManagementProfile": [
                            {
                                "HasCreds": 0,
                                "ManagementId": 12345,
                                "ManagementProfileId": 12345,
                                "ManagementURL": "https://192.168.0.1:443",
```

```
                                "Status": 1000,
                                "StatusDateTime": "2019-01-21 06:30:08.501"
                            }
                        ],
                        "ManagementType": 2,
                        "NetworkAddress": "192.168.0.1"
                    }
                ],
                "DeviceName": "MX-0003I",
                "DeviceServiceTag": "MXL1234",
                "DeviceSubscription": null,
                "LastInventoryTime": "2019-01-21 06:30:08.501",
                "LastStatusTime": "2019-01-21 06:30:02.492",
                "ManagedState": 3000,
                "Model": "PowerEdge MX7000",
                "PowerState": 17,
                "SlotConfiguration": {},
                "Status": 4000,
                "SystemId": 2031,
                "Type": 2000
            }
        ]
    }
```


Examples

```
- name: Retrieve basic inventory of all devices.
  dellemc_ome_device_facts:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"


- name: Retrieve basic inventory for devices identified by IDs 33333 or 11111 using filtering.
  dellemc_ome_device_facts:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    fact_subset: "basic_inventory"
    system_query_options:
      filter: "Id eq 33333 or Id eq 11111"


- name: Retrieve inventory details of specified devices identified by IDs 11111 and 22222.
  dellemc_ome_device_facts:
     hostname:  "192.168.0.1"
     username: "username"
     password: "password"
     fact_subset: "detailed_inventory"
     system_query_options:
       device_id:
        - 11111
        - 22222


- name: Retrieve inventory details of specified devices identified by service tags MXL1234
and MXL4567.
  dellemc_ome_device_facts:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    fact_subset: "detailed_inventory"
    system_query_options:
      device_service_tag:
        - MXL1234
        - MXL4567


- name: Retrieve details of specified inventory type of specified devices identified by ID
and service tags.
  dellemc_ome_device_facts:
    hostname: "192.168.0.1"
    username: "username"
```

```
    password: "password"
    fact_subset: "detailed_inventory"
    system_query_options:
      device_id:
        - 11111
      device_service_tag:
        - MXL1234
        - MXL4567
      inventory_type: "serverDeviceCards"


- name: Retrieve subsystem health of specified devices identified by service tags.
  dellemc_ome_device_facts:
    hostname:  "192.168.0.1"
    username:  "username"
    password:  "password"
    fact_subset: "subsystem_health"
    system_query_options:
      device_service_tag:
        - MXL1234
        - MXL4567
```

Manage device configuration templates

This section describes the specifications for template operations on devices managed by OME for hardware configuration and deployment operations.

Following are the tasks for managing device configuration templates:

1. View templates
2. Template operations
3. Attach or detach an identity pool
4. Set tagged and untagged vLANs

View templates

Module: ome_template_info

Synopsis

This module retrieves the list and details of all templates or details of a specific template.

Options

Table 57. ome_template_info

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
template_id	No	NA	NA	ID of the template.
system_query_options	No	NA	filter: Filter records by the supported values.	Provides the option to filter the output for the supported values. I(filter) query format must be aligned with OData standards.

Return Values

```
msg:
  type: str
  description: Overall template facts status.
  returned: on error
  sample: "Failed to fetch the template facts"
ansible_facts:
  type: dict
  description: Details of the templates.
  returned: success
  sample: {
        "192.168.0.1": {
            "CreatedBy": "system",
            "CreationTime": "1970-01-31 00:00:56.372144",
            "Description": "Tune workload for Performance Optimized Virtualization",
            "HasIdentityAttributes": false,
            "Id": 1,
            "IdentityPoolId": 0,
            "IsBuiltIn": true,
            "IsPersistencePolicyValid": false,
            "IsStatelessAvailable": false,
            "LastUpdatedBy": null,
            "LastUpdatedTime": "1970-01-31 00:00:56.372144",
            "Name": "iDRAC 14G Enable Performance Profile for Virtualization",
            "SourceDeviceId": 0,
            "Status": 0,
            "TaskId": 0,
            "TypeId": 2,
            "ViewTypeId": 4
        }
    }
```

Examples

```
- name: Retrieve basic details of all templates.
    ome_template_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"

- name: Retrieve details of a specific template identified by its template ID.
    ome_template_info
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    template_id: 1

- name: Get filtered template info based on name.
   ome_template_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    system_query_options:
     filter: "Name eq 'new template'"
```

Module: dellemc_ome_template_facts

Synopsis

This module retrieves the list and details of all templates or details of a specific template.

NOTE: This module is deprecated and replaced with ome_template_info.

Options

Table 58. dellemc_ome_template_facts

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname

Table 58. dellemc_ome_template_facts(continued)

Parameter	Required	Default	Choices	Comments
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
template_id	No	Na	Na	Unique ID of the template

Return Values

```
msg:
  type: str
  description: Over all template facts status.
  returned: on error
  sample: "Failed to fetch the template facts"
ansible_facts:
  type: dict
  description: Details of the templates.
  returned: success
  sample: {
        "192.168.0.1": {
            "CreatedBy": "system",
            "CreationTime": "1970-01-31 00:00:56.372144",
            "Description": "Tune workload for Performance Optimized Virtualization",
            "HasIdentityAttributes": false,
            "Id": 1,
            "IdentityPoolId": 0,
            "IsBuiltIn": true,
            "IsPersistencePolicyValid": false,
            "IsStatelessAvailable": false,
            "LastUpdatedBy": null,
            "LastUpdatedTime": "1970-01-31 00:00:56.372144",
            "Name": "iDRAC 14G Enable Performance Profile for Virtualization",
            "SourceDeviceId": 0,
            "Status": 0,
            "TaskId": 0,
            "TypeId": 2,
            "ViewTypeId": 4
        }
    }
```

Examples

```
- name: Retrieve basic details of all templates.
  dellemc_ome_template_facts:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"

- name: Retrieve details of a specific template identified by its template ID.
  dellemc_ome_template_facts:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    template_id: 1
```

Template operations

Module: ome_template

Synopsis

This module creates, modifies, deploys, deletes, exports, imports, or clones a template.

Options

Table 59. ome_template

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target device HTTPS port
command	False	create	create, modify, deploy, delete, export, import or clone.	• C(create) creates a new template. • C(modify) modifies an existing template. • C(deploy) creates a template-deployment job. • C(delete) deletes an existing template. • C(export) exports an existing template. • C(import) creates a template from a specified configuration text in SCP XML format. • C(clone) creates a clone of an existing template.
template_id	False	NA	NA	• ID of the existing template. • This option is applicable when I(command) is C(modify), C(deploy), C(delete) and C(export). • It is mutually exclusive with I(template_name).
template_name	False	NA	NA	• Name of the existing template. • This option is applicable when I(command) is C(modify), C(deploy), C(delete). • It is mutually exclusive with I(template_name).
device_id	False	[ ]	NA	• Specify the list of targeted device IDs when I(command) is C(deploy). When I (Command) is C(create), specify a single device ID. • Either I(device_id), or I(device_service_tag) can be used individually or together.
device_service_tag	False	[ ]	NA	• Specify the list of targeted device service tags when I (command) is C(deploy). When I(Command) is C(create), specify the service tag of a single device. • Either I(device_id), or I(device_service_tag) can be used individually or together.
template_view_type	False	Deployment	Deployment, Compliance, Inventory, Sample, or None	• Select the type of view of the OME template. • This is applicable when I(command) is C(create), C(clone), or C(import).
attributes	No	{ }	NA	Payload data for the template operations. All the variables in this option are added as payload for C(create), C(modify), C(deploy), C(import), and C(clone) operations. It takes the following attributes. • Name: Name of the template. This is mandatory when I(command) is C(create), C(import), C(clone), and optional when I(command) is C(modify). • Description for the template. This is applicable when I(command) is C(deploy) or C(modify). • Fqdds: This allows to create [Creates] a template using components from a specified reference server. One or more, of the following values must be specified in a comma-separated string: iDRAC, System, BIOS, NIC, LifeCycleController, RAID,

Table 59. ome_template(continued)

Parameter	Required	Default	Choices	Comments
				EventFilters, and All. If none of the values are specified, the default value 'All' is selected. This is applicable when I (command) is C(create). • Options: Allows to control device shutdown or end power state during template deployment. This is applicable when I(command) is C(deploy). • Schedule: Provides options to schedule the deployment task immediately, or at a specified time. This is applicable when I (command) is C(deploy). • NetworkBootIsoModel: Payload to specify the ISO deployment details. This is applicable when I(command) is C(deploy). • Attributes: List of dictionaries of attributes (if any) to be modified in the deployment template. This is applicable for when I(command) is C(deploy) and C(modify). • Content: The XML content of template. This is applicable when I(command) is C(import). • Type: Template type ID, indicating the type of device for which configuration is supported, such as chassis and servers. This is applicable when I(command) is C(import). • TypeId: Template type ID, indicating the type of device for which configuration is supported, such as chassis and servers. This is applicable when I(command) is C(create). **NOTE: See OpenManage Enterprise API Reference Guide for more details.**

Return Values

```
msg:
  description: Overall status of the template operation.
  returned: always
  type: str
  sample: "Successfully created a template with ID 123"
return_id:
  description: ID of the template used for C(create), C(modify), C(import), and C(clone) or
task created in case of C(deploy).
  returned: on success
  type: int
  sample: 12
TemplateId:
  description: ID of the template for C(export).
  returned: success, when I(command) is C(export)
  type: int
  sample: 13
Content:
  description: XML content of the exported template.
  returned: success, when I(command) is C(export)
  type: str
  sample: "<SystemConfiguration Model=\"PowerEdge R940\" ServiceTag=\"DG22TR2\" TimeStamp=
\"Tue Sep 24 09:20:57.872551
     2019\">\n<Component FQDD=\"AHCI.Slot.6-1\">\n<Attribute Name=\"RAIDresetConfig\">True</
Attribute>\n<Attribute
     Name=\"RAIDforeignConfig\">Clear</Attribute>\n</Component>\n<Component FQDD=
\"Disk.Direct.0-0:AHCI.Slot.6-1\">\n
     <Attribute Name=\"RAIDPDState\">Ready</Attribute>\n<Attribute Name=\"RAIDHotSpareStatus
\">No</Attribute>\n
     </Component>\n<Component FQDD=\"Disk.Direct.1-1:AHCI.Slot.6-1\">\n<Attribute Name=
\"RAIDPDState\">Ready
     </Attribute>\n<Attribute Name=\"RAIDHotSpareStatus\">No</Attribute>\n</Component>\n</
SystemConfiguration>\n"
```

```
error_info:
  description: Details of the HTTP Error.
  returned: on HTTP error
  type: dict
  sample: {
    "error": {
      "code": "Base.1.0.GeneralError",
      "message": "A general error has occurred. See ExtendedInfo for more information.",
      "@Message.ExtendedInfo": [
        {
          "MessageId": "GEN1234",
          "RelatedProperties": [],
          "Message": "Unable to process the request because an error occurred.",
          "MessageArgs": [],
          "Severity": "Critical",
          "Resolution": "Retry the operation. If the issue persists, contact your system
administrator."
        }
      ]
    }
  }
```

Examples

```
- name: Create a template from a reference device.
  ome_template:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_id: 25123
    attributes:
      Name: "New Template"
      Description: "New Template description"

- name: Modify template name, description, and attribute value.
  ome_template:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    command: "modify"
    template_id: 12
    attributes:
      Name: "New Custom Template"
      Description: "Custom Template Description"
      # Attributes to be modified in the template.
      # For information on any attribute ID, use API /TemplateService/Templates(Id)/Views(Id)/
AttributeViewDetails
      # This section is optional
      Attributes:
        - Id: 1234
          Value: "Test Attribute"
          IsIgnored: false

- name: Deploy template on multiple devices.
  ome_template:
    hostname:  "192.168.0.1"
    username: "username"
    password: "password"
    command: "deploy"
    template_id: 12
    device_id:
      - 12765
      - 10173
    device_service_tag:
      - 'SVTG123'
      - 'SVTG456'

