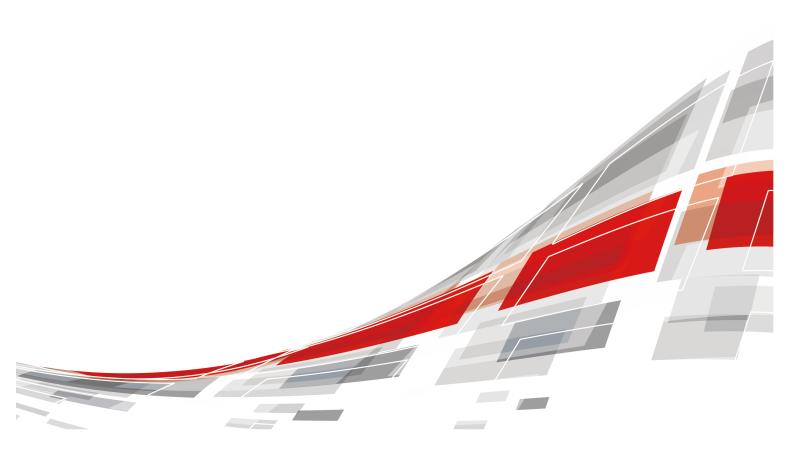
xFusion iBMC Ansible Module 2.0.14

User Guide

Issue 01

Date 2023-12-30



Copyright © xFusion Digital Technologies Co., Ltd. 2023. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of xFusion Digital Technologies Co., Ltd.

Trademarks and Permissions

CFUSION and other xFusion trademarks are trademarks of xFusion Digital Technologies Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

In this document, "xFusion" is used to refer to "xFusion Digital Technologies Co., Ltd." for concise description and easy understanding, which does not mean that "xFusion" may have any other meaning. Any "xFusion" mentioned or described hereof may not be understood as any meaning other than "xFusion Digital Technologies Co., Ltd.", and xFusion Digital Technologies Co., Ltd. shall not bear any liability resulting from the use of "xFusion".

The purchased products, services and features are stipulated by the contract made between xFusion and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

xFusion Digital Technologies Co., Ltd.

Address: 9th Floor, Building 1, Zensun Boya Square, Longzihu Wisdom Island

Zhengdong New District 450046 Zhengzhou, Henan Province People's Republic of China

Website: https://www.xfusion.com

About This Document

Purpose

This document describes how to install and uninstall the Ansible plug-in and how to use the plug-in to implement the information query, health status query, configuration, deployment, and firmware upgrade functions on server.

Intended Audience

This document is intended for:

- Technical support engineers
- System maintenance engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
▲ DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
À WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
⚠ CAUTION	Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal
	injury.
Ш NОТЕ	Calls attention to important information, best practices, and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Change History

Issue	Date	Description
01	2023-12-30	This issue is the first official release.

Contents

About This Document	ii
1 Introduction	1
2 Installing and Uninstalling the Ansible Plug-in	4
2.1 Installing the Ansible Plug-in	
2.2 Uninstalling the Ansible Plug-in	
3 Configuring the Ansible Plug-in	7
3.1 Configuring /etc/ansible/hosts	
3.2 Configuring the /group_vars/myhosts File	8
3.3 Configuring the SSL Certificate Authentication and TLS 1.2 Communication Mode	
4 Using the Ansible Plug-in	13
4.1 Querying Basic Server Information	16
4.2 Configuring the Boot Device	17
4.2.1 Querying Boot Option Information	17
4.2.2 Setting the Boot Device	18
4.3 Managing the Power Supply	18
4.3.1 Querying PSU Status (JSON File Generated)	19
4.3.2 Setting the Power Supply Status.	19
4.4 Managing iBMC Users	20
4.4.1 Querying iBMC User Information (JSON File Generated)	20
4.4.2 Creating an iBMC User	20
4.4.3 Modify iBMC User Information	
4.4.4 Deleting an iBMC User	23
4.5 Configuring iBMC Network Information	24
4.5.1 Querying iBMC Network Configuration (JSON File Generated)	24
4.5.2 Configuring iBMC IP Information	25
4.5.3 Configuring iBMC Network Information	
4.6 Managing the NTP Service	
4.6.1 Querying NTP Service Information	29
4.6.2 Configure NTP Settings	
4.7 Managing the SNMP Trap Service	
4.7.1 Querying SNMP Service Information (JSON File Generated)	
4.7.2 Configuring SNMP Trap	31

4.8 Importing or Exporting a Profile	33
4.8.1 Importing a Profile	33
4.8.2 Exporting the Profile	35
4.9 Upgrading Firmware	37
4.9.1 Querying Firmware Version (JSON File Generated)	37
4.9.2 Upgrading Firmware	38
4.9.2.1 Out-of-Band Firmware Upgrade	38
4.9.2.2 In-Band Firmware Upgrade	39
4.10 Configuring RAID	41
4.10.1 Querying RAID Configuration (JSON File Generated)	42
4.10.2 Deleting a RAID Array	42
4.10.3 Creating a RAID Array	43
4.10.4 Modifying RAID Array Configuration	44
4.10.5 Modifying RAID Mode	46
4.10.6 Modifying Status of RAID-managed Drives	47
4.11 Deploying the OS	48
4.11.1 Deploying the OS Using Smart Provisioning	48
4.12 BIOS Management	55
4.12.1 Querying BIOS Information (JSON File Generated)	55
4.12.2 Setting BIOS Information	56
4.12.3 Restoring Default BIOS Settings	58
4.13 Log Management	59
4.13.1 Collecting iBMC Logs in One-Click Mode	59
4.13.2 Collecting SELs	60
4.13.3 Clearing SELs.	61
4.14 Common Interface	62
4.15 Local File Transfer	63
4.15.1 Uploading Local Files	63
4.15.2 Downloading a File to a Local Computer	64
4.16 Managing the HTTPS Server Root Certificate	65
4.16.1 Importing the Root Certificate of a Remote HTTPS Server	66
4.16.2 Deleting the Root Certificate of a Remote HTTPS Server	67
4.16.3 Importing a CRL of a Remote HTTPS Server	68
4.17 Querying Security Service Information (JSON File Generated)	70
4.18 Enabling or Disabling HTTPS File Server Certificate Verification	71
4.19 Setting the Web Timeout Period	72
4.19.1 Querying the Web Session Timeout Period (JSON File Generated)	72
4.19.2 Configuring the Web Session Timeout Period	73
A FAQ	75
A.1 How Do I Encrypt Files and View, Edit, and Execute Encrypted Files	
A.1.1 How Do I Encrypt Files	
A 1.2 How Do I View Encrypted Files	76

C Communication Matrix	24
B Obtaining Technical Support	80
A.2 Message invalid upload file Is Displayed When the Root Certificate or CRL of Is Imported	
A.1.4 How Do I Run Configuration Commands After Files Are Encrypted?	77
A.1.3 How Do I Edit Encrypted Files	76

1 Introduction

Function

Integrated in the Ansible software, Ansible plug-in is used to manage servers and connects to the iBMC through the Redfish interface. With this plug-in, you can query, configure, deploy, and upgrade the server.

It supports the following functions:

- Query basic information and health status of a server.
- Set the server boot device.
- Perform power control of the server.
- Manage iBMC users.
- Query and configure iBMC network settings.
- Query and configure the NTP service.
- Query and configure the SNMP service.
- Import or export the server profiles.
- Upgrade the out-of-band and in-band server firmware.
- Query and configure RAID settings.
- Deploy the server OS using ServiceCD 2.0 and Smart Provisioning.
- Manage BIOS (including querying and setting BIOS information and restoring default BIOS settings).
- Manage logs (including collecting iBMC logs in one-click mode, collecting SEL logs, and clearing SEL logs).
- Provide a common public interface.
- Upload local files.
- Download files to a local computer.
- Manage the HTTPS server root certificate.
- Query the security service information.
- Enable or Disable HTTPS file server certificate verification.

NOTICE

The Ansible plugin does not collect or process users' personal data.