- name: Deploy template on multiple devices along with the attribute values to be modified on
the target devices.
  ome_template:
    hostname:  "192.168.0.1"
    username: "username"
    password: "password"
```

```
    command: "deploy"
    template_id: 12
    device_id:
      - 12765
      - 10173
    device_service_tag:
      - 'SVTG123'
    attributes:
      # Device specific attributes to be modified during deployment.
      # For information on any attribute id, use API /TemplateService/Templates(Id)/Views(Id)/
AttributeViewDetails
      # This section is optional
      Attributes:
        # The device where attribute to be modified during deployment runtime.
        # The Device ID should be mentioned above in the 'device_id' section.
        # Service tags not allowed.
        - DeviceId: 12765
          Attributes:
            - Id : 15645
              Value : "0.0.0.0"
              IsIgnored : false
        - DeviceId: 10173
          Attributes:
            - Id : 18968,
              Value : "hostname-1"
              IsIgnored : false

- name: Deploy template and Operating System (OS) on multiple devices.
  ome_template:
    hostname:  "192.168.0.1"
    username: "username"
    password: "password"
    command: "deploy"
    template_id: 12
    device_id:
      - 12765
    device_service_tag:
      - 'SVTG123'
    attributes:
      # Include this to install OS on the devices.
      # This section is optional
      NetworkBootIsoModel:
        BootToNetwork: false
        ShareType: "NFS"
        IsoPath: "/home/iso_path/filename.iso"
        ShareDetail:
          IpAddress: "192.168.0.2"
          ShareName: "sharename"
          User: "share_user"
          Password: "share_password"
      Options:
        EndHostPowerState: 1
        ShutdownType: 0
        TimeToWaitBeforeShutdown: 300
      Schedule:
        RunLater: true
        RunNow: false

- name: Deploy template on multiple devices and change the device-level attributes. After the
template is deployed, install OS using its image.
  ome_template:
    hostname:  "192.168.0.1"
    username: "username"
    password: "password"
    command: "deploy"
    template_id: 12
    device_id:
      - 12765
      - 10173
    device_service_tag:
      - 'SVTG123'
      - 'SVTG456'
    attributes:
```

```
      Attributes:
        - DeviceId: 12765
          Attributes:
           - Id : 15645
             Value : "0.0.0.0"
             IsIgnored : false
        - DeviceId: 10173
          Attributes:
            - Id : 18968,
              Value : "hostname-1"
              IsIgnored : false
      NetworkBootIsoModel:
        BootToNetwork: false
        ShareType: "NFS"
        IsoPath: "/home/iso_path/filename.iso"
        ShareDetail:
          IpAddress: "192.168.0.2"
          ShareName: "sharename"
          User: "share_user"
          Password: "share_password"
      Options:
        EndHostPowerState: 1
        ShutdownType: 0
        TimeToWaitBeforeShutdown: 300
      Schedule:
        RunLater: true
        RunNow: false

- name: Delete a template.
  ome_template:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    command: "delete"
    template_id: 12

- name: Export a template.
  ome_template:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    command: "export"
    template_id: 12

- name: Export template to local xml file
  ome_template:
    hostname: "{{hostname}}"
    username: "{{username}}"
    password: "{{password}}"
    command: "export"
    template_name: "my_template"
  register: result
  tags:
    - export_xml_to_file
- copy:
    content: "{{ result.Content}}"
    dest: "/path/to/exported_template.xml"
  tags:
    - export_xml_to_file

- name: Clone a template.
  ome_template:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    command: "clone"
    template_id: 12
    attributes:
      Name: "New Cloned Template Name"

- name: Import template from XML content.
  ome_template:
    hostname: "192.168.0.1"
```

```
    username: "username"
    password: "password"
    command: "import"
    attributes:
      Name: "Imported Template Name"
      # Template Type from TemplateService/TemplateTypes
      Type: 2
      # xml string content
      Content: "<SystemConfiguration Model=\"PowerEdge R940\" ServiceTag=\"SVCTAG1\"
      TimeStamp=\"Tue Sep 24 09:20:57.872551 2019\">\n<Component FQDD=\"AHCI.Slot.6-1\">
\n<Attribute
      Name=\"RAIDresetConfig\">True</Attribute>\n<Attribute Name=\"RAIDforeignConfig\">Clear</
Attribute>\n
      </Component>\n<Component FQDD=\"Disk.Direct.0-0:AHCI.Slot.6-1\">\n<Attribute Name=
\"RAIDPDState\">Ready
      </Attribute>\n<Attribute Name=\"RAIDHotSpareStatus\">No</Attribute>\n</Component>\n
      <Component FQDD=\"Disk.Direct.1-1:AHCI.Slot.6-1\">\n<Attribute Name=\"RAIDPDState
\">Ready</Attribute>\n
      <Attribute Name=\"RAIDHotSpareStatus\">No</Attribute>\n</Component>\n</
SystemConfiguration>\n"
      Description: "Imported Template description"

- name: Import template from local XML file.
  ome_template:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    command: "import"
    attributes:
      name: "Imported Template Name"
      Type: 2
      Content: "{{ lookup('file', '/path/to/xmlfile') }}"
```

Module: dellemc_ome_template

Synopsis

This module creates, modifies or deploys a template.

NOTE: This module is deprecated and replaced with ome_template.

Options

Table 60. dellemc_ome_template

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
state	No	create	· create · modify · deploy	· C(create) creates a new template. · C(modify) modifies an existing template. · C(deploy) deploys an existing template.
template_id	No	NA	NA	Unique ID of the template to be modified or deployed. This option is mandatory for C(modify) and C(deploy) operations.
device_id	No	[ ]	NA	List of targeted device id(s) for C(deploy) or a single id for C(create) operation. Either I(device_id) or I(device_service_tag) is mandatory or both can be applicable.
device_service_tag	No	[ ]	NA	List of targeted device service tag(s) for C(deploy) or a single service tag for C(create) operation. Either I(device_id) or I(device_service_tag) is mandatory or both can be applicable.

Table 60. dellemc_ome_template(continued)

Parameter	Required	Default	Choices	Comments
template_view_type	No	Deployment	• Deployment, • Compliance • Inventory • Sample • None	The features that support template operations. This is applicable only for C(create) operation.
attributes	No	{ }	NA	• Name: Name of the template. This is mandatory for C(create) and C(modify) operations. • Description: Description of the template. This is applicable for C(create) and C(modify) operations. • Fqdds: This provides functionality to copy only certain areas of system configuration from the specified reference server. One or more of the following values may be specified in a comma-separated string: iDRAC, System, BIOS, NIC, LifeCycleController, RAID, EventFilters, All. Default value is 'All'. This is applicable for C(create) operation. • Options: Options to control device shutdown or end power state during template deployment. This is applicable for C(deploy) operation. • Schedule: Options to schedule the deployment task immediately or at a specified time. This is applicable for C(deploy) operation. • NetworkBootIsoModel: Payload to specify the ISO deployment details. This is applicable for C(deploy) operation. • Attributes: list of dictionaries of attribute values (if any) to be modified in the template to be deployed. This is applicable for C(deploy) operation. **NOTE: See OpenManage Enterprise API Reference Guide for more details.**

Return Values

```
msg:
  description: Overall status of the template operation.
  returned: always
  type: str
  sample: "Successfully created a Template with id 123"
return_id:
  description: id of the template for C(create) and C(modify) or task created in case of
C(deploy)
  returned: success
  type: int
  sample: 124
template_status:
  description: Details of the HTTP Error.
  returned: on HTTP error
  type: dict
  sample: {
    "error": {
      "code": "Base.1.0.GeneralError",
      "message": "A general error has occurred. See ExtendedInfo for more information.",
      "@Message.ExtendedInfo": [
        {
          "MessageId": "GEN1234",
          "RelatedProperties": [],
          "Message": "Unable to process the request because an error occurred.",
          "MessageArgs": [],
          "Severity": "Critical",
          "Resolution": "Retry the operation. If the issue persists, contact your system
administrator."
        }
```

```
        ]
      }
    }
```

Examples

```
- name: create template.
  dellemc_ome_template:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_id: 25123
    attributes:
      Name: "New Template"
      Description: "New Template description"


- name: modify template
  dellemc_ome_template:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    state: "modify"
    template_id: 1234
    attributes:
      Name: "New Custom Template"
      Description: "Custom Template Description"


- name: deploy template.
  dellemc_ome_template:
    hostname:  "192.168.0.1"
    username: "username"
    password: "password"
    state: "deploy"
    template_id: 1234
    device_id:
      - 12345
      - 45678
    device_service_tag: ['SVTG123', 'SVTG456']
    attributes:
      NetworkBootIsoModel:
        BootToNetwork: false
        ShareType: "NFS"
        IsoPath: "bootToIsoPath.iso"
        ShareDetail:
          IpAddress: "192.168.0.2"
          ShareName: "/nfsshare"
          User: null
          Password: null
      Attributes:
        - Id: 1234
          Value: "Test Attribute"
          IsIgnored: false
      Options:
        EndHostPowerState: 1
        ShutdownType: 0
        TimeToWaitBeforeShutdown: 300
      Schedule:
        RunLater: true
        RunNow: false
```

Attach or detach an identity pool

ome_template_identity_pool

Synopisis

This module allows to-

- Attach an identity pool to a requested template.

- Detach an identity pool from a requested template.

Options

Table 61. ome_template_identity_pool

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
template_name	True	NA	NA	Name of the template to which an identity pool is attached to or detached from.
identity_pool_name	False	NA	NA	Name of the identity pool. - To attach an identity pool to a template, provide the name of the identity pool. - This option is not applicable when detaching an identity pool from a template.

Return Values

```
msg:
  type: str
  description: Overall identity pool status of the attach or detach operation.
  returned: always
  sample: Successfully attached identity pool to template.
error_info:
  description: Details of the HTTP Error.
  returned: on HTTP error
  type: dict
  sample: {
    "error": {
      "code": "Base.1.0.GeneralError",
      "message": "A general error has occurred. See ExtendedInfo for more information.",
      "@Message.ExtendedInfo": [
        {
          "MessageId": "GEN1234",
          "RelatedProperties": [],
          "Message": "Unable to process the request because an error occurred.",
          "MessageArgs": [],
          "Severity": "Critical",
          "Resolution": "Retry the operation. If the issue persists, contact your system
administrator."
        }
      ]
    }
  }
```

Examples

```
- name: Attach an identity pool to a template.
  ome_template_identity_pool:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    template_name: template_name
```

```
    identity_pool_name: identity_pool_name

- name: Detach an identity pool from a template.
  ome_template_identity_pool:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    template_name: template_name
```

Set tagged and untagged vLANs in a template

ome_template_network_vlan

Synopsis

This module lets you select tagged and untagged vLANs to be used in the OpenManage Enterprise template.

Options

Table 62. ome_template_network_vlan

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname.
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
template_name	False	NA	NA	• Name of the template • It is mutually exclusive with l(template_id).
template_id	False	NA	NA	• ID of the template • It is mutually exclusive with l(template_name).
nic_identifier	True	NA	NA	Display name of the NIC port in the template for vLAN configuration.
untagged_networks	False	NA	NA	List of untagged networks and their corresponding NIC ports. Suboptions- • port- NIC port number of the untagged vLAN • untagged_network_id- ○ ID of the untagged vLAN ○ Enter 0 to clear the untagged vLAN from the port. ○ This option is mutually exclusive with l(untagged_network_name). ○ To get the vLAN network ID use the API U(https://l(hostname)/api/NetworkConfigurationService/Networks) • untagged_network_name- ○ Name of the untagged vLAN ○ Enter 0 to clear the untagged vLAN from the port. ○ This option is mutually exclusive with l(untagged_network_id).

Table 62. ome_template_network_vlan(continued)

Parameter	Required	Default	Choices	Comments
tagged_networks	False	NA	NA	List of tagged vLANs and their corresponding NIC ports. Suboptions- • port- NIC port number of the tagged vLAN • tagged_network_ids- ○ List of IDs of the tagged vLANs ○ Enter [] to remove the tagged VLAN from a port. ○ List of I(tagged_network_ids) is combined with list of I(tagged_network_names) when adding tagged vLANs to a port. ○ To get the vLAN network ID use the API U(https://I(hostname)/api/NetworkConfigurationService/Networks) • tagged_network_names- ○ List of names of tagged vLANs ○ Enter [] to remove the tagged VLAN from a port. ○ List of I(tagged_network_names) is combined with list of I(tagged_network_ids) when adding tagged vLANs to a port.

Return Values

```
msg:
  type: str
  description: Overall status of the template vlan operation.
  returned: always
  sample: "Successfully applied the network settings to template"
error_info:
  description: Details of the HTTP Error.
  returned: on HTTP error
  type: dict
  sample:  {
      "error": {
                "@Message.ExtendedInfo": [{
                "Message": "Unable to process the request because an error occurred:",
                "MessageArgs": "",
                "MessageId": "CGEN6001",
                "RelatedProperties": [],
                "Resolution": "Retry the operation. If the issue persists, contact your system
administrator.",
                "Severity": "Critical"
                }],
      "code": "Base.1.0.GeneralError",
      "message": "A general error has occurred. See ExtendedInfo for more information."
      }}
'''
```

Examples

```
- name: Add tagged or untagged vLANs to a template using vLAN ID and name.
  ome_template_network_vlan:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    template_id: 78
    nic_identifier: NIC Slot 4
```

```
    untagged_networks:
      - port: 1
        untagged_network_id: 127656
      - port: 2
        untagged_network_name: vlan2
    tagged_networks:
      - port: 1
        tagged_network_ids:
          - 12767
          - 12768
      - port: 4
        tagged_network_ids:
          - 12767
          - 12768
        tagged_network_names:
          - vlan3
      - port: 2
        tagged_network_names:
          - vlan4
          - vlan1

- name: Clear the tagged and untagged vLANs from a template.
  ome_template_network_vlan:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    template_id: 78
    nic_identifier: NIC Slot 4
    untagged_networks:
      - port: 1
        untagged_network_id: 0
      - port: 2
        untagged_network_name: 0
    tagged_networks:
      - port: 1
        tagged_network_ids: []
      - port: 4
        tagged_network_ids: []
        tagged_network_names: []
      - port: 2
        tagged_network_names: []
```

Manage the device firmware

This section describes the following firmware processes that can be carried out on the devices managed by OME, using OpenManage Ansible Modules-

- Update device firmware.
- Create a firmware catalog.
- Create a firmware baseline.
- Retrieve baseline compliance details.

Update device firmware

Module: ome_firmware

Synopsis

This module updates the firmware of PowerEdge devices and all its components.

Options

Table 63. ome_firmware

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username

Table 63. ome_firmware(continued)

Parameter	Required	Default	Choices	Comments
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
device_service_tag	False	NA	NA	• List of targeted device service tags. • Either I(device_id) or I(device_service_tag) can be used individually or together. • I(device_service_tag) is mutually exclusive with I(device_group_names).
device_id	False	NA	NA	• List of targeted device ids. • Either I(device_id), or I(device_service_tag) can be used individually or together. • I(device_id) is mutually exclusive with I(device_group_names).
device_group_names	False	NA	NA	• Enter the name of the group to update the firmware of all the devices within the group. • I(device_group_names) is mutually exclusive with I(device_id) and I(device_service_tag).
basline_name	False	NA	NA	• Enter the baseline name to update the firmware of all the devices or groups of devices against the available compliance report. • The firmware update can also be done by providing the baseline name and the path to the single DUP file. To update multiple baselines at once, provide the baseline names separated by commas. • I(baseline_names) is mutually exclusive with I(device_group_names), I(device_id) and I(device_service_tag).
dup_file	False	NA	NA	Executable file to apply on the targets.

Return Values

```
msg:
  type: str
  description: Overall firmware update status.
  returned: always
  sample: "Successfully submitted the firmware update job."
update_status:
  type: dict
  description: Firmware Update job and progress details from the OME.
  returned: success
  sample: {
    'LastRun': None,
    'CreatedBy': 'user',
    'Schedule': 'startnow',
    'LastRunStatus': {
      'Id': 1111,
      'Name': 'NotRun'
    },
    'Builtin': False,
    'Editable': True,
    'NextRun': None,
    'JobStatus': {
      'Id': 1111,
      'Name': 'New'
    },
    'JobName': 'Firmware Update Task',
    'Visible': True,
    'State': 'Enabled',
```

```
    'JobDescription': 'dup test',
    'Params': [{
      'Value': 'true',
      'Key': 'signVerify',
      'JobId': 11111}, {
      'Value': 'false',
      'Key': 'stagingValue',
      'JobId': 11112}, {
      'Value': 'false',
      'Key': 'complianceUpdate',
      'JobId': 11113}, {
      'Value': 'INSTALL_FIRMWARE',
      'Key': 'operationName',
      'JobId': 11114}],
    'Targets': [{
      'TargetType': {
      'Id': 1000,
      'Name': 'DEVICE'},
      'Data': 'DCIM:INSTALLED#701__NIC.Mezzanine.1A-1-1=1111111111111',
      'Id': 11115,
      'JobId': 11116}],
    'StartTime': None,
    'UpdatedBy': None,
    'EndTime': None,
    'Id': 11117,
    'JobType': {
      'Internal': False,
      'Id': 5,
      'Name': 'Update_Task'}
}
error_info:
  description: Details of the HTTP Error.
  returned: on HTTP error
  type: dict
  sample: {
    "error": {
      "code": "Base.1.0.GeneralError",
      "message": "A general error has occurred. See ExtendedInfo for more information.",
      "@Message.ExtendedInfo": [
      {
       "MessageId": "GEN1234",
       "RelatedProperties": [],
       "Message": "Unable to process the request because an error occurred.",
       "MessageArgs": [],
       "Severity": "Critical",
       "Resolution":"Retry the operation. If the issue persists, contact your system
administrator."
      }
]
```

Examples

```
- name: Update firmware from a DUP file using device ids.
  dellemc_ome_firmware:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_id:
      - 11111
      - 22222
    dup_file: "/path/Chassis-System-Management_Firmware_6N9WN_WN64_1.00.01_A00.EXE"

- name: Update firmware from a DUP file using device service tags.
  dellemc_ome_firmware:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_service_tag:
      - KLBR111
      - KLBR222
   dup_file: "/path/Network_Firmware_NTRW0_WN64_14.07.07_A00-00_01.EXE"

-name: Update firmware from a DUP file using a device group name.
```

```
  ome_firmware:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_group_names:
      - servers
    dup_file: "/path/BIOS_87V69_WN64_2.4.7.EXE"

-name: Update firmware using a baseline name.
  ome_firmware:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    baseline_name: baseline_devices
```

Update device firmware

Module: dellemc_ome_firmware

Synopsis

This module updates the device firmware and all its components.

NOTE: This module is deprecated and replaced with ome_firmware.

Options

Table 64. dellemc_ome_firmware

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target HTTPS port
device_service_tag	No	NA	NA	List of targeted device service tags.
device_id	No	NA	NA	List of targeted device ids.
dup_file	Yes	NA	NA	Executable file to apply on the targets.

Return Values

```
msg:
  type: str
  description: "Overall firmware update status."
  returned: always
  sample: "Successfully updated the firmware."
update_status:
  type: dict
  description: "Firmware Update job and progress details from the OME."
  returned: success
  sample: {
    'LastRun': None,
    'CreatedBy': 'user',
    'Schedule': 'startnow',
    'LastRunStatus': {
      'Id': 1111,
      'Name': 'NotRun'
    },
    'Builtin': False,
    'Editable': True,
    'NextRun': None,
```

```
    'JobStatus': {
      'Id': 1111,
      'Name': 'New'
    },
    'JobName': 'Firmware Update Task',
    'Visible': True,
    'State': 'Enabled',
    'JobDescription': 'dup test',
    'Params': [{
      'Value': 'true',
      'Key': 'signVerify',
      'JobId': 11111}, {
      'Value': 'false',
      'Key': 'stagingValue',
      'JobId': 11112}, {
      'Value': 'false',
      'Key': 'complianceUpdate',
      'JobId': 11113}, {
      'Value': 'INSTALL_FIRMWARE',
      'Key': 'operationName',
      'JobId': 11114}],
    'Targets': [{
      'TargetType': {
      'Id': 1000,
      'Name': 'DEVICE'},
      'Data': 'DCIM:INSTALLED#701__NIC.Mezzanine.1A-1-1=1111111111111',
      'Id': 11115,
      'JobId': 11116}],
    'StartTime': None,
    'UpdatedBy': None,
    'EndTime': None,
    'Id': 11117,
    'JobType': {
      'Internal': False,
      'Id': 5,
      'Name': 'Update_Task'}
}
```

Examples

```
- name: "Update firmware from DUP file using device ids."
  dellemc_ome_firmware:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_id:
      - 11111
      - 22222
    dup_file: "/path/Chassis-System-Management_Firmware_6N9WN_WN64_1.00.01_A00.EXE"


- name: "Update firmware from DUP file using device service tags."
  dellemc_ome_firmware:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_service_tag:
      - KLBR111
      - KLBR222
   dup_file: "/path/Network_Firmware_NTRW0_WN64_14.07.07_A00-00_01.EXE"
```

Create a firmware catalog

Module: ome_firmware_catalog

Synopsis

This module triggers the job to create a catalog.

Options

Table 65. ome_firmware_catalog

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
catalog_name	True	NA	NA	Name of the firmware catalog being created.
catalog_description	False	NA	NA	Description of the catalog being created.
source	False	NA	NA	The share address of the system where the firmware catalog is stored on the network.
source_path	False	NA	NA	Full path of the catalog file location excluding the file name.
file_name	False	NA	NA	Catalog file name relative to the l (source_path).
repository_type	False	HTTPS	HTTP, NFS, CIFS, HTTPS	The type of supported repositories are: HTTP, NFS, CIFS, HTTPS.
repository_username	False	NA	NA	User name of the repository where the catalog is stored. This option is mandatory when l(repository_type) is CIFS.
repository_password	False	NA	NA	Password to access the repository. This option is mandatory when l(repository_type) is CIFS.
repository_domain	False	NA	NA	Domain name of the repository.
check_certificate	False	False	NA	Specifies if certificate warnings are ignored when l(repository_type) is HTTPS. If C(True) option is set, then the certificate warnings are ignored otherwise certificate warnings are not ignored.

Return Values

```
msg:
          description: Overall status of the firmware catalog creation
          returned: always
          type: str
          sample: "Successfully triggered the job to create a catalog with Task Id : 10094"
        catalog_status:
          description: Details of the catalog creation.
          returned: on success
          type: dict
          sample:  {
                "AssociatedBaselines": [],
                "BaseLocation": null,
                "BundlesCount": 0,
```

```
                "Filename": "catalog.gz",
                "Id": 0,
                "LastUpdated": null,
                "ManifestIdentifier": null,
                "ManifestVersion": null,
                "NextUpdate": null,
                "PredecessorIdentifier": null,
                "ReleaseIdentifier": null,
                "Repository": {
                   "CheckCertificate": true,
                   "Description": "HTTPS Desc",
                   "DomainName": null,
                   "Id": null,
                   "Name": "catalog4",
                   "Password": null,
                   "RepositoryType": "HTTPS",
                   "Source": "company.com",
                   "Username": null
                },
                "Schedule": null,
                "SourcePath": "catalog",
                "Status": null,
                "TaskId": 10094
            }
error_info:
  type: dict
  description: Details of http error.
  returned: on http error
  sample:  {
        "error": {
            "@Message.ExtendedInfo": [
                {
                    "Message": "Unable to create or update the catalog because a
                    repository with the same name already exists.",
                    "Resolution": "Enter a different name and retry the operation.",
                    "Severity": "Critical"
                }
             ],
             "code": "Base.1.0.GeneralError",
             "message": "A general error has occurred. See ExtendedInfo for more information."
         }
     }
```

Examples

```
- name: create catalog from a repository on a HTTPS.
  ome_firmware_catalog:
    hostname: "192.168.0.1"
    username: "username"
    catalog_name: "catalog_name"
    catalog_description: "catalog_description"
    repository_type: "HTTPS"
    source: "downloads.dell.com"
    source_path: "catalog"
    file_name: "catalog.gz"
    check_certificate: True