Servers Supported by Ansible Plug-in

Architecture	Туре	Model	
x86	Rack server	RH2288H V3	
		2488 V5	
		2288H V5	
		1288H V6	
		2288H V6	
		5288 V6	
		1288H V7	
		2288H V7	
		2288 V7	
		5288 V7	
		5885H V7	
		1258H V7	
x86	Blade server	CH121 V3	
		CH242 V3 DDR4	
		CH121 V5	
		CH242 V5	
		CH121L V5	
		MM921	
		CX621	
		CX320	
x86	High-density server	XH622 V3	
		XH321 V5	
x86	Heterogeneous server	G560 V5	

Matching Versions

Software	Matching Versions
iBMC	V7 server: V3.06.06.03 or later
	V6 server: V3.01.12.23 or later
	V5 server: V325 or later
	V3 server: V323 or later
BIOS	V7 server: V01.01.00.05 or later
	V6 server: V66 or later
	V5 server: V119 or later
	V3 server: V513 or later
Smart Provisioning	V118 or later, which can be downloaded from Smart Provisioning

2 Installing and Uninstalling the Ansible Plug-in

Software Requirements

- Ansible: 2.5.0 or later (2.10 and later versions are recommended)
- Python: 2.7 or later, 3.7 or later
 - □ NOTE

You need to install the requests-toolbelt dependency package (0.9.1 or later) in the Python environment to support some functions.

- 2.1 Installing the Ansible Plug-in
- 2.2 Uninstalling the Ansible Plug-in

2.1 Installing the Ansible Plug-in

Step 1 Download the software package and verify the integrity.

- Obtain the Ansible plug-in installation package (for example, xFusion_iBMC_Ansible_Module_vx.x.zip) and its SHA256 verification file (for example, xFusion_Ansible.sha256.sum) from GitHub.
- 2. Check the integrity of the Ansible plug-in software package (on Linux).
 - Go to the directory where the plug-in installation package and SHA256 verification file are stored.
 - b. Run the **sha256sum -c < (grep** software package name sha256 verification file name) command to verify the software package.

Example: sha256sum -c <(grep xFusion_iBMC_Ansible_Module_vx.x.zip xFusion_Ansible.sha256.sum)

- c. Check whether the verification result is **OK**.
 - If yes, the software package has not been tampered with and can be used.
 - If no, the software package has been tampered with. Obtain a new software package.

- **Step 2** Log in to the Ansible server as the **root** user.
- **Step 3** Upload the installation package to the **root** user directory on the Ansible server.
- **Step 4** Go to the directory where the installation package of the Ansible plug-in is stored.
- **Step 5** Run the following command to decompress the software package of the Ansible plug-in:

unzip xFusion_iBMC_Ansible_Module_x.x.zip

Step 6 Run the following command to go to the **xFusion_iBMC_Ansible_Module** directory generated after the decompression:

cd xFusion_iBMC_Ansible_Module

Step 7 Run the following command to install the Ansible plug-in:

python install.py

◯ NOTE

After the installation is successful, the **ibmc_ansible** folder is added to the **/home** directory. The **ibmc_ansible** folder stores the SSL configuration file of the Ansible plug-in, and the **/home/ibmc_ansible/examples** folder stores the YML sample file that needs to be configured for executing the Ansible plug-in.

----End

2.2 Uninstalling the Ansible Plug-in

- Step 1 Log in to the Ansible server as the root user.
- **Step 2** Go to the directory where the installation package of the Ansible plug-in is stored.

cd xFusion_iBMC_Ansible_Module/

Step 3 Run the uninstallation command.

python uninstall.py

Step 4 Enter **n** (not save) or **y** (save) when a message is displayed asking you whether to save the **.yml** file. In this example, the **.yml** file is not saved.

```
[root@localhost xFusion_iBMC_Ansible_Module]# python uninstall.py start uninstalling xFusion_ibmc_ansible module do you want to keep the yml files?(y/n)
```

Step 5 Enter **n** (not save) or **y** (save) when a message is displayed asking you whether to save the log file and the files generated by the plug-in. In this example, the files are not saved.

```
[root@localhost xFusion_iBMC_Ansible_Module]# python uninstall.py
start uninstalling xFusion_ibmc_ansible module
do you want to keep the yml files?(y/n)
n
do you want to keep the log files and plug-in generation file?(y/n)
n
rm ibmc_ansible log successfully!
```

After the plug-in is uninstalled, a message is displayed, indicating that the uninstallation is successful.

```
[root@localhost xFusion_iBMC_Ansible_Module]# python uninstall.py start uninstalling xFusion_ibmc_ansible module do you want to keep the yml files?(y/n)

n
do you want to keep the log files and plug-in generation file?(y/n)

n
rm ibmc_ansible log successfully!
uninstalling xFusion_ibmc_ansible successfully!
```

----End

3 Configuring the Ansible Plug-in

NOTICE

Files in this section are created in encryption mode. For details about how to encrypt files, view or set encrypted files, and how to run configuration commands after files are encrypted, see A.1 How Do I Encrypt Files and View, Edit, and Execute Encrypted Files.

- 3.1 Configuring /etc/ansible/hosts
- 3.2 Configuring the /group_vars/myhosts File
- 3.3 Configuring the SSL Certificate Authentication and TLS 1.2 Communication Mode

3.1 Configuring /etc/ansible/hosts

Step 1 Run the following command to create the /etc/ansible directory:

mkdir /etc/ansible

Step 2 Run the following command to switch to the **/etc/ansible** directory:

cd /etc/ansible

Step 3 Run the following command to create the hosts file:

vi hosts

Step 4 Write the myhost information to the **hosts** file.

[myhosts] host0 ibmc_ip=192.168.2.20 host=xfusionserver0 host1 ibmc_ip=192.168.2.21 host=xfusionserver1

----End

NOTICE

The names in the first column, for example **host0** and **host1**, cannot be the same. Otherwise, the command is executed only for the last server.

3.2 Configuring the /group_vars/myhosts File

NOTICE

You can perform the configuration as the **root** user or a non-root user. A non-root user is recommended because sensitive data such as passwords is involved during the configuration.

Set parameters in the **myhosts** file in the **/home/user/ibmc_ansible/examples/ group_vars** directory (for non-root users) or **/home/ibmc_ansible/examples/ group_vars** directory (for the **root** user). The parameters include the iBMC user name and password, SFTP/CIFS/SCP user name and password, SNMP community name, the administrator password required for OS deployment, and others.

◯ NOTE

/home/user/ibmc_ansible/examples/group_vars: Replace *user* with the actual non-root user name. In this example, the user is **plugin**.

Procedure (Non-root User)

- **Step 1** Run the following command to switch to the **plugin** user:
 - **su** plugin
- **Step 2** Run the following command to copy the **ibmc_ansible** folder from the **/home** directory to the **/home/plugin** directory.
 - cp -r /home/ibmc_ansible/ /home/p/ugin/ibmc_ansible/
- **Step 3** Run the following command to create the /home/plugin/ibmc_ansible/examples/ group_vars directory:
 - mkdir /home/p/ugin/ibmc_ansible/examples/group_vars
- **Step 4** Run the following command to go to the /home/plugin/ibmc_ansible/examples/ group_vars directory:
 - cd /home/plugin/ibmc_ansible/examples/group_vars
- **Step 5** Run the encryption command to create the **myhosts** file.
 - ansible-vault create myhosts

NOTICE

- Encrypted creation of the myhosts file is recommended because the file contains sensitive data such as passwords. When the file is created with encryption, you need to use the --ask-vault-pass command for decryption. For details, see A.1 How Do I Encrypt Files and View, Edit, and Execute Encrypted Files.
- You can also use the **create myhosts** command to create the file without encryption, which is not recommended due to sensitive data leakage risks.