- name: create catalog from a repository on a HTTP.
  ome_firmware_catalog:
    hostname: "192.168.0.1"
    username: "username"
    catalog_name: "catalog_name"
    catalog_description: "catalog_description"
    repository_type: "HTTP"
    source: "downloads.dell.com"
    source_path: "catalog"
    file_name: "catalog.gz"

- name: create catalog from a CIFS network share.
  ome_firmware_catalog:
    hostname: "192.168.0.1"
    username: "username"
    catalog_name: "catalog_name"
```

```
    catalog_description: "catalog_description"
    repository_type: "CIFS"
    source: "192.167.0.1"
    source_path: "cifs/R940"
    file_name: "catalog.gz"
    repository_username: "repository_username"
    repository_password: "repository_password"
    repository_domain: "repository_domain"

- name: create catalog from a NFS network share.
  ome_firmware_catalog:
    hostname: "192.168.0.1"
    username: "username"
    catalog_name: "catalog_name"
    catalog_description: "catalog_description"
    repository_type: "NFS"
    source: "192.166.0.2"
    source_path: "/nfs/R940"
    file_name: "catalog.xml"
```

Create a firmware baseline

Module: ome_firmware_baseline

Synopsis

This module allows to create a baseline.

Options

Table 66. ome_firmware_baseline

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target share address or hostname.
username	True	NA	NA	Target username.
password	True	NA	NA	Target user password.
port	False	443	NA	Target HTTPS port.
baseline_name	True	NA	NA	Name of the baseline being created.
baseline_description	False	NA	NA	Description of the baseline being created.
catalog_name	False	NA	NA	Name of the catalog associated with the baseline.
downgrade_enabled	False	True	NA	Indicates if a downgrade is allowed or not.
is_64_bit	False	True	NA	Indicates if 64 bit is supported.
device_ids	False	NA	NA	List olf device ids. I(device_ids) is mutually exclusive with I(device_service_tags) and I(device_group_names).
device_service_tags	False	NA	NA	List of service tags I(device_service_tags) is mutually exclusive with I(device_ids) and I(device_group_names).
device_group_names	False	NA	NA	List of group names. I(device_group_names) is

Table 66. ome_firmware_baseline(continued)

Parameter	Required	Default	Choices	Comments
				mutually exclusive with l(device_ids) and l(device_service_tags).

Return Values

```
msg:
  description: Overall status of the firmware baseline creation
  returned: always
  type: str
  sample: "Successfully created task for creating Baseline"
baseline_status:
  description:
  returned: success
  type: dict
  sample: {
    "CatalogId": 123,
    "Description": "BASELINE DESCRIPTION",
    "DeviceComplianceReports": [],
    "DowngradeEnabled": true,
    "Id": 0,
    "Is64Bit": true,
    "Name": "my_baseline",
    "RepositoryId": 123,
    "RepositoryName": "catalog123",
    "RepositoryType": "HTTP",
    "Targets": [
        {
            "Id": 10083,
            "Type": {
                "Id": 1000,
                "Name": "DEVICE"
            }
        },
        {
            "Id": 10076,
            "Type": {
                "Id": 1000,
                "Name": "DEVICE"
            }
        }
    ],
    "TaskId": 11235,
    "TaskStatusId": 0
  }
error_info:
  type: dict
  description: Details of http error.
  returned: on http error
  sample:  {
        "error": {
            "@Message.ExtendedInfo": [
                {
                    "Message": "Unable to retrieve baseline list either because the device
ID(s) entered are invalid",
                    "Resolution": "Make sure the entered device ID(s) are valid and retry the
operation.",
                    "Severity": "Critical"
                }
            ],
            "code": "Base.1.0.GeneralError",
            "message": "A general error has occurred. See ExtendedInfo for more information."
        }
    }
```

Examples

```
- name: create baseline from device Ids.
  ome_firmware_baseline:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    baseline_name: "baseline_name"
    baseline_description: "baseline_description"
    catalog_name: "catalog_name"
    device_ids:
      - 1010
      - 2020

- name: create baseline from device service tags.
  ome_firmware_baseline:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    baseline_name: "baseline_name"
    baseline_description: "baseline_description"
    catalog_name: "catalog_name"
    device_service_tags:
      - "SVCTAG1"
      - "SVCTAG2"

- name: create baseline from device group names.
  ome_firmware_baseline:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    baseline_name: "baseline_name"
    baseline_description: "baseline_description"
    catalog_name: "catalog_name"
    device_group_names:
      - "Group1"
      - "Group2"
```

Retrieve firmware baseline compliance details

Module: ome_firmware_baseline_compliance_info

Synopsis

This module allows to retrieve firmware compliance for a list of devices, or against a specified baseline.

Options

Table 67. ome_firmware_baseline_compliance_info

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target share address or hostname.
username	True	NA	NA	Target username.
password	True	NA	NA	Target user password.
port	False	443	NA	Target HTTPS port.
baseline_name	False	NA	NA	• Name of the baseline for which the device based compliance report is generated. • This option is mandatory for generating baseline based device compliance report.

Table 67. ome_firmware_baseline_compliance_info(continued)

Parameter	Required	Default	Choices	Comments
				• l(baseline_name) is mutually exclusive with l(device_ids), l(device_service_tags), and l(device_group_names).
device_ids	False	NA	NA	• A list of unique identifiers for which the device based compliance report is generated. • Either l(device_ids), l(device_service_tags), or l(device_group_names) is required to generate device based compliance report. • l(device_ids) is mutually exclusive with l (device_service_tags),l(device_group_names), and l(baseline_name). • Devices without reports are ignored.
device_service_tags	False	NA	NA	• A list of service tags for which the device based compliance report is generated. • Either l(device_ids), l(device_service_tags), or l(device_group_names) is required to generate device based compliance report. • l(device_service_tags) is mutually exclusive with l(device_ids), l(device_group_names), and l(baseline_name). • Devices without reports are ignored.
device_group_names	False	NA	NA	• A list of group names for which the device based compliance report is generated. • Either l(device_ids), l(device_service_tags), or l(device_group_names) is required to generate device based compliance report. • l(device_group_names) is mutually exclusive with l(device_ids), l(device_service_tags), and l(baseline_name). • Devices without reports are ignored.

Return values

```
msg:
  type: str
  description: Overall baseline compliance report status.
  returned: on error
  sample: "Failed to fetch the compliance baseline information"
baseline_compliance_info:
  type: dict
  description: Details of the baseline compliance report.
  returned: success
  sample: [
            {
                "CatalogId": 53,
                "ComplianceSummary": {
                    "ComplianceStatus": "CRITICAL",
                    "NumberOfCritical": 2,
                    "NumberOfDowngrade": 0,
                    "NumberOfNormal": 0,
                    "NumberOfWarning": 0
                },
                "Description": "",
                "DeviceComplianceReports": [
                    {
                        "ComplianceStatus": "CRITICAL",
                        "ComponentComplianceReports": [
                            {
                                "ComplianceDependencies": [],
                                "ComplianceStatus": "DOWNGRADE",
                                "Criticality": "Ok",
                                "CurrentVersion": "OSC_1.1",
                                "Id": 1258,
                                "ImpactAssessment": "",
                                "Name": "OS COLLECTOR 2.1",
                                "Path": "FOLDER04118304M/2/
Diagnostics_Application_JCCH7_WN64_4.0_A00_01.EXE",
                                "PrerequisiteInfo": "",
                                "RebootRequired": false,
                                "SourceName": "DCIM:INSTALLED#802__OSCollector.Embedded.1",
                                "TargetIdentifier": "101734",
                                "UniqueIdentifier": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                                "UpdateAction": "DOWNGRADE",
                                "Uri": "http://www.dell.com/support/home/us/en/19/Drivers/
DriversDetails?driverId=XXXXX",
                                "Version": "4.0"
                            },
                            {
                                "ComplianceDependencies": [],
                                "ComplianceStatus": "CRITICAL",
                                "Criticality": "Recommended",
                                "CurrentVersion": "DN02",
                                "Id": 1259,
                                "ImpactAssessment": "",
                                "Name": "TOSHIBA AL14SE 1.8 TB 2.5 12Gb 10K 512n SAS HDD
Drive",
                                "Path": "FOLDER04086111M/1/SAS-
Drive_Firmware_VDGFM_WN64_DN03_A00.EXE",
                                "PrerequisiteInfo": "",
                                "RebootRequired": true,
                                "SourceName":
"DCIM:INSTALLED#304_C_Disk.Bay.1:Enclosure.Internal.0-1:RAID.Integrated.1-1",
                                "TargetIdentifier": "103730",
                                "UniqueIdentifier": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
                                "UpdateAction": "UPGRADE",
                                "Uri": "http://www.dell.com/support/home/us/en/19/Drivers/
DriversDetails?driverId=XXXXX",
                                "Version": "DN03"
                            }
                        ],
                        "DeviceId": 11603,
                        "DeviceModel": "PowerEdge R630",
                        "DeviceName": null,
                        "DeviceTypeId": 1000,
```

```
                        "DeviceTypeName": "CPGCGS",
                        "FirmwareStatus": "Non-Compliant",
                        "Id": 194,
                        "RebootRequired": true,
                        "ServiceTag": "MXL1234"
                    }
                ],
                "DowngradeEnabled": true,
                "Id": 53,
                "Is64Bit": false,
                "LastRun": "2019-09-27 05:08:16.301",
                "Name": "baseline1",
                "RepositoryId": 43,
                "RepositoryName": "catalog2",
                "RepositoryType": "CIFS",
                "Targets": [
                    {
                        "Id": 11603,
                        "Type": {
                            "Id": 1000,
                            "Name": "DEVICE"
                        }
                    }
                ],
                "TaskId": 11710,
                "TaskStatusId": 0
            }
        ]
error_info:
  type: dict
  description: Details of http error.
  returned: on http error
  sample:  {
        "error": {
            "@Message.ExtendedInfo": [
                {
                    "Message": "Unable to retrieve baseline list either because the device
ID(s) entered are invalid",
                    "Resolution": "Make sure the entered device ID(s) are valid and retry the
operation.",
                    "Severity": "Critical"
                }
            ],
            "code": "Base.1.0.GeneralError",
            "message": "A general error has occurred. See ExtendedInfo for more information."
        }
    }
```

Examples

```
- name: Retrieves baseline based compliance report for specific device IDs.
  ome_firmware_baseline_compliance_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_ids:
        - 11111
        - 22222

- name: Retrieves device based compliance report for specific device service Tags.
  ome_firmware_baseline_compliance_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_service_tags:
        - MXL1234
        - MXL4567
```

```
- name: Retrieves device based compliance report for specific group names.
  ome_firmware_baseline_compliance_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_group_names:
        - "group1"
        - "group2"

- name: Retrieves device compliance report for a specific baseline.
  ome_firmware_baseline_compliance_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    baseline_name: "baseline_name"
```

Manage jobs

This section describes the modules using which you can manage job operations.

Following are the tasks for managing jobs:

- View job details
- Manage power state operations

View job details

Module: dellemc_ome_job_facts

Synopsis

This module retrieves job details for a given job ID or the entire job queue.

Options

Table 68. dellemc_ome_job_facts

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target HTTPS port
job_id	No	NA	NA	Unique ID of the job
system_query_options	No	NA	• top: Number of records to return. Default value is 100. • skip: Number of records to skip. Default value is 0. • filter: Filter records by the values supported.	Options for pagination of the output

Return Values

```
msg:
  description: Overall status of the job facts operation.
  returned: always
  type: str
job_facts:
  description: Details of the OpenManage Enterprise jobs.
  returned: success
  type: dict
  sample: {
    "value": [
```

```
    {
      "Builtin": false,
      "CreatedBy": "system",
      "Editable": true,
      "EndTime": null,
      "Id": 12345,
      "JobDescription": "Refresh Inventory for Device",
      "JobName": "Refresh Inventory for Device",
      "JobStatus": {
        "Id": 2080,
        "Name": "New"
      },
      "JobType": {
        "Id": 8,
        "Internal": false,
        "Name": "Inventory_Task"
      },
      "LastRun": "2000-01-29 10:51:34.776",
      "LastRunStatus": {
        "Id": 2060,
        "Name": "Completed"
      },
      "NextRun": null,
      "Params": [],
      "Schedule": "",
      "StartTime": null,
      "State": "Enabled",
      "Targets": [
        {
          "Data": "''",
          "Id": 123123,
          "JobId": 12345,
          "TargetType": {
            "Id": 1000,
            "Name": "DEVICE"
          }
        }
      ],
      "UpdatedBy": null,
      "Visible": true
    }
  ]}
```


Examples