Step 6 Add the following information to the myhosts file:

```
# Here we define global variables for our server group, but if some servers
# require custom values place these variables in /etc/ansible/hosts to override
# for each individual host
#for create or modify ibmc account
account user: "account user"
account_pswd: "account_pswd"
# input the xfusion ibmc user and password
ibmc user: "ibmc user"
ibmc_pswd: "ibmc_pwd"
# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
sftp pswd: "sftp pwd"
# input the cifs user and password when we need to use the cifs service
cifs_user: "cifs_user"
cifs_pswd: "cifs_pwd"
# input the scp user and password when we need to use the scp service
scp user: "scp user"
scp_pswd: "scp_pwd"
# if you select SNMP Trap mode as V1 or V2C, you can set the community name
community: "community_name"
# input the os password when you deploy the server os by sp
os_pswd: "os_pswd"
```

----End

Procedure (root User)

- **Step 1** Run the following command to create the /home/ibmc_ansible/examples/ group_vars directory:
 - mkdir/home/ibmc_ansible/examples/group_vars
- **Step 2** Run the following command to go to the /home/ibmc_ansible/examples/ group_vars directory:
 - cd /home/ibmc_ansible/examples/group_vars
- **Step 3** Run the encryption command to create the **myhosts** file.
 - ansible-vault create myhosts

NOTICE

- Encrypted creation of the myhosts file is recommended because the file contains sensitive data such as passwords. When the file is created with encryption, you need to use the --ask-vault-pass command for decryption. For details, see A.1 How Do I Encrypt Files and View, Edit, and Execute Encrypted Files.
- You can also use the create myhosts command to create the file without encryption, which is not recommended due to sensitive data leakage risks.

Step 4 Add the following information to the myhosts file:

Here we define global variables for our server group, but if some servers # require custom values place these variables in /etc/ansible/hosts to override

```
# for each individual host
#for create or modify ibmc account
account user: "account user"
account_pswd: "account_pswd"
# input the xfusion ibmc user and password
ibmc user: "ibmc user"
ibmc_pswd: "ibmc_pwd"
# input the sftp user and password when we need to use the sftp service
sftp user: "sftp user"
sftp pswd: "sftp pwd"
# input the cifs user and password when we need to use the cifs service
cifs user: "cifs user"
cifs_pswd: "cifs_pwd"
# input the scp user and password when we need to use the scp service
scp_user: "scp_user"
scp_pswd: "scp_pwd"
# if you select SNMP Trap mode as V1 or V2C, you can set the community name
community: "community_name"
# input the os password when you deploy the server os by sp
os_pswd: "os_pswd"
```

----End

3.3 Configuring the SSL Certificate Authentication and TLS 1.2 Communication Mode

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/set_request_cfg.yml file.

The force_tls1_2 parameter sets the TLS 1.2 communication mode. The default value is True, indicating that the TLS 1.2 communication mode is used forcibly. If "import ssl.PROTOCOL_TLSv1_2 exception" is generated after a command is executed when TLS 1.2 is enabled, you need to set force_tls1_2 to False.

NOTICE

Setting **force_tls1_2** to **False** may pose security risks. Exercise caution when performing this operation.

- The verify and certify parameters set the SSL certificate authentication function. If the certify parameter is not set, one of the following default certificate libraries is used. You need to import the CA certificate to the corresponding library first.
 - If the certifi certificate library is not installed, the system certificate library is used by default. For example:

/etc/pki/tls/certs/ca-bundle.crt

 If Python is used to install the certificate library, the certificate library is used by default. For example: Python 2: /usr/lib/python2.7/site-packages/certifi-2019.11.28-py2.7.egg/certifi/cacert.pem

Python 3: /usr/local/python3/lib/python3.7/site-packages/certifi-2020.6.20-py3.7.egg/certifi/cacert.pem

NOTICE

Disabling SSL certificate authentication (**verify** is set to **False**) may pose security risks. Exercise caution when performing this operation.

• ciphers: cipher suite used when the Ansible plug-in functions as a client to establish sessions with the server.

NOTICE

Use a secure cipher suite. An insecure cipher suite may bring security risks.

```
[plugin@localhost examples]$ vi set_request_cfg.yml
- hosts: 127.0.0.1
 connection: local
 name: set request config
 gather_facts: False
 # verify: the requests module verify server certify or not; Available values: True, False;
 # certify: the certify use to verify the server, if this params do not set, requests module will used the certificate
 #which is in the certifi module or the system default certificate. Format: /etc/pki/tls/certs/ca-bundle.crt
 # force_tls1_2: force to use tls1.2, the default value is true.
  - name: set request config
   ibmc_set_redfish_request_cfg:
    force_tls1_2: False
    verify: False
    certify:
    ciphers: "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-
AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-
AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DHE-DSS-AES256-GCM-SHA384:ECDHE-RSA-
CHACHA20-POLY1305"
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command:

ansible-playbook set_request_cfg.yml

◯ NOTE

This command can be executed only by the root user.

The operation is successful if the following information is displayed:

******	******	******	******	******	******	*****	******
*******	******	*****	******	******	******	*****	****
ok: [127.0.0.1]							
PLAY RECAP ********	******	*****	*****	*****	*****	*****	******
******	******	******	******	******	******	******	******
**							
127.0.0.1	: ok=1	changed=0	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0

4 Using the Ansible Plug-in

The command output varies with the Ansible version used. The command output provided in this section is for reference only.

NOTICE

- This section uses unencrypted files as an example. You are advised to encrypt
 files that contain sensitive data such as passwords. For details about how to
 encrypt files, view encrypted files, configure encrypted files, and run configuration
 commands after files are encrypted, see A.1 How Do I Encrypt Files and View,
 Edit, and Execute Encrypted Files.
- This document uses a non-root user **plugin** as an example. You need to replace **plugin** in the **/home/plugin/**** directory with the actual login user.
- If the HTTPS certificate verification has been performed on the Ansible plug-in's matching iBMC versions, the Ansible plug-in cannot use the remote transmission over HTTPS. Use other protocols.

Querying Help Information

1. Run the following command to view all command module names of the Ansible plug-in. The following uses Python 3 as an example.

ansible-doc -l |grep ibmc

```
[plugin@localhost ~]$ ansible-doc -l |grep ibmc
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86 64.egg/cryptography/hazmat/
bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported
by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it.
utils.CryptographyDeprecationWarning,
ibmc_ansible_show_version
                                    Show xFusion iBMC ansible modules ver...
ibmc clear_sel_logs
                                Clear SEL ...
ibmc_collect_logs
                               Collect iBMC
ibmc collect sel logs
                                Collect iBMC SEL ...
ibmc_common_api
                                 Common...
ibmc create account
                                 Create an ibmc ...
ibmc_create_raid
                               Create vo...
ibmc_delete_account
                                Delete an ibmc ...
ibmc delete https ca
                                delete http...
ibmc_delete_raid
                               Delete vo..
ibmc deploy os by service cd
                                     deploy os by servic...
ibmc_deploy_os_by_sp
                                  deploy os b...
ibmc_download_file
                                Download f...
```

```
ibmc_get_account
                                 Get ibmc user ...
ibmc_get_basic_info
                                 Get server informa...
ibmc_get_bios
                               Get bios ...
                                   get boot de...
ibmc get boot device
ibmc get firmware info by sp
                                       get firmware ...
ibmc_get_ip
                              Get ibmc ip ...
ibmc_get_ntp
                              Get ntp ...
ibmc_get_power_status
                                   get ibmc power ...
                              Get raid ...
ibmc get raid
ibmc_get_security_service_information get security service informa...
ibmc get snmp trap
                                  Get snmp trap resource ...
ibmc_https_ca_import
                                  import http...
ibmc https crl import
                                  import https...
ibmc inband fw update
                                   update inband firm...
ibmc_modify_account
                                   modify an ibmc ...
ibmc modify raid
                                 Modify vo...
ibmc_outband_fw_update
                                    update outband firm...
ibmc profile export
                                 export the server pro...
ibmc_profile_import
                                 import the server pro...
ibmc_reset_bios
                               Reset BIOS resource attrib...
ibmc_set_bios
                              Set bios ...
ibmc_set_boot_device
                                  Set boot de...
ibmc_set_https_cert_verification
                                    set https cert verifica...
ibmc_set_ip
                              Set ibmc ip ...
ibmc_set_ntp
                              Set ntp ...
ibmc_set_power
                                manager server p...
ibmc set redfish request cfg
                                    set request co...
ibmc_set_snmp_trap
                                  Set snmp trap ...
ibmc_upload_file
                                upload ...
```

2. Run the following command to query the help information about a command module:

ansible-doc Command module name

Example: ansible-doc ibmc_get_account