```
- name: Get all jobs details.
  dellemc_ome_job_facts:
    hostname:  "192.168.0.1"
    username: "username"
    password:  "password"


- name: Get job details for id.
  dellemc_ome_job_facts:
    hostname:  "192.168.0.1"
    username: "username"
    password:  "password"
    job_id: 12345


- name: Get filtered job details.
  dellemc_ome_job_facts:
    hostname:  "192.168.0.1"
    username: "username"
    password:  "password"
    system_query_options:
      top: 2
      skip: 1
      filter: "JobType/Id eq 8"
```

Manage power state operations

Module: ome_power_state

Synopsis

This module performs the supported power state management operations.

Options

Table 69. ome_power_state

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
power_state	Yes	NA	• on • off • coldboot • warmboot • shutdown	Desired end power state
device_id	No	NA	NA	Targeted device id. **NOTE: I(device_id) is mutually exclusive with I(device_service_tag).**
device_service_tag	No	NA	NA	Targeted device service tag. **NOTE: I(device_service_tag) is mutually exclusive with I(device_id).**

Return Values

```
msg:
  type: str
  description: "Overall power state operation job status."
  returned: always
  sample: "Power State operation job submitted successfully."
job_status:
  type: dict
  description: "Power state operation job and progress details from the OME."
  returned: success
  sample: {
    "Builtin": false,
    "CreatedBy": "user",
    "Editable": true,
    "EndTime": null,
    "Id": 11111,
    "JobDescription": "DeviceAction_Task",
    "JobName": "DeviceAction_Task_PowerState",
    "JobStatus": {
      "Id": 1111,
      "Name": "New"
      },
    "JobType": {
      "Id": 1,
      "Internal": false,
      "Name": "DeviceAction_Task"
      },
    "LastRun": "2019-04-01 06:39:02.69",
    "LastRunStatus": {
      "Id": 1112,
      "Name": "Running"
      },
```

```
    "NextRun": null,
    "Params": [
      {
        "JobId": 11111,
        "Key": "powerState",
        "Value": "2"
      },
      {
        "JobId": 11111,
        "Key": "operationName",
        "Value": "POWER_CONTROL"
      },
    ],
    "Schedule": "",
    "StartTime": null,
    "State": "Enabled",
    "Targets": [
      {
        "Data": "",
        "Id": 11112,
        "JobId": 11111,
        "TargetType": {
          "Id": 0000,
          "Name": "DEVICE"
        },
      },
    ],
    "UpdatedBy": null,
    "Visible": true
  }
```

Examples

```
- name: Power state operation based on device id.
  ome_powerstate:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_id: 11111
    power_state: "off"


- name: Power state operation based on device service tag.
  ome_powerstate:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_service_tag: "KLBR111"
    power_state: "on"


- name: Power state operation based on list of device ids.
  ome_powerstate:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_id: "{{ item.device_id }}"
    power_state: "{{ item.state }}"
  with_items:
    - { "device_id": 11111, "state": "on" }
    - { "device_id": 22222, "state": "off" }


- name: Power state operation based on list of device service tags.
  ome_powerstate:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    device_service_tag: "{{ item.service_tag }}"
    power_state: "{{ item.state }}"
  with_items:
```

```
    - { "service_tag": "KLBR111", "state": "on" }
    - { "service_tag": "KLBR222", "state": "off" }
```

Manage users

The following tasks are responsible for managing user accounts:

- View user account details
- Configure user accounts

View user account details

Module: ome_user_info

Synopsis

This module retrieves the list and basic details of all user accounts or details of a specific user account.

Options

Table 70. ome_user_info

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
account_id	No	NA	NA	Unique ID of the account
system_query_options	No	NA	filter: Filter records for the supported values	Provides the option to filter the output for the supported values. I(filter) query format must be aligned with OData standards.

Return Values

```
msg:
  type: str
  description: Over all status of fetching user facts.
  returned: on error
  sample: "Failed to fetch the user facts"
user_info:
  type: dict
  description: Details of the users.
  returned: success
  sample: {
        "192.168.0.1": {
            "Id": "1814",
            "UserTypeId": 1,
            "DirectoryServiceId": 0,
            "Description": "user name description",
            "Name": "user_name",
            "Password": null,
            "UserName": "user_name",
            "RoleId": "10",
            "Locked": false,
            "IsBuiltin": true,
            "Enabled": true
        }
    }
```

Examples

```
- name: Retrieve basic details of all accounts.
  ome_user_info:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"


- name: Retrieve details of a specific account identified by its account ID.
  ome_user_info:
  hostname: "192.168.0.1"
  username: "username"
  password: "password"
  account_id: 1


- name: Get filtered user info based on user name
  ome_user_info:
  hostname: "192.168.0.1"
  username: "username"
  password: "password"
  system query options:
    filter: "Username eq 'test'"
```

Module: dellemc_ome_user_facts

Synopsis

This module retrieves the list and basic details of all user accounts or details of a specific user account.

NOTE: This module is deprecated and replaced with ome_user_info.

Options

Table 71. dellemc_ome_user_facts

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
account_id	No	NA	NA	Unique ID of the account

Return Values

```
msg:
  type: str
  description: Over all status of fetching user facts.
  returned: on error
  sample: "Failed to fetch the user facts"
ansible_facts:
  type: dict
  description: Details of the users.
  returned: success
  sample: {
        "192.168.0.1": {
            "Id": "1814",
            "UserTypeId": 1,
            "DirectoryServiceId": 0,
            "Description": "user name description",
            "Name": "user_name",
            "Password": null,
            "UserName": "user_name",
            "RoleId": "10",
            "Locked": false,
```

```
            "IsBuiltin": true,
            "Enabled": true
        }
    }
```

Examples

```
- name: Retrieve basic details of all accounts.
  dellemc_ome_user_facts:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"


- name: Retrieve details of a specific account identified by its account ID.
  dellemc_ome_user_facts:
  hostname: "192.168.0.1"
  username: "username"
  password: "password"
  account_id: 1
```

Configure user accounts

Module: ome_user

Synopsis

This module:

- creates a new user account.
- modifies or deletes an existing user account.

Options

Table 72. ome_user

Parameter	Required	Default	Choices	Comments
hostname	Yes	NA	NA	Target IP Address or hostname
username	Yes	NA	NA	Target username
password	Yes	NA	NA	Target user password
port	No	443	NA	Target device HTTPS port
state	No	present	• present • absent	• C(present) creates a user in case the I(UserName) provided inside I(attributes) does not exist . • C(present) modifies a user in case the I(UserName) provided inside I(attributes) exists . • C(absent) deletes an existing user.
user_id	No	NA	NA	• ID of the user to be deleted. • Either I (user_id) or I (name) is mandatory for C (absent) operation.
name	No	NA	NA	• Name of the user to be deleted • Either I (user_id) or I (name) is mandatory for C (absent) operation.
attributes	No	{ }	NA	Payload data for the user operations. It can take the following attributes for C(present): • UserTypeId • DirectoryServiceId • Description • Name

Table 72. ome_user(continued)

Parameter	Required	Default	Choices	Comments
				• Password • UserName • RoleId • Locked • Enabled **NOTE: OME will throw an error message if required parameter is not provided for the operation.** **NOTE: See OpenManage Enterprise API Reference Guide for more details.**

Return Values

```
msg:
  description: Overall status of the user operation.
  returned: always
  type: str
  sample: "Successfully created a User"
user_status:
  description: Details of the user operation when I(state) is C(present).
  returned: When I(state) is C(present).
  type: dict
  sample:
    {
        "Description": "Test user creation",
        "DirectoryServiceId": 0,
        "Enabled": true,
        "Id": "61546",
        "IsBuiltin": false,
        "Locked": false,
        "Name": "test",
        "ObjectGuid": null,
        "Oem": null,
        "Password": null,
        "PlainTextPassword": null,
        "RoleId": "10",
        "UserName": "test",
        "UserTypeId": 1
    }
```

Examples

```
- name: Create user with required parameters.
  ome_user:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    attributes:
      UserName: "user1"
      Password: "UserPassword"
      RoleId: "10",
      Enabled: True

- name: Create user with all parameters
  ome_user:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    attributes:
      UserName: "user2"
      Description: "user2 description"
      Password: "UserPassword"
      RoleId: "10"
      Enabled: True
      DirectoryServiceId: 0
```

```
      UserTypeId: 1
      Locked: False
      Name: "user2"

- name: Modify existing user
  ome_user:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    state: "present"
    attributes:
      UserName: "user3"
      RoleId: "10"
      Enabled: True
      Description: "Modify user Description"

- name: Delete existing user using id.
  ome_user:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    state: "absent"
    user_id: "1234"

- name: Delete existing user using name.
  ome_user:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    state: "absent"
    name:"name"
```

Manage identity pool

Identity pools are used in template-based deployment of servers. They facilitate the virtualization of network identities required for accessing systems using Ethernet, iSCSI, FCoE, or Fibre Channel (FC). This section describes how to manage the settings of an identity pool.

Manage Identity pool settings

Module: ome_identity_pool

Synopsis

This module allows to create, modify, or delete a single identity pool on OpenManage Enterprise.

Options

Table 73. ome_identity_pool

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
state	False	Present	Present	• C(present) modifies an existing identity pool. If the provided l (pool_name) does not exist, it creates an identity pool. • C(absent) deletes an existing identity pool.

Table 73. ome_identity_pool(continued)

Parameter	Required	Default	Choices	Comments
pool_name	True	NA	NA	This option is mandatory if l(command) is C(present) when creating, modifying, and deleting an identity pool.
new_pool_name	False	NA	NA	• After creating an identity pool, l(pool_name) can be changed to l(new_pool_name). • This option is ignored when creating an identity pool.
pool_description	False	NA	NA	Description of the identity pool.
ethernet_settings	False	NA	NA	Applicable for creating and modifying an identity pool using Ethernet settings. l(starting_mac_address) and l(identity_count) are required to create an identity pool. Suboptions- • starting_mac_address- Starting MAC address of the Ethernet setting. • identity_count- Number of MAC addresses.
fcoe_settings	False	NA	NA	Applicable for creating and modifying an identity pool using FCoE settings. l(starting_mac_address) and l(identity_count) are required to create an identity pool. Suboptions- • starting_mac_address- Starting MAC address of the FCoE setting. • identity_count- Number of MAC addresses.
iSCI_settings	False	NA	NA	Applicable for creating and modifying an identity pool using iSCSI settings. l(starting_mac_address), l(identity_count), l(iqn_prefix), l(ip_range) and l(subnet_mask) are required to create an identity pool. Suboptions: • starting_mac_address- Starting MAC address of the iSCSI setting. • identity_count- Number of MAC addresses

Table 73. ome_identity_pool(continued)

Parameter	Required	Default	Choices	Comments
				• initiator_config- Applicable for creating and modifying an identity pool using iSCSI Initiator settings • iqn_prefix- IQN prefix addresses • initiator_ip_pool_settings- Applicable for creating and modifying an identity pool using iSCSI Initiator IP pool settings • ip_range- Range of non-multicast IP addresses • subnet_mask- Subnet mask for I(ip_range) • gateway- IP address of gateway • primary_dns_server- IP address of the primary DNS server. • secondary_dns_server- IP address of the secondary DNS server
FC_settings	False	NA	NA	Applicable for creating and modifying an identity pool using fibre channel (FC) settings. I(starting_address) and I(identity_count) are required to create an identity pool. This option allows OpenManage Enterprise to generate a Worldwide port name (WWPN) and Worldwide node name (WWNN) address. The value 0x2001 is prefixed to the beginning address for the generation of a WWPN, and 0x2000 for a WWNN. suboptions: • starting_address- Starting MAC address of the FC setting • identity_count- Number of MAC addresses

Return Values