```
[plugin@localhost ~]$ ansible-doc ibmc get account
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/
bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported
by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it.
utils.CryptographyDeprecationWarning,
> IBMC_GET_ACCOUNT (/usr/local/python3/lib/python3.7/site-packages/ansible-2.9.9-py3.7.egg/ansible/
modules/i
    Get ibmc user info
 * This module is maintained by The Ansible Community
OPTIONS (= is mandatory):
= ibmc_ip
    iBMC IP address
    [Default: None]
= ibmc pswd
    iBMC user password used for authentication
    [Default: None]
= ibmc_user
    iBMC user name used for authentication
    [Default: None]
    METADATA:
     status:
     - preview
     supported_by: community
EXAMPLES:
```

```
...skipping...
> IBMC_GET_ACCOUNT (/usr/local/python3/lib/python3.7/site-packages/ansible-2.9.9-py3.7.egg/ansible/
modules/i
    Get ibmc user info
 * This module is maintained by The Ansible Community
OPTIONS (= is mandatory):
= ibmc_ip
    iBMC IP address
    [Default: None]
= ibmc pswd
    iBMC user password used for authentication
    [Default: None]
= ibmc_user
    iBMC user name used for authentication
    [Default: None]
    METADATA:
     status:
     - preview
     supported_by: community
EXAMPLES:
- name: get ibmc account
 ibmc_get_account:
  ibmc_ip: "{{ ibmc_ip }}"
  ibmc_user: "{{ ibmc_user }}"
 ibmc_pswd: "{{ ibmc_pswd }}"
```

Querying Plug-in Version Information

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- Run the following command to view the version:

ansible-playbook -v show_ibmc_ansible_version.yml

- 4.1 Querying Basic Server Information
- 4.2 Configuring the Boot Device
- 4.3 Managing the Power Supply

- 4.4 Managing iBMC Users
- 4.5 Configuring iBMC Network Information
- 4.6 Managing the NTP Service
- 4.7 Managing the SNMP Trap Service
- 4.8 Importing or Exporting a Profile
- 4.9 Upgrading Firmware
- 4.10 Configuring RAID
- 4.11 Deploying the OS
- 4.12 BIOS Management
- 4.13 Log Management
- 4.14 Common Interface
- 4.15 Local File Transfer
- 4.16 Managing the HTTPS Server Root Certificate
- 4.17 Querying Security Service Information (JSON File Generated)
- 4.18 Enabling or Disabling HTTPS File Server Certificate Verification
- 4.19 Setting the Web Timeout Period

4.1 Querying Basic Server Information

Function

- Query the BMC version, BIOS version, CPLD version, Smart Provisioning version, serial number, asset label, server model, server health status, memory information and health status, CPU information and health status, drive information, and health status information.
- By default, a JSON file is generated. To generate a CSV file, set **csv_format** in the **get_basic_info.yml** file to **True** before running the query command.

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- Run the following command to query the basic information about the server: ansible-playbook get_basic_info.yml

The command execution is successful if the following information is displayed:
[plugin@localhost examples]\$ ansible-playbook get_basic_info.yml
PLAY [get bmc basic info] ************************************
TASK [get bmc basic info] ************************************
ok: [host0.domain.com]

The file (for example, **172.26.100.9_BasicInfo.json**) generated after the query is saved in the *I*home/plugin/ansible_ibmc/report/basic_info directory by default. You are advised to export the file before viewing it.

4.2 Configuring the Boot Device

NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Query and set the boot device, boot parameter enabling status, and boot mode.

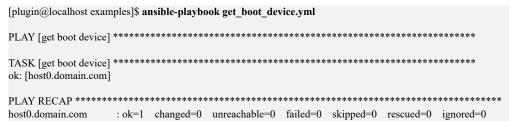
4.2.1 Querying Boot Option Information

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- Run the following command to query the configuration of boot options: ansible-playbook get_boot_device.yml
 - **□** NOTE

To view the query result in the command output, run the **ansible-playbook -vv get_boot_device.yml** command.

The command execution is successful if the following information is displayed:



You can view the query result in the **ansibleibmc.report** file in the **/home/plugin/ansible_ibmc/report**directory, as shown in **3**.

3. Run the following command to view the information:

cd /home/p/ugin/ansible_ibmc/report

cat ansibleibmc.report

[2019-12-02 06:41:10 INFO] - 172.26.100.10 -- Get boot device info successful! The boot device info is: {'Boot': {u'BootSourceOverrideTarget': u'Hdd', u'BootSourceOverrideMode': u'UEFI', u'BootSourceOverrideEnabled': u'Continuous', u'BootSourceOverrideTarget@Redfish.AllowableValues': [u'None', u'Pxe', u'Floppy', u'Cd', u'Hdd', u'BiosSetup']}}

4.2.2 Setting the Boot Device

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/set_boot_device.yml file.

```
[plugin@localhost examples]$ vi set_boot_device.yml
- hosts: myhosts
connection: local
 name: set boot device
 gather facts: False
# boot_target: Current boot device, Available values: Cd, None, Pxe, Floppy, Hdd, BiosSetup.
# boot_enabled: Whether the boot settings are effective, Available values: Disabled, Once, Continuous.
# boot_mode: Boot mode, Available values: UEFI, Legacy.
 tasks:
 - name: set boot device
  ibmc_set_boot_device:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   boot target: "Cd"
   boot_enabled: "Once"
   boot mode: "Legacy"
```

Commands

- 1. Go to the /home/plugin/ibmc_ansible/examples directory.
 - cd /home/p/ugin/ibmc_ansible/examples
- Run the following command to configure the boot device.

ansible-playbook set_boot_device.yml

4.3 Managing the Power Supply

NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Query and set the power supply status.

4.3.1 Querying PSU Status (JSON File Generated)

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to query the power supply status.

ansible-playbook get_power_status.yml

The command execution is successful if the following information is displayed:

The query result is in the **ansibleibmc.report** file in the **/home/plugin/ansible_ibmc/report/ansibleibmc.report** directory.

You can also view the query result in the JSON file (for example, 172.26.100.9_power_status.json, which is saved in the /home/plugin/ansible_ibmc/report/power_status/ directory by default) generated after the query.

3. Run the following commands to view the information:

```
cd /home/plugin/ansible_ibmc/report cat ansibleibmc.report
```

[2019-12-02 06:38:32 INFO] - 172.26.100.10 -- get system power state successful! power status is :Off

4.3.2 Setting the Power Supply Status

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/set_power.yml file.

```
[plugin@localhost examples]$ vi set_power.yml
---
-hosts: myhosts
connection: local
name: power manager
gather_facts: False
#power_cmd: Available values:"poweron" "poweroff" "forcerestart" "gracefulshutdown" "forcepowercycle" "nmi"
tasks:
- name: power manager
ibmc_set_power:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   power_cmd: "poweron"
```

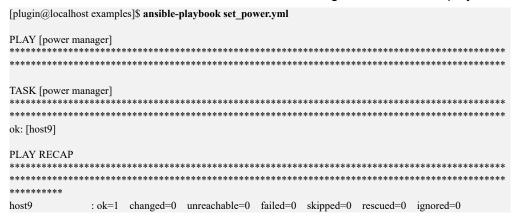
Commands

Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples

2. Run the following command to configure the power supply status.

ansible-playbook set_power.yml

The command execution is successful if the following information is displayed:



4.4 Managing iBMC Users

Function

TQuery, create, modify, or delete iBMC users.

4.4.1 Querying iBMC User Information (JSON File Generated)

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to guery the iBMC user.

ansible-playbook get_account.yml

The JSON file (for example, **172.26.100.9_AccountInfo.json**) generated after the query is saved in the **/home/plugin/ansible_ibmc/report/account_info** directory by default. You are advised to export the JSON file before viewing it.

4.4.2 Creating an iBMC User

Parameter Configuration

 Modify the account_user (new user name) and account_pswd (user password) parameters in the /home/plugin/ibmc_ansible/examples/ group_vars/myhosts file.

```
[plugin@localhost examples]$ vi /home/plugin/ibmc_ansible/examples/group_vars/myhosts
# Here we define global variables for our server group, but if some servers
# require custom values place these variables in /etc/ansible/hosts to override
# for each individual host
#for create or modify ibmc account
account user: "account user"
account_pswd: "account_pswd"
# input the xfusion ibmc user and password
ibmc user: "ibmc user"
ibmc_pswd: "ibmc_pwd"
# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
sftp_pswd: "sftp_pwd"
# input the cifs user and password when we need to use the cifs service
cifs user: "cifs user"
cifs_pswd: "cifs_pwd"
# input the scp user and password when we need to use the scp service
scp_user: "scp_user"
scp_pswd: "scp_pwd"
# if you select SNMP Trap mode as V1 or V2C, you can set the community name
community: "community name"
# input the os password when you deploy the server os by sp
os_pswd: "os_pswd"
```

Modify the /home/plugin/ibmc_ansible/examples/create_account.yml file.

```
[plugin@localhost examples]$ vi create_account.yml
- hosts: myhosts
connection: local
 name: create ibmc Account
gather_facts: False
#roleid: role id; Available values: Administrator, Operator, Commonuser, Noaccess, CustomRole1, CustomRole2,
CustomRole3, CustomRole4
tasks:
 - name: create ibmc Account
  ibmc create account:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   new_account_user: "{{ account_user }}"
   new_account_pswd: "{{ account_pswd }}"
   roleid: "Administrator"
```

Commands

1. Go to the /home/plugin/ibmc_ansible/examples directory.

cd /home/p/ugin/ibmc_ansible/examples

2. Run the following command to create the iBMC user.

ansible-playbook create_account.yml

The command execution is successful if the following information is displayed:

4.4.3 Modify iBMC User Information

Parameter Configuration

Modify the account_user (new user name) and account_pswd (new password)
 parameters in the /home/plugin/ibmc_ansible/examples/group_vars/
 mvhosts file.

```
[plugin@localhost examples]$ vi /home/plugin/ibmc_ansible/examples/group_vars/myhosts
# Here we define global variables for our server group, but if some servers
# require custom values place these variables in /etc/ansible/hosts to override
# for each individual host
#for create or modify ibmc account
account user: "account user"
account_pswd: "account_pswd"
# input the xfusion ibmc user and password
ibmc user: "ibmc user"
ibmc_pswd: "ibmc_pwd"
# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
sftp_pswd: "sftp_pwd"
# input the cifs user and password when we need to use the cifs service
cifs user: "cifs user"
cifs_pswd: "cifs_pwd"
# input the scp user and password when we need to use the scp service
scp user: "scp user"
scp_pswd: "scp_pwd"
# if you select SNMP Trap mode as V1 or V2C, you can set the community name
community: "community name"
# input the os password when you deploy the server os by sp
os_pswd: "os_pswd"
```

Modify the /home/plugin/ibmc_ansible/examples/modify_account.yml file.

```
[plugin@localhost examples]$ vi modify_account.yml
```

```
- hosts: myhosts
connection: local
name: modify ibmc Account
gather_facts: False
#roleid: role id; Available values: Administrator, Operator, Commonuser, Noaccess, CustomRole1, CustomRole2,
CustomRole3, CustomRole4
#locked: it must be False
#enable: Whether the user is enabled; Available values: True, False
#login_interface:list of service the account can access,can be set to empty list []; Available values in list:Web,
SNMP, IPMI, SSH, SFTP, Local, Redfish
#login_rule: list of login rules,can be set to empty list []; Available values in list:Rule1, Rule2, Rule3
#account_insecure_prompt_enabled: enable or disable account insecure prompt; Available values: True, False
tasks:
- name: modify ibmc Account
ibmc_modify_account:
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- 2. Run the following command to modify the iBMC user.

ansible-playbook modify_account.yml

The command execution is successful if the following information is displayed:

4.4.4 Deleting an iBMC User

Parameter Configuration

Modify the **delete_account** parameter (user name to be deleted) in the **/home/ plugin/ibmc_ansible/examples/delete_account.yml** file.

```
[plugin@localhost examples]$ vi delete_account.yml
---
-hosts: myhosts
connection: local
name: delete ibmc Account
gather_facts: False

tasks:
-name: delete ibmc Account
ibmc_delete_account:
ibmc_ip: "{{ ibmc_ip}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_pswd: "{{ ibmc_pswd}}"
delete_account: "test"
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to delete the iBMC user.

ansible-playbook delete_account.yml

The command execution is successful if the following information is displayed:

4.5 Configuring iBMC Network Information

◯ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

- Query iBMC network information.
- Modify the IP enabling mode or IP address separately.

NOTICE

- Do not change the IP enabling mode and IP address at the same time. Otherwise, the modification fails.
- The IPv4 enabling mode and IPv6 enabling mode cannot be switched to each other. Otherwise, the server cannot be connected.

4.5.1 Querying iBMC Network Configuration (JSON File Generated)

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- 2. Run the following command to query the iBMC network information.

ansible-playbook get_ibmc_ip.yml

The JSON file (for example, **172.26.100.9_iBMCIPInfo.json**) generated after the query is saved in the **/home/plugin/ansible_ibmc/report/ibmc_ip** directory by default. You are advised to export the JSON file before viewing it.

4.5.2 Configuring iBMC IP Information

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/set_ibmc_ip.yml file.

```
[plugin@localhost examples]$ vi set_ibmc_ip.yml
- hosts: myhosts
 connection: local
 name: set ibmc ip
 gather_facts: False
# target bmc ip: ibmc ip that you specify to set network information, and you can only choose from the group of hosts.
# ip version: Whether IPv4/IPv6 is enabled, Available values: IPv4, IPv6, IPv4AndIPv6.
# ipv4 addr: IPv4 address info.
 # address: IPv4 address.
 # subnet_mask: Subnet mask of the IPv4 address.
 # gateway: Gateway of the IPv4 address.
 # address origin: How the IPv4 address is allocated. Available values: Static, DHCP.
# ipv6 addr: IPv6 address info.
 # address: IPv6 address.
 # prefix_length: Prefix length of the IPv6 address, must be an integer, value range: 0 to 128.
 # address origin: How the IPv6 address is allocated. Available values: Static, DHCPv6.
# ipv6_gateway: IPv6 gateway address of the iBMC network port.
# hostname: iBMC HostName. Contains a maximum of 64 characters, including only letters, digits, and hyphens(-).
# Cannot start or end with a hyphen.
# domain_name: Domain name. Contains a maximum of 67 characters. The format of FQDN is hostname.domain_name.
# For example, if hostname is "testhostname" and domain name is "ibmc.com", then FQDN is "testhostname.ibmc.com".
 tasks:
 - name: set ibmc ip
  ibmc_set_ip:
   ibmc\_ip: "\{\{\ ibmc\_ip\ \}\}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   target_bmc_ip: "192.168.3.11"
   ip_version: "IPv4AndIPv6"
   ipv4 addr:
    - address: "192.168.2.10"
     subnet mask: "255.255.0.0"
      gateway: "192.168.0.1"
     address_origin: "Static"
   ipv6 addr:
     - address: "fc00:192::10"
     prefix length: 7
      address_origin: "Static"
   ipv6_gateway: "fc00:192::1"
   hostname: "testhostname"
   domain_name: "ibmc.com"
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- Run the following command to configure the iBMC network information. ansible-playbook set_ibmc_ip.yml

The command execution is successful if the following information is displayed:

4.5.3 Configuring iBMC Network Information

Parameter Configuration

◯ NOTE

The **target_bmc_ip** parameter is mandatory.