```
msg:
  type: str
  description: "Overall status of the identity pool operation"
  returned: always
  sample: "Successfully created an identity pool."
pool_status:
  type: dict
  description: Details of the user operation when I(state) is C(present).
  returned: success
  sample: {
            "Id":29,
```

```
            "IsSuccessful":True,
            "Issues":[]
        }
error_info:
  description: Details of the HTTP Error.
  returned: on HTTP error
  type: dict
  sample: {
  "error": {
         "@Message.ExtendedInfo": [{
         "Message": "Unable to process the request because an error occurred:
          Ethernet-MAC Range overlap found (in this Identity Pool or in a different one) .",
         "MessageArgs": [Ethernet-MAC Range overlap found (in this Identity Pool or in a
different one)"],
         "MessageId": "CGEN6001",
         "RelatedProperties": [],
         "Resolution": "Retry the operation. If the issue persists, contact your system
administrator.",
         "Severity": "Critical"
         }],

  "code": "Base.1.0.GeneralError",
  "message": "A general error has occurred. See ExtendedInfo for more information."
  }}
```

Examples

```
- name: Create an identity pool using ethernet, FCoE, iCSCI and FC settings.
  ome_identity_pool:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    state: present
    pool_name: "pool1"
    pool_description: "Identity pool with Ethernet, FCoE, ISCSI and FC settings"
    ethernet_settings:
        starting_mac_address: "50:50:50:50:50:00"
        identity_count: 60
    fcoe_settings:
        starting_mac_address: "70:70:70:70:70:00"
        identity_count: 75
    iscsi_settings:
        starting_mac_address: "60:60:60:60:60:00"
        identity_count: 30
        initiator_config:
            iqn_prefix: "iqn.myprefix."
        initiator_ip_pool_settings:
            ip_range: "10.33.0.1-10.33.0.255"
            subnet_mask: "255.255.255.0"
            gateway: "192.168.4.1"
            primary_dns_server : "10.8.8.8"
            secondary_dns_server : "8.8.8.8"
    fc_settings:
        starting_address: "30:30:30:30:30:00"
        identity_count: 45

- name: Create an identity pool using only ethernet settings.
  ome_identity_pool:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    pool_name: "pool2"
    pool_description: "create identity pool with ethernet"
    ethernet_settings:
        starting_mac_address: "aa-bb-cc-dd-ee-aa"
        identity_count: 80

- name: Modify an identity pool.
  ome_identity_pool:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    pool_name: "pool2"
```

```
    new_pool_name: "pool3"
    pool_description: "modifying identity pool with ethernet and fcoe settings"
    ethernet_settings:
        starting_mac_address: "90-90-90-90-90-90"
        identity_count: 61
    fcoe_settings:
        starting_mac_address: "aabb.ccdd.5050"
        identity_count: 77

-name: Modify an identity pool using  iSCSI and FC settings.
  ome_identity_pool:
    hostname: "{{hostname}}"
    username: "{{username}}"
    password: "{{password}}"
    pool_name: "pool_new"
    new_pool_name: "pool_new2"
    pool_description: "modifying identity pool with iscsi and fc settings"
    iscsi_settings:
      identity_count: 99
      initiator_config:
        iqn_prefix: "iqn1.myprefix2."
      initiator_ip_pool_settings:
        gateway: "192.168.4.5"
    fc_settings:
      starting_address: "10:10:10:10:10:10"
      identity_count: 98

- name: Delete an identity pool.
  ome_identity_pool:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    state: "absent"
    pool_name: "pool2"
```

Manage network settings

This section describes how to manage the following network settings on OpenManage enterprise-

- Configuring a network proxy
- Managing the network configuration

Update proxy configuration

Module: ome_application_network_proxy

Synopsis

This module allows the configuration of a network proxy.

Options

Table 74. ome_application_network_proxy

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
enable_proxy	True	NA	NA	• Enables or disables the HTTP proxy configuration.

Table 74. ome_application_network_proxy(continued)

Parameter	Required	Default	Choices	Comments
				• If I(enable proxy) is false, the HTTP proxy configuration is set to its default value.
ip_address	False	NA	NA	• Proxy server address • This option is mandatory when I(enable_proxy) is true.
proxy_port	False	NA	NA	• Port number of the proxy server. • This option is mandatory when I(enable_proxy) is true.
enable_authentication	False	NA	NA	• Enables or disables proxy authentication. • If I(enable_authentication) is true, a username and password must be provided. • If I(enable_authentication) is false, the proxy username and password are set to its default values.
proxy_username	False	NA	NA	Username of the proxy server. This option is mandatory when I(enable_authentication) is true.
proxy_password	False	NA	NA	Password of the proxy server. This option is mandatory when I(enable_authentication) is true.

Return Values

```
r'''
---
msg:
  type: str
  description: Overall status of the network proxy configuration change.
  returned: always
  sample: "Successfully updated network proxy configuration."
proxy_setting:
  type: dict
  description: Updated network proxy configuration.
  returned: success
  sample: {
        "EnableAuthentication": true,
        "EnableProxy": true,
        "IpAddress": "192.168.0.2",
```

```
        "Password": null,
        "PortNumber": 444,
        "Username": "root"
        }
error_info:
  description: Details of the HTTP error.
  returned: on HTTP error
  type: dict
  sample: {
        "error": {
            "@Message.ExtendedInfo": [
                {
                   "Message": "Unable to complete the request because the input value
                    for  PortNumber  is missing or an invalid value is entered.",
                    "MessageArgs": [
                        "PortNumber"
                    ],
                    "MessageId": "CGEN6002",
                    "RelatedProperties": [],
                    "Resolution": "Enter a valid value and retry the operation.",
                    "Severity": "Critical"
                }
            ],
            "code": "Base.1.0.GeneralError",
            "message": "A general error has occurred. See ExtendedInfo for more information."
        }
    }
'''
```

Examples

```
- name: Update proxy configuration and enable authentication.
  ome_application_network_proxy:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    enable_proxy: true
    ip_address: "192.168.0.2"
    proxy_port: 444
    enable_authentication: true
    proxy_username: "proxy_username"
    proxy_password: "proxy_password"

- name: Reset proxy authentication.
  ome_application_network_proxy:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    enable_proxy: true
    ip_address: "192.168.0.2"
    proxy_port: 444
    enable_authentication: false

- name: Reset proxy configuration.
  ome_application_network_proxy:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    enable_proxy: false
```

Update network configuration

Module: ome_application_network_address

Synopsis

- This module allows the configuration of a DNS and an IPV4 or IPV6 network on OpenManage Enterprise. It is only applicable on versions 3.3 and above of OpenManage Enterprise and OpenManage Enterprise Modular.

  NOTE:

- **The configuration changes can only be applied to one interface at a time.**
- **Once the configuration changes are applied, the system management consoles might be unreachable for 2 minutes, based on the changes made.**

Options

Table 75. ome_application_network_address

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
enable_nic	False	True	NA	Enable or disable Network Interface Card (NIC) configuration.
interface_name	False	NA	NA	• If there are multiple interfaces, network configuration changes can be applied to a single interface using the interface name of the NIC. • If this option is not specified, the primary interface is chosen by default.
ipv4_configuration	False	NA	ipv4 options	IPv4 network configuration **NOTE: Ensure that you have an alternate interface to access OpenManage Enterprise as these options can change the current IPv4 address for I(hostname).** Suboptions: • enable: Enable or disable access to the network using IPv4. • enable_dhcp: Enable or disable the automatic request to get an IPv4 address from the IPv4 Dynamic Host Configuration Protocol (DHCP) server. If this option is true, then OpenManage Enterprise retrieves the IP configuration—IPv4 address, subnet mask, and gateway from a DHCP server on the existing network. • static_ip_address: Static IPv4 address. This option is applicable when I(enable_dhcp) is false. • static_subnet_mask: Static IPv4 subnet mask address. This option is applicable when I(enable_dhcp) is false. • static_gateway: Static IPv4 gateway address. This option is applicable when I(enable_dhcp) is false. • use_dhcp_for_dns_server_names: This option allows to automatically request and obtain a DNS server IPv4 address from the DHCP server. This option is applicable when I(enable_dhcp) is true. • static_preferred_dns_server: Static IPv4 DNS preferred server. This option is applicable when I(use_dhcp_for_dns_server_names) is false. • static_alternate_dns_server: Static IPv4 DNS alternate server. This option is applicable when I(use_dhcp_for_dns_server_names) is false.
ipv6_configuration	False	NA	ipv6 options	IPv6 network configuration **NOTE: Ensure that you have an alternate interface to access OpenManage Enterprise as**

Table 75. ome_application_network_address(continued)

Parameter	Required	Default	Choices	Comments
				these options can change the current IPv6 address for I(hostname). Suboptions: • enable: Enable or disable access to the network using IPv6. • enable_auto_configuration: Enable or disable the automatic request to get an IPv6 address from the IPv6 DHCP server or router advertisements(RA). If I(enable_auto_configuration) is true, then OpenManage Enterprise retrieves the IP configuration —IPv6 address, prefix, and gateway, from a DHCPv6 server on the existing network. • static_ip_address: Static IPv6 address. This is applicable when I(enable_auto_configuration) is false. • static_prefix_le.ngth: Static IPv6 prefix length. This is applicable when I(enable_auto_configuration) is false. • static_gateway: Static IPv6 gateway address. This is applicable when I(enable_auto_configuration) is false. • use_dhcp_for_dns_server_names: This option allows to automatically request and obtain a DNS server IPv6 address from the DHCP server. This is applicable when I(enable_auto_configuration) is true. • static_preferred_dns_server: Static IPv6 DNS preferred server. This is applicable when I(use_dhcp_for_dns_server_names) is false. • static_alternate_dns_server: Static IPv6 DNS alternate server. This is applicable when I(use_dhcp_for_dns_server_names) is false.
management_vlan				• vLAN configuration • These settings are only applicable for OpenManage Enterprise Modular. Suboptions: • enable_vlan: ○ Enable or disable vLAN for management. ○ The vLAN configuration cannot be updated if the I(register_with_dns) field under I(dns_configuration) is true. ○ I(WARNING) Ensure that the network cable is plugged to the correct port after the vLAN configuration changes have been made. If not, the configuration change may not be effective. • vlan_id: ○ vLAN ID ○ This option is applicable when I(enable_vlan) is true.
dns_configuration	False	NA	NA	DNS settings Suboptions: • register_with_dns: Register or unregister I(dns_name) on the DNS Server. This option cannot be updated if vLAN configuration changes.

Table 75. ome_application_network_address(continued)

Parameter	Required	Default	Choices	Comments
				• use_dhcp_for_dns_domain_name: Get the I(dns_domain_name) using a DHCP server. • dns_name: DNS name for I(hostname). This is applicable when I(register_with_dns) is true. • dns_domain_name: Static DNS domain name. This is applicable when I(use_dhcp_for_dns_domain_name) is false.
reboot_delay	False	NA	NA	The time in seconds, after which settings are applied. This option is not mandatory.

Retuen values

```
msg:
  type: str
  description: Overall status of the network address configuration change.
  returned: always
  sample: Successfully updated network address configuration
network_configuration:
  type: dict
  description: Updated application network address configuration.
  returned: on success
  sample: {
    "Delay": 0,
    "DnsConfiguration": {
        "DnsDomainName": "",
        "DnsName": "MX-SVCTAG",
        "RegisterWithDNS": false,
        "UseDHCPForDNSDomainName": true
    },
    "EnableNIC": true,
    "InterfaceName": "eth0",
    "PrimaryInterface": true,
    "Ipv4Configuration": {
        "Enable": true,
        "EnableDHCP": false,
        "StaticAlternateDNSServer": "",
        "StaticGateway": "192.168.0.2",
        "StaticIPAddress": "192.168.0.3",
        "StaticPreferredDNSServer": "192.168.0.4",
        "StaticSubnetMask": "255.255.254.0",
        "UseDHCPForDNSServerNames": false
    },
    "Ipv6Configuration": {
        "Enable": true,
        "EnableAutoConfiguration": true,
        "StaticAlternateDNSServer": "",
        "StaticGateway": "",
        "StaticIPAddress": "",
        "StaticPreferredDNSServer": "",
        "StaticPrefixLength": 0,
        "UseDHCPForDNSServerNames": true
    },
    "ManagementVLAN": {
        "EnableVLAN": false,
        "Id": 1
    }
  }
error_info:
  description: Details of the HTTP error.
  returned: on HTTP error
  type: dict
  sample: {
    "error": {
    "@Message.ExtendedInfo": [
        {
            "Message": "Unable to update the address configuration because a dependent field
```

```
is missing for  Use DHCP
            for DNS Domain Name, Enable DHCP for ipv4 or Enable Autoconfig for ipv6 settings
for valid configuration .",
            "MessageArgs": [
                "Use DHCP for DNS Domain Name, Enable DHCP for ipv4 or Enable Autoconfig for
ipv6 settings for valid
                configuration"
            ],
            "MessageId": "CAPP1304",
            "RelatedProperties": [],
            "Resolution": "Make sure that all dependent fields contain valid content and
retry the operation.",
            "Severity": "Critical"
        }
    ],
    "code": "Base.1.0.GeneralError",
    "message": "A general error has occurred. See ExtendedInfo for more information."
    }
  }
```

Examples

```
- name: IPv4 network configuration for primary interface
  ome_application_network_address:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    enable_nic: true
    ipv4_configuration:
      enable: true
      enable_dhcp: false
      static_ip_address: 192.168.0.2
      static_subnet_mask: 255.255.254.0
      static_gateway: 192.168.0.3
      use_dhcp_for_dns_server_names: false
      static_preferred_dns_server: 192.168.0.4
      static_alternate_dns_server: 192.168.0.5
    reboot_delay: 5

- name: IPv6 network configuration for primary interface
  ome_application_network_address:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    ipv6_configuration:
      enable: true
      enable_auto_configuration: true
      static_ip_address: 2626:f2f2:f081:9:1c1c:f1f1:4747:10
      static_prefix_length: 10
      static_gateway: 2626:f2f2:f081:9:1c1c:f1f1:4747:1
      use_dhcp_for_dns_server_names: true
      static_preferred_dns_server: 2626:f2f2:f081:9:1c1c:f1f1:4747:2
      static_alternate_dns_server: 2626:f2f2:f081:9:1c1c:f1f1:4747:3
    reboot_delay: 10

- name: Management vLAN configuration for primary interface
  ome_application_network_address:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    management_vlan:
      enable_vlan: true
      vlan_id: 3344
    dns_configuration:
      register_with_dns: false
    reboot_delay: 1

- name: DNS settings
  ome_application_network_address:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    ipv4_configuration:
```

```
      enable: true
      use_dhcp_for_dns_server_names: false
      static_preferred_dns_server: 192.168.0.4
      static_alternate_dns_server: 192.168.0.5
    dns_configuration:
      register_with_dns: true
      use_dhcp_for_dns_domain_name: false
      dns_name: "MX-SVCTAG"
      dns_domain_name: "dnslocaldomain"
    reboot_delay: 1

- name: Disbale nic interface eth1
  ome_application_network_address:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    enable_nic: false
    interface_name: eth1

- name: Complete network settings for interface eth1
  ome_application_network_address:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    enable_nic: true
    interface_name: eth1
    ipv4_configuration:
      enable: true
      enable_dhcp: false
      static_ip_address: 192.168.0.2
      static_subnet_mask: 255.255.254.0
      static_gateway: 192.168.0.3
      use_dhcp_for_dns_server_names: false
      static_preferred_dns_server: 192.168.0.4
      static_alternate_dns_server: 192.168.0.5
    ipv6_configuration:
      enable: true
      enable_auto_configuration: true
      static_ip_address: 2607:f2b1:f081:9:1c8c:f1c7:47e:f120
      static_prefix_length: 10
      static_gateway: ffff::2607:f2b1:f081:9
      use_dhcp_for_dns_server_names: true
      static_preferred_dns_server: 2626:f2f2:f081:9:1c1c:f1f1:4747:1
      static_alternate_dns_server: 2626:f2f2:f081:9:1c1c:f1f1:4747:2
    dns_configuration:
      register_with_dns: false
      use_dhcp_for_dns_domain_name: true
      dns_name: "MX-SVCTAG"
      dns_domain_name: "dnslocaldomain"
    reboot_delay: 1
```

Update web server configuration

Module: ome_application_network_webserver

Synopsis

This module allows the configuration of a network web server.

Options

Table 76. ome_application_network_webserver

Parameter	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port

Table 76. ome_application_network_webserver(continued)

Parameter	Required	Default	Choices	Comments
webserver_port	False	NA	NA	Port number used by OpenManage Enterprise to establish a secure server connection. **WARNING: A change in port number results in a loss of connectivity in the current session for a minute or more.**
webserver_timeout	False	NA	NA	• The duration, in minutes, after which a web user interface session is automatically disconnected. • If a change is made to the session timeout, it will only take effect after the next login.

Return Values

```
msg:
  type: str
  description: Overall status of the network web server configuration change.
  returned: always
  sample: "Successfully updated network web server configuration."
webserver_configuration:
  type: dict
  description: Updated application network web server configuration.
  returned: success
  sample: {
        "TimeOut": 20,
        "PortNumber": 443,
        "EnableWebServer": true
        }
error_info:
  description: Details of the HTTP error.
  returned: on HTTP error
  type: dict
  sample: {
    "error": {
        "@Message.ExtendedInfo": [
            {
               "Message": "Unable to complete the request because the input value
                for  PortNumber  is missing or an invalid value is entered.",
                "MessageArgs": [
                    "PortNumber"
                ],
                "MessageId": "CGEN6002",
                "RelatedProperties": [],
                "Resolution": "Enter a valid value and retry the operation.",
                "Severity": "Critical"
            }
        ],
        "code": "Base.1.0.GeneralError",
        "message": "A general error has occurred. See ExtendedInfo for more information."
    }
    }
```

Examples

```
- name: Update web server port and session time out configuration.
  ome_application_network_webserver:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    webserver_port: 443
    webserver_timeout: 10

- name: Update session time out.
  ome_application_network_webserver:
    hostname: "192.168.0.1"
    username: "username"
```

```
    password: "password"
    webserver_timeout: 30

- name: Update web server port.
  ome_application_network_webserver:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    webserver_port: 8443
```

Update time configuration

Module: ome_application_network_time

Synopsis

This module allows the configuration of network time.

Options

Table 77. ome_application_network_time

Parameters	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP Address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
enable_ntp	True	NA	NA	• This option enables or disables Network Time Protocol(NTP). • If I(enable_ntp) is false, then the NTP addresses reset to their default values.
system_time	False	NA	NA	• Time in the current system. • This option is only applicable when I(enable_ntp) is false. • This option must be provided in following format - 'yyyy-mm-dd hh:mm:ss'.
time_zone	False	NA	NA	• The valid time zone ID to be used. • This option is applicable for both system time and NTP time synchronization.
primary_ntp_address	False	NA	NA	• The primary NTP address • This option is applicable when I(enable_ntp) is true.
secondary_ntp_address1	False	NA	NA	• The first secondary NTP address • This option is applicable when I(enable_ntp) is true.
secondary_ntp_address2	False	NA	NA	• The second secondary NTP address • This option is applicable when I(enable_ntp) is true.

Return Values

```
msg:
  type: str
  description: Overall status of the network time configuration change.
  returned: always
  sample: "Successfully configured network time."
proxy_configuration:
  type: dict
  description: Updated application network time configuration.
  returned: success
  sample: {
        "EnableNTP": false,
        "JobId": null,
        "PrimaryNTPAddress": null,
        "SecondaryNTPAddress1": null,
        "SecondaryNTPAddress2": null,
        "SystemTime": null,
        "TimeSource": "Local Clock",
        "TimeZone": "TZ_ID_1",
        "TimeZoneIdLinux": null,
        "TimeZoneIdWindows": null,
        "UtcTime": null
        }
error_info:
  description: Details of the HTTP error.
  returned: on HTTP error
  type: dict
  sample: {
        "error": {
            "@Message.ExtendedInfo": [
                {
                   "Message":  "Unable to complete the request because the input value
                    for  SystemTime  is missing or an invalid value is entered.",
                    "MessageArgs": [
                        "SystemTime"
                    ],
                    "MessageId": "CGEN6002",
                    "RelatedProperties": [],
                    "Resolution": "Enter a valid value and retry the operation.",
                    "Severity": "Critical"
                }
            ],
            "code": "Base.1.0.GeneralError",
            "message": "A general error has occurred. See ExtendedInfo for more information."
        }
    }
```

Examples

```
- name: Configure system time.
  ome_application_network_time:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    enable_ntp: false
    system_time: "2020-03-31 21:35:18"
    time_zone: "TZ_ID_11"

- name: Configure NTP server for time synchronization.
  ome_application_network_time:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    enable_ntp: true
    time_zone: "TZ_ID_66"
    primary_ntp_address: "10.136.112.220"
    secondary_ntp_address1: "10.136.112.221"
    secondary_ntp_address2: "10.136.112.222"
```

Generate a certificate signing request

Module: ome_application_certificate

Synopsis

This module allows the generation of a new certificate signing request (CSR).

Options

Table 78. ome_application_certifcate

Parameters	Required	Default	Choices	Comments
hostname	True	NA	NA	Target IP address or hostname
username	True	NA	NA	Target username
password	True	NA	NA	Target user password
port	False	443	NA	Target HTTPS port
command	False	NA	NA	C(generate_csr) allows the generation of a CSR. The following options are applicable for this command- l(distinguished_name), l(department_name), l(business_name), l(locality), l(country_state), l(country), and l(email).
distinguished_name	False	NA	NA	Name of the certificate issuer
department_name	False	NA	NA	Name of the department that issued the certificate.
business_name	False	NA	NA	Name of the business that issued the certificate.
locality	False	NA	NA	Local address of the issuer of the certificate.
country_state	False	NA	NA	State in which the issuer resides.
country	False	NA	NA	Country in which the issuer resides.
email	False	NA	NA	Email associated with the issuer.

Return Values

```
msg:
  type: str
  description: Overall status of the certificate signing request.
  returned: always
  sample: "Successfully generated certificate signing request."
csr_status:
  type: dict
  description: details of the generated certificate.
  returned: on success
  sample:
    {"CertificateData": "-----BEGIN CERTIFICATE REQUEST-----MIIFFjCCAv4
      CAQAwgZ8xCzAJBgNVBAYTAlVTMQ4wDAYDVQQIDAVUZXhhczETMBEGA1UEBwwKUm91
      bmQgUm9jazESMBAGA1UECgwJRGVsbCBJbmMuMRwwGgYDVQQLDBNSZW1vdGUgQWNjZ
      XNzIEdyb3VwMRwwGgYDVQQDDBNob3N0bmN8Mq6gnvxVmucGbUGmRyrXizGcpTCj5p
      Uv7cALZWqoHblPirAgjmJ8PipTkV93bWr0i34tUJgEb9g/aHOJ6nV4zAyc3zhfqjt
      p4PHAaBqIXPe0tbiqj7WZwE6GPPaW5seRGvzAIPuwn4kod4tXB0DQt4kSIh9TyCSG
      mh5mBAMdOD7Wd0ddXxmeoFJPa/sYQJZarJ/TPr2JAJAAKdxz2XLPokLHmjG02Xje3
      RWQDNm+ngR/UTdXs/51kLrSwlU2LXFaQeBdcrwMdiZCOJPsfl6kf9fxobvqScdRYl
      gjJO7S5UcjJkBkeNURc080N9DCknV4bO1lo9BOA4aEhjo9gFFIUNk8iscMJJqyvHh
      BhzRSWH6fx7u9NGhnlDEOoyJnjceuI7zDS3CT/7pByuCoDc+dK2DezansSJHV4xYC
      eBmO14MpukxfoMxbSXZUdfkQgZZ1LmJGTYH0omGIm0KC+7g2ITZf1FrR8HcjEbKgV
      ZopugdSPXGp4P7eLRA/xIIp3GbrRXbSAumAO5fNefVsIzxZ34fw5O+msj/IH/IAJy
      EP3fq8iflVyV3hQjlUPSq/ZGYy7vPvwZHGhPPDXjvNVgyyD7zKSOkKZIyOL2Xvpom
```