Modify the /home/plugin/ibmc_ansible/examples/set_ibmc_network.yml file.

```
[plugin@localhost examples] $ vi set_ibmc_network.yml
- hosts: myhosts
 connection: local
 name: set ibmc network
 gather facts: False
# Public parameters:
# ip version: Whether IPv4/IPv6 is enabled, Available values: IPv4, IPv6, IPv4AndIPv6.
# ipv4 subnet mask: Subnet mask of the IPv4 address.
# ipv4 gateway: Gateway of the IPv4 address.
# ipv4_address_origin: How the IPv4 address is allocated. Available values: Static, DHCP.
# ipv6 prefix length: Prefix length of the IPv6 address, must be an integer, value range: 0 to 128.
# ipv6_gateway: IPv6 gateway address of the iBMC network port.
# ipv6_address_origin: How the IPv6 address is allocated. Available values: Static, DHCPv6.
# vlan: iBMC network port VLAN information.
 # vlan_enable: Specifies whether VLAN is enabled, must be a valid boolean.
 # vlan id: Ranges from 1 to 4094.
# dns_address_origin: How DNS IP addresses are allocated. Available values: IPv4, Static, IPv6.
# domain name: Domain name. Contains a maximum of 67 characters. The format of FQDN is "hostname.domain name".
         For example, if hostname is "hostname0" and domain_name is "ibmc.com", then FQDN is
"hostname0.ibmc.com".
         NOTE: To view specific domain name format restrictions, please refer to iBMC's documentation
         or iBMC's redfish interface documentation.
# name servers: Addresses of the preferred and alternate DNS servers if iBMC network port addresses are
         dynamically allocated. The server IP address can be an IPv4 or IPv6 address.
```

```
# IP address of the preferred DNS server.
 # IP address of alternate DNS server 1.
 # IP address of alternate DNS server 2.
# network port mode: Network port mode. Available values: Fixed or Automatic.
# management_network_port: Set the management network port.
 # type: Network port type. Available values: Dedicated, Aggregation, LOM, ExternalPCIe, LOM2 or OCP.
 # port_number: Silkscreen. NOTE: For a dedicated network port, this parameter indicates the serial number
           of the port, not the silkscreen.
# auto mode extend: Auto Mode Extensions. NOTE: The iBMC version must be iBMC V639 or later, and iBMC
3.03.07.17 or later.
 # high priority mode: Enabling status of High Priority Port, must be a valid boolean.
 # high_priority_port: High Priority Port.
  # type: Port type. Available values: Dedicated, Aggregation, LOM, External PCIe, LOM2 or OCP.
  # port number: Silkscreen.
# adaptive_port: Autonegotiation of each network port.
 # type: Network port type. Available values: Dedicated, Aggregation, LOM, ExternalPCIe, LOM2 or OCP.
 # port_number: Silkscreen. NOTE: For a dedicated network port, this parameter indicates the serial number
          of the port, not the silkscreen.
 # adaptive_flag: Autonegotiation flag, must be a valid boolean.
# Private parameters:
# private info list: Private network information list.
 # target_bme_ip: ibme_ip that you specify to set network information, and you can only choose from the group of hosts.
 # hostname: iBMC HostName. Contains a maximum of 64 characters, including only letters, digits, and hyphens(-).
         Cannot start or end with a hyphen.
 # ip_version: The meaning and format is the same as the public parameter.
 # ipv4_addr: IPv4 address information.
  # address: IPv4 address.
  # subnet mask: Subnet mask of the IPv4 address.
  # gateway: Gateway of the IPv4 address.
  # address_origin: How the IPv4 address is allocated. Available values: Static, DHCP.
 # ipv6 addr: IPv6 address information.
  # address: IPv6 address.
  # prefix_length: Prefix length of the IPv6 address, must be an integer, value range: 0 to 128.
  # address origin: How the IPv6 address is allocated. Available values: Static, DHCPv6.
 # ipv6_gateway: The meaning and format is the same as the public parameter.
 # vlan: The meaning and format is the same as the public parameter.
 # dns_address_origin: The meaning and format is the same as the public parameter.
 # domain_name: The meaning and format is the same as the public parameter.
 # name servers: The meaning and format is the same as the public parameter.
 # network_port_mode: The meaning and format is the same as the public parameter.
 # management network port: The meaning and format is the same as the public parameter.
 # auto_mode_extend: The meaning and format is the same as the public parameter.
 # adaptive port: The meaning and format is the same as the public parameter.
# NOTE: 1. When you set the public parameter and the private parameter,
       the private parameter overrides the public parameter,
       but if you don't set the private parameter, the public parameter is used.
     2. To view specific parameter description information, please refer to
       iBMC's documentation or iBMC's redfish interface documentation.
 tasks:
 - name: set ibmc network
  ibmc set network:
   ibmc ip: "{{ ibmc ip }}"
   ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
   # Public parameters
   ip version: "IPv4AndIPv6"
   ipv4 subnet mask: "255.255.0.0"
   ipv4 gateway: "192.168.0.1"
   ipv4_address_origin: "Static"
   ipv6 prefix length: 7
   ipv6_gateway: "fc00:192::1"
   ipv6_address_origin: "Static"
   vlan:
    vlan_enable: true
    vlan id: 1
   dns_address_origin: "Static"
```

```
domain_name: "ibmc.com"
name servers:
 - "192.168.10.254"
 - "192.168.10.253"
- "192.168.10.252"
network_port_mode: "Fixed"
management\_network\_port:
 type: "Dedicated"
 port number: 1
auto_mode_extend:
high priority mode: true
 high_priority_port:
  - type: "Dedicated"
   port number: 1
adaptive_port:
 - type: "Dedicated"
  port_number: 1
  adaptive flag: false
private_info_list:
 # Private parameters
 # The first ip address to be configured
 - target_bmc_ip: "192.168.20.20"
  hostname: "hostname1"
  ipv4 addr:
    - address: "192.168.30.30"
    subnet mask: "255.255.255.0"
    gateway: "192.168.30.1"
    address_origin: "Static"
  ipv6 addr:
   - address origin: "DHCPv6"
  vlan:
   vlan_enable: false
  dns_address_origin: "IPv6"
  network_port_mode: "Automatic"
  management\_network\_port:
   type: "LOM"
   port number: 1
  adaptive_port:
   - type: "LOM2"
    port_number: 1
    adaptive_flag: true
 # The second ip address to be configured
 - target_bmc_ip: "192.168.40.40"
  hostname: "hostname2"
  ipv4_addr:
    - address: "192.168.50.50"
  ipv6 addr:
   - address: "fc00:192::50"
```

◯ NOTE

When you set the public parameter and the personal parameter, the personal parameter overrides the public parameter, but if you don't set the personality parameter, the public parameter is used.

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to configure the iBMC network information.

ansible-playbook set_ibmc_network.yml

The command execution is successful if the following information is displayed: [plugin@localhost examples]\$ ansible-playbook set_ibmc_network.yml

PLAY [set ibmc_network]

4.6 Managing the NTP Service

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

- Query NTP service information.
- Enable or disable the NTP service, configure IP addresses of the preferred and alternate NTP servers, enable or disable server identity authentication, set the NTP address mode (IPv4/IPv6/Static), and minimum/maximum polling interval.

4.6.1 Querying NTP Service Information

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- Run the following command to query the NTP service information.

ansible-playbook get ntp.yml

The query result is in the ansibleibmc.report file in /home/plugin/ansible ibmc/report.