```
      1cuJ1veYniuZkVvENkRNxzTmKlzUlYk4326Xauw==-----END CERTIFICATE REQUEST-----"
    }
error_info:
  description: Details of the HTTP error.
 returned: on HTTP error
  type: dict
  sample:
    {
        "error": {
            "code": "Base.1.0.GeneralError",
            "message": "A general error has occurred. See ExtendedInfo for more information.",
            "@Message.ExtendedInfo": [
                {
                    "MessageId": "CSEC9002",
                    "RelatedProperties": [],
                    "Message": "Unable to upload the certificate because the certificate file
provided is invalid.",
                    "MessageArgs": [],
                    "Severity": "Critical",
                    "Resolution": "Make sure the CA certificate and private key are correct
and retry the operation."
                }
            ]
        }
    }
```

Examples

```
- name: Generate a certificate signing request.
  ome_application_certificate:
    hostname: "192.168.0.1"
    username: "username"
    password: "password"
    command: "generate_csr"
    distinguished_name: "hostname.com"
    department_name: "Remote Access Group"
    business_name: "Dell Inc."
    locality: "Round Rock"
    country_state: "Texas"
    country: "US"
    email: "support@dell.com"
```

5

Modules for Redfish APIs

How OpenManage Ansible Modules work with Redfish APIs

The Redfish Scalable Platforms Management API is a standard defined by the Distributed Management Task Force (DMTF). Redfish is a next-generation systems management interface standard which enables scalable, secure, and open server management. It is an interface that uses RESTful interface semantics to access data that is defined in model format to perform out-of-band systems management.

OpenManage Anisble modules use standard redfish URIs supported by iDRAC, to perform firmware updates or manage storage volume configurations on PowerEgde servers.

Firmware update using standard Redfish URI

Module: redfish_firmware

Synopsis

- This module allows the firmware update of only one component at a time. If the module is run for more than one component, an error message is returned.
- Depending on the component, the firmware update is applied after an automatic or manual reboot.

Options

Table 79. redfish_firmware

Parameter	Required	Default	Choices	Comments
baseuri	True	NA	NA	IP Address of the target out-of-band controller. For example- <ipaddress>:<port>
username	True	NA	NA	Username of the target out-of-band controller .
password	True	NA	NA	Password of the target out-of-band controller .
image_uri	True	NA	NA	• Firmware image location URI or local path. • For example- U(http://<web_address>/components.exe) or /home/firmware_repo/component.exe
transfer_protocol	False	HTTP	HTTP, HTTPS, FTP, NSF, CIFS, FTP, OEM, SCP, SFTP, or TFTP	• Protocol used to transfer the firmware image file. • Applicable for URI-based update. **NOTE: Dell PowerEdge servers**

Table 79. redfish_firmware(continued)

Parameter	Required	Default	Choices	Comments
				support transfer protocols only through HTTP-based shares.

Return values

```
msg:
  description: Overall status of the firmware update task.
  returned: always
  type: str
  sample: Successfully submitted the firmware update task.
task:
  description: Returns ID and URI of the created task.
  returned: success
  type: dict
  sample: {
        "id": "JID_XXXXXXXXXXX",
        "uri": "/redfish/v1/TaskService/Tasks/JID_XXXXXXXXXXX"
    }
error_info:
  type: dict
  description: Details of a http error.
  returned: on http error
  sample:  {
        "error": {
            "@Message.ExtendedInfo": [
                {
                    "Message": "Unable to complete the operation because the JSON data format
entered is invalid.",
                    "Resolution": "Do the following and the retry the operation:
                        1) Enter the correct JSON data format and retry the operation.
                        2) Make sure that no syntax error is present in JSON data format.
                        3) Make sure that a duplicate key is not present in JSON data
format.",
                    "Severity": "Critical"
                },
                {
                    "Message": "The request body submitted was malformed JSON and
                        could not be parsed by the receiving service.",
                    "Resolution": "Ensure that the request body is valid JSON and resubmit
the request.",
                    "Severity": "Critical"
                }
            ],
            "code": "Base.1.2.GeneralError",
            "message": "A general error has occurred. See ExtendedInfo for more information."
        }
    }
```

Examples

```
- name: Update the firmware from a single executable file available in a HTTP protocol
  redfish_firmware:
    baseuri: "192.168.0.1"
    username: "user_name"
    password: "user_password"
    image_uri: "http://192.168.0.2/firmware_repo/component.exe"
    transfer_protocol: "HTTP"

- name: Update the firmware from a single executable file available in a local path
  redfish_firmware:
    baseuri: "192.168.0.1"
    username: "user_name"
```

```
    password: "user_password"
    image_uri: "/home/firmware_repo/component.exe"
```

Manage storage volume configuration

Module: redfish_storage_volume

Synopsis

This module allows to create, modify, initialize, or delete a single storage volume.

Options

Table 80. redfish_storage_volume

Parameter	Required	Default	Choices	Comments
baseuri	True	NA	NA	IP address of the target out-of-band controller. For example- <ipaddress>:<port>
username	True	NA	NA	Username of the target out-of-band controller.
password	True	NA	NA	Password of the target out-of-band controller.
controller_id	False	NA	NA	• Fully Qualified Device Descriptor (FQDD) of the storage controller. • For example- RAID.Slot.1-1. • This option is mandatory when I(state) is C(present) when creating a volume.
volume_id	False	NA	NA	• FQDD of existing volume. • For example- Disk.Virtual.4:RAID.Slot.1-1. • This option is mandatory in the following scenarios-: ○ I(state) is C(present), when updating a volume. ○ I(state) is C(absent), when deleting a volume. ○ I(command) is C(initialize), when initializing a volume.
state	False	NA	Present, or absent.	• C(present) creates a storage volume for a specified I (controller_id), or modifies the storage volume for a specified I (volume_id).

Table 80. redfish_storage_volume(continued)

Parameter	Required	Default	Choices	Comments
				NOTE: Modification of an existing volume depends on drive and controller capabilities. • C(absent) deletes the volume for a specified I(volume_id).
command	False	NA	Initialize	C(initialize) initializes an existing storage volume for a specified I (volume_id).
volume_type	False	NA	NonRedundant, Mirrored,StripedWithParity, SpannedMirrors, or SpannedStripesWithParity.	One of the following volume types must be selected to create a volume- • C(Mirrored) The volume is a mirrored device. • C(NonRedundant) The volume is a non-redundant storage device. • C(SpannedMirrors) The volume is a spanned set of mirrored devices. • C(SpannedStripesWith Parity) The volume is a spanned set of devices which uses parity to retain redundant information. • C(StripedWithParity) The volume is a device which uses parity to retain redundant information.
name	False	NA	NA	• Name of the volume to be created. • Only applicable when I(state) is C(present).
drives	False	NA	NA	• FQDD of the Physical disks. • For example-Disk.Bay.0:Enclosure.Internal.0-1:RAID.Slot.1-1. • Only applicable when I(state) is C(present) when creating a new volume.
block_size_bytes	False	NA	NA	Block size in bytes.Only applicable when I(state) is C(present).
capacity_bytes	False	NA	NA	• Virtual disk size in bytes.

Table 80. redfish_storage_volume(continued)

Parameter	Required	Default	Choices	Comments
				• Only applicable when I(state) is C(present).
optimum_io_size_bytes	False	NA	NA	• Stripe size value must be in multiples of 64 * 1024. • Only applicable when I(state) is C(present).
encryption_types	False	NA	NativeDriveEncryption, ControllerAssisted, or SoftwareAssisted.	The following encryption types can be selected. • C(ControllerAssisted) The volume is encrypted by the storage controller entity. • C(NativeDriveEncryption) The volume utilizes the native drive encryption capabilities of the drive hardware. • C(SoftwareAssisted) The volume is encrypted by the software running on the system or the operating system. • Only applicable when I(state) is C(present).
encrypted	False	NA	NA	• Indicates whether volume is currently utilizing encryption or not. • Only applicable when I(state) is C(present).
oem	False	NA	NA	• Includes OEM extended payloads. • Only applicable when I(state) is I(present).
initialize_type	False	NA	Fast, or slow.	• Initialization type of existing volume. • Only applicable when I(command) is C(initialize).

Return Values

```
msg:
  description: Overall status of the storage configuration operation.
  returned: always
  type: str
  sample: "Successfully submitted create volume task."
task:
  type: dict
  description: Returns ID and URI of the created task.
  returned: success
  sample: {
```

```
    "id": "JID_XXXXXXXXXXXX",
    "uri": "/redfish/v1/TaskService/Tasks/JID_XXXXXXXXXXXX"
  }
error_info:
  type: dict
  description: Details of a http error.
  returned: on http error
  sample:  {
    "error": {
        "@Message.ExtendedInfo": [
            {
                "Message": "Unable to perform configuration operations because a
                          configuration job for the device already exists.",
                "MessageArgs": [],
                "MessageArgs@odata.count": 0,
                "MessageId": "IDRAC.1.6.STOR023",
                "RelatedProperties": [],
                "RelatedProperties@odata.count": 0,
                "Resolution": "Wait for the current job for the device to complete
                    or cancel the current job before attempting more configuration
                     operations on the device.",
                "Severity": "Informational"
            }
        ],
        "code": "Base.1.2.GeneralError",
        "message": "A general error has occurred. See ExtendedInfo for more information"
    }
  }
```

Examples

```
- name: Create a volume with supported options.
  redfish_storage_volume:
    baseuri: "192.168.0.1"
    username: "username"
    password: "password"
    state: "present"
    volume_type: "Mirrored"
    name: "VD0"
    controller_id: "RAID.Slot.1-1"
    drives:
      - Disk.Bay.5:Enclosure.Internal.0-1:RAID.Slot.1-1
      - Disk.Bay.6:Enclosure.Internal.0-1:RAID.Slot.1-1
    block_size_bytes: 512
    capacity_bytes: 299439751168
    optimum_io_size_bytes: 65536
    encryption_types: NativeDriveEncryption
    encrypted: true

- name: Create a volume with minimum options.
  redfish_storage_volume:
    baseuri: "192.168.0.1"
    username: "username"
    password: "password"
    state: "present"
    controller_id: "RAID.Slot.1-1"
    volume_type: "NonRedundant"
    drives:
       - Disk.Bay.1:Enclosure.Internal.0-1:RAID.Slot.1-1

- name: Modify a volume's encryption type settings.
  redfish_storage_volume:
    baseuri: "192.168.0.1"
    username: "username"
    password: "password"
    state: "present"
    volume_id: "Disk.Virtual.5:RAID.Slot.1-1"
    encryption_types: "ControllerAssisted"
    encrypted: true

- name: Delete an existing volume.
  redfish_storage_volume:
    baseuri: "192.168.0.1"
```

```
    username: "username"
    password: "password"
    state: "absent"
    volume_id: "Disk.Virtual.5:RAID.Slot.1-1"

- name: Initialize an existing volume.
  redfish_storage_volume:
    baseuri: "192.168.0.1"
    username: "username"
    password: "password"
    command: "initialize"
    volume_id: "Disk.Virtual.6:RAID.Slot.1-1"
    initialize_type: "Slow"
```

6

Accessing documents from the Dell EMC support site

You can access the required documents using the following links:

- For Dell EMC Enterprise Systems Management documents — www.dell.com/esmmanuals
- For Dell EMC OpenManage documents — www.dell.com/openmanagemanuals
- For Dell EMC Remote Enterprise Systems Management documents — www.dell.com/esmmanuals
- For iDRAC and Dell Lifecycle Controller documents — www.dell.com/idracmanuals
- For Dell EMC OpenManage Connections Enterprise Systems Management documents — www.dell.com/esmmanuals
- For Dell EMC Serviceability Tools documents — www.dell.com/serviceabilitytools
- 1. Go to www.dell.com/support.
  2. Click **Browse all products**.
  3. From **All products** page, click **Software**, and then click the required link from the following:
     - **Analytics**
     - **Client Systems Management**
     - **Enterprise Applications**
     - **Enterprise Systems Management**
     - **Public Sector Solutions**
     - **Utilities**
     - **Mainframe**
     - **Serviceability Tools**
     - **Virtualization Solutions**
     - **Operating Systems**
     - **Support**
  4. To view a document, click the required product and then click the required version.
- Using search engines:
  - Type the name and version of the document in the search box.