3. Run the following command to view the information:

cd /home/plugin/ansible_ibmc/report cat ansibleibmc.report

[2019-12-02 06:42:10 INFO] - 172.26.100.10 -- Get NTP configuration resource info successful! The NTP configuration resource info is: {'NtpAddressOrigin': u'Static', 'ServiceEnabled': True, 'ServerAuthenticationEnabled': True, 'MinPollingInterval': 3, 'NTPKeyStatus': u'Uploaded', 'AlternateNtpServer': u", 'PreferredNtpServer': u'172.26.207.1', 'MaxPollingInterval': 17}

4.6.2 Configure NTP Settings

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/set_ntp.yml file.

```
[plugin@localhost examples]$ vi set_ntp.yml
- hosts: myhosts
 connection: local
 name: set ntp
 gather_facts: False
# service enabled: Enable or disable bmc ntp service, Available values: True, False.
# pre_ntp_server: Config preferred NtpServer, you can enter ipv4 ipv6 or domain name, NTP Server will be blanked when
set to an empty string.
# alt_ntp_server: Config alternate NtpServer, you can enter ipv4 ipv6 or domain name, NTP Server will be blanked when
set to an empty string.
# server auth enabled: Enable or disable Server Authentication service, Available values: True, False.
# ntp_address_origin: Config Ntp Address Origin, Available values: IPv4, IPv6, Static.
# min polling interval: Config Min Polling Interval time, must be an integer, in 3~17 and <= max polling interval.
# max_polling_interval: Config Max Polling Interval time, must be an integer, in 3~17 and >= min_polling_interval.
 tasks:
 - name: set ntp
  ibmc set ntp:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
   service_enabled: True
   pre_ntp_server: "192.168.2.10" alt_ntp_server: "192.168.2.20"
   server auth enabled: False
   ntp_address_origin: "Static"
   min polling interval: 3
   max_polling_interval: 17
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to configure the NTP:

ansible-playbook set_ntp.yml

The command execution is successful if the following information is displayed:

4.7 Managing the SNMP Trap Service

◯ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function

Function

- Query SNMP trap service information.
- Set the trap enabling status, the SNMPv3 trap user name, reporting mode, host ID, community name, alarm severity, and trap server.

4.7.1 Querying SNMP Service Information (JSON File Generated)

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to query the SNMP service information.

ansible-playbook get_snmp_trap.yml

The JSON file (for example, **172.26.100.9_SNMPTrapInfo.json**) generated after the query is saved in the **/home/plugin/ansible_ibmc/report/snmp_trap** directory by default. You are advised to export the JSON file before viewing it.

4.7.2 Configuring SNMP Trap

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/set_snmp_trap.yml file.

```
[plugin@localhost examples]$ vi set_snmp_trap.yml
---
- hosts: myhosts
connection: local
name: set snmp trap
gather_facts: False

# service_enabled: Whether trap is enabled, Available values: True, False.
# trap_version: Trap version, Available values: V1, V2C, V3.
# trap_v3_user: SNMPv3 user name, valid only for trap version is V3.
```

```
# trap_mode: Trap mode, Available values: OID, EventCode, PreciseAlarm.
# trap server identity: Host identifier, Available values: BoardSN, ProductAssetTag, HostName.
# alarm_severity: Severity levels of the alarm to be sent, Available values: Critical, Major, Minor, Normal.
# trap servers: Can set one or more trap server, When all parameters of the trap server are empty, it indicates that
the trap server is not configured.
# trap_server_enabled: Whether the trap server is enabled, Available values: True, False.
 # trap_server_address: Server address, you can enter ipv4 ipv6 or domain name.
 # trap_server_port: Server port number, must be an integer, Available value range: 1 to 65535.
 - name: set snmp trap
  ibmc_set_snmp_trap:
   ibmc ip: "{{ ibmc ip }}"
   ibmc user: "{{ ibmc user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
community: "{{ community }}"
   service_enabled: True
   trap version: "V3"
   trap_v3_user: "root"
   trap_mode: "OID"
   trap_server_identity: "HostName"
   alarm_severity: "Normal"
   trap_servers:
    - trap_server_enabled: True
     trap_server_address: "192.168.2.10"
     trap_server_port: 160
    - trap_server_enabled: True
     trap_server_address: "192.168.2.11"
     trap server port: 161
     - trap server enabled: False
     trap_server_address: "192.168.2.12"
     trap_server_port: 162
    - trap_server_enabled: False
     trap_server_address: "192.168.2.13"
     trap_server_port: 163
```

 Modify the community parameter in the /home/plugin/ibmc_ansible/ examples/group_vars/myhosts file.

◯ NOTE

When trap_version is set to V1 or V2C, set community in the /home/plugin/ibmc_ansible/examples/group_vars/myhosts file.

```
[plugin@localhost examples]$ vi /home/plugin/ibmc_ansible/examples/group_vars/myhosts
# Here we define global variables for our server group, but if some servers
# require custom values place these variables in /etc/ansible/hosts to override
# for each individual host
#for create or modify ibmc account
account user: "account user"
account pswd: "account pswd"
# input the xfusion ibmc user and password
ibmc user: "ibmc user"
ibmc_pswd: "ibmc_pwd"
# input the sftp user and password when we need to use the sftp service
sftp user: "sftp user"
sftp_pswd: "sftp_pwd"
# input the cifs user and password when we need to use the cifs service
cifs_user: "cifs_user"
cifs_pswd: "cifs_pwd"
# input the scp user and password when we need to use the scp service
scp_user: "scp_user"
scp_pswd: "scp_pwd"
```

```
# if you select SNMP Trap mode as V1 or V2C, you can set the community name community: "community_name"

# input the os password when you deploy the server os by sp
os_pswd: "os_pswd"
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to configure the SNMP trap:

ansible-playbook set_snmp_trap.yml

The command execution is successful if the following information is displayed:

4.8 Importing or Exporting a Profile

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Imports and exports BIOS and iBMC profiles.

- The profile can be imported to the server locally or remotely.
- The profile on the server can be exported to the local environment or a remote path.

4.8.1 Importing a Profile

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/profile_import.yml file.

When importing the file from the local host, set the parameters as follows:

- file_name: name of the file to be imported.
- local_import: local file path to be imported in the Ansible environment.

When importing the file remotely, set the parameters as follows:

- file_name: name of the file to be imported.
- remote_file: file path to be imported in the remote directory.
 - /tmp directory of the iBMC.

SCP.

- Folder in the directory of the remote file server. The format of the file path is
 File transfer protocol:IIIP address of the remote file server/folder. Supported file transfer protocols include SFTP, HTTPS, NFS, CIFS, and
- file_server_user: user name for logging in to the remote file server.
- file_server_pswd: password for logging in to the remote file server.

For local import, set the parameters as follows:

```
[plugin@localhost examples]$ vi profile import.yml
- hosts: myhosts
 connection: local
 name: import profile
 gather_facts: False
# file_name: the file name you want to import
# local_import: local file path of the Ansible environment to be imported.
# remote_import: remote path for saving imported files. The file path can be /tmp on the BMC; or a folder on a remote file
server, the format is protocol://ip/folder
# protocols: Available values: sftp,https,nfs,cifs,scp
# file server user: remote file server user name
# file_server_pswd: remote file server password
 tasks:
 - name: import profile
  ibmc_profile_import:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
   file_name: "192.168.1.1_20210318045050_profile.xml"
   local_import: "/home"
```

For remote import, set the parameters as follows:

```
[plugin@localhost examples]$ vi profile_import.yml
- hosts: myhosts
 connection: local
 name: import profile
 gather_facts: False
# file_name: the file name you want to import
# local import: local file path of the Ansible environment to be imported.
# remote_import: remote path for saving imported files. The file path can be /tmp on the BMC; or a folder on a remote file
server, the format is protocol://ip/folder
# protocols: Available values: sftp,https,nfs,cifs,scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password
 - name: import profile
  ibmc profile import:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   file_name: "192.168.1.1_20210318045050_profile.xml"
   remote import: "sftp://192.168.1.1/data/"
   file_server_user: "{{sftp_user}}}"
   file_server_pswd: "{{sftp_pswd}}}"
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- 2. Run the following command to import the profile:

ansible-playbook profile_import.yml

The command execution is successful if the following information is displayed:

4.8.2 Exporting the Profile

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/profile_export.yml file.

When exporting files to the Ansible environment, configure the following parameters:

- file_name: name of the exported file. This parameter is optional. If this
 parameter is not set, the name of the exported file is IP address_profile.xml by
 default.
- local export: local path for storing exported files in the Ansible environment.

When exporting files to a remote path, configure the following parameters:

- file_name: name of the exported file. This parameter is optional. If this
 parameter is not set, the name of the exported file is IP address_profile.xml by
 default.
- remote_export: remote path for storing exported files.
 - /tmp directory of the iBMC.
 - Folder in the directory of the remote file server. The format of the file path is File transfer protocol: IIIP address of the remote file server/folder.
 Supported file transfer protocols include SFTP, HTTPS, NFS, CIFS, and SCP.
- file_server_user: user name for logging in to the remote file server.
- file_server_pswd: password for logging in to the remote file server.

When exporting files to the Ansible environment, configure the following parameters:

```
[plugin@localhost examples]$ vi profile_export.yml

    hosts: myhosts

 connection: local
 name: import profile
 gather_facts: False
# file_name: the file name you want to import
# local import: local file path of the Ansible environment to be imported.
# remote_import: remote path for saving imported files. The file path can be /tmp on the BMC; or a folder on a remote file
server, the format is protocol://ip/folder
# protocols: Available values: sftp,https,nfs,cifs,scp
# file server user: remote file server user name
# file server pswd: remote file server password
 - name: import profile
  ibmc_profile_import:
   ibmc ip: "{{ ibmc ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc pswd: "{{ ibmc pswd }}"
   file_name: "192.168.1.1_20210318045050_profile.xml"
   local_export: "/home"
```

When exporting files to a remote path, configure the following parameters:

```
[plugin@localhost examples]$ vi profile_export.yml
- hosts: myhosts
 connection: local
 name: import profile
 gather_facts: False
# file_name: the file name you want to import
# local_import: local file path of the Ansible environment to be imported.
# remote_import: remote path for saving imported files. The file path can be /tmp on the BMC; or a folder on a remote file
server, the format is protocol://ip/folder
# protocols: Available values: sftp,https,nfs,cifs,scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password
 tasks:
 - name: import profile
  ibmc_profile_import:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   file_name: "192.168.1.1_20210318045050_profile.xml"
   remote_export: "sftp://192.168.1.1/data/"
   file server_user: "{{sftp_user}}}"
   file_server_pswd: "{{sftp_pswd}}}"
```

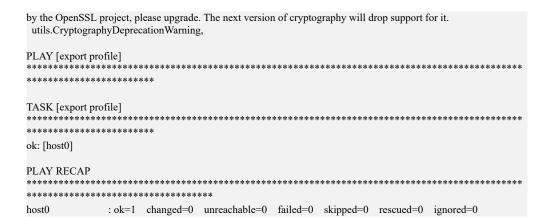
Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to export the profile:

ansible-playbook profile_import.yml

The command execution is successful if the following information is displayed:

```
[plugin@localhost examples]$ ansible-playbook profile_export.yml /usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported
```



4.9 Upgrading Firmware

Function

- Query firmware information.
- Upgrade the BMC, BIOS, and CPLD firmware.
- Upgrade the in-band firmware using Smart Provisioning.

4.9.1 Querying Firmware Version (JSON File Generated)

■ NOTE

- After upgrading the in-band firmware using other tools, you need to restart Smart Provisioning to obtain the latest firmware version information.
- The MM921 management module and CX320/CX621 switch modules do not support this function.

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to query the firmware version.

ansible-playbook get_firmware_info_by_sp.yml

The JSON file (for example, 172.26.100.9_fwlnfo.json) generated after the query is saved in the /home/plugin/ansible_ibmc/report/inband_fw_info directory by default. You are advised to export the JSON file before viewing it.

4.9.2 Upgrading Firmware

4.9.2.1 Out-of-Band Firmware Upgrade

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/update_outband_fw.yml file.

When using a local upgrade file, set the parameters as follows:

local_file: path of the local upgrade file. The format is *directory/file name*, for example, */home/cpldimage.hpm*.

When using a upgrade file in a remote path, set the parameters as follows:

- remote_file: path of the upgrade file in the remote directory.
 - If a temporary directory of the iBMC is used, the directory must be /tmp, and a file name is required, for example tmp/cpldimage.hpm.
 - If a remote file server directory is used, the format of the file path is file transfer protocol:/// address of the remote file server/directory name/file name, for example, sftp://192.168.1.1/data/cpldimage.hpm.
 Supported file transfer protocols include SFTP, HTTPS, NFS, CIFS, and
 - Supported file transfer protocols include SFTP, HTTPS, NFS, CIFS, and SCP.
- file_server_user: user name for logging in to the remote file server.
- file_server_pswd: password for logging in to the remote file server.

◯ NOTE

- The CX320 and CX621 switch modules support only SFTP, and the MM921 management module supports only SFTP and NFS.
- Before the upgrade, ensure that the hpm file in the firmware package has been uploaded to the corresponding directory on the file server. Obtain the firmware package as follows:
 - 1. Visit Support.
 - 2. Click a server model.
 - 3. Click the Software Download tab.
 - 4. Select the patch version.
 - 5. Download the required firmware package.

When using a local upgrade file, set the configuration as follows:

```
[plugin@localhost examples]$ vi update_outband_fw.yml
---
- hosts: myhosts
connection: local
name: update outband fw
gather_facts: False

tasks:
- name: update outband fw
ibmc_outband_fw_update:
ibmc_ip: "{{ ibmc_ip}}"
```

```
ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
local_file: "/home/cpldimage.hpm"
```

When using a upgrade file in a remote path, set the configuration as follows:

```
[plugin@localhost examples]$ vi update_outband_fw.yml
- hosts: myhosts
 connection: local
 name: update outband fw
 gather_facts: False
# local_file: the local firmware file needs to be upgraded, the format is directory/filename, for example: /home/
cpldimage.hpm.
# remote file: remote firmware file. The firmware file can be /tmp on the BMC, the format is directory/filename, for
example: /tmp/cpldimage.hpm;
       # or a firmware file on a remote file server, the format is protocol://ip/folder/filename, for example: sftp://
172.26.200.11/data/cpldimage.hpm.
# protocols: Available values: sftp,https,nfs,cifs,scp
# file server user: remote file server user name
# file_server_pswd: remote file server password
 - name: update outband fw
  ibmc_outband_fw_update:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   remote file: "sftp://192.168.1.1/data/cpldimage.hpm"
   file_server_user: "{{sftp_user}}}"
   file_server_pswd: "{{sftp_pswd}}}"
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the upgrade command:

ansible-playbook update_outband_fw.yml

The command execution is successful if the following information is displayed:

4.9.2.2 In-Band Firmware Upgrade

◯ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Upgrade the in-band firmware using Smart Provisioning. For details about the supported servers and firmware, see the **Smart Provisioning User Guide**.

Parameter Configuration

Modify the image_url parameter in the /home/plugin/ibmc_ansible/examples/update_inband_fw.yml file.

image_url: specifies the path of the upgrade file. It is in the *File transfer protocol:// User name:Password@Server IP address/Directory/File name* format. The file transfer protocols SFTP, HTTPS, NFS, CIFS, and SCP are supported.

Before the upgrade, ensure that the firmware upgrade package and the digital signature file have been uploaded to the corresponding directory on the file server. You can obtain the firmware upgrade package and digital signature file from **FusionServer iDriver**.

```
[plugin@localhost examples]$ vi update inband fw.yml
- hosts: myhosts
connection: local
 name: update inband fw
 gather_facts: False
# image_url: specifies the path of the upgrade file.
# It is in the File transfer protocol://User name:Password@Server IP address/Directory/File name format.
# The file transfer protocols SFTP, HTTPS, NFS, CIFS, and SCP are supported.
# Before the upgrade, ensure that the firmware upgrade package and the digital signature
# file have been uploaded to the corresponding directory on the file server.
 tasks:
 - name: update inband fw
  ibmc inband fw update:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc user: "{{ ibmc user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   image url:
    - "sftp://172.26.200.11/data/NIC-LOM-X722-10GE_SFP-GE_Electrical-FW-3.33_0x80000f09.zip"
   file_server_user: "{{sftp_user}}}"
   file_server_pswd: "{{sftp_pswd}}}"
```

Modify the user name and password of the file server in the **myhosts** file under / home/plugin/ibmc_ansible/examples/group_vars.

```
[plugin@localhost examples]$ vi /home/plugin/ibmc_ansible/examples/group_vars/myhosts
---

# Here we define global variables for our server group, but if some servers
# require custom values place these variables in /etc/ansible/hosts to override
# for each individual host

# for create or modify ibmc account
account_user: "account_user"
account_pswd: "account_pswd"

# input the xfusion ibmc user and password
ibmc_user: "ibmc_user"
ibmc_pswd: "ibmc_pwd"

# input the sftp user and password when we need to use the sftp service

* sftp_user: "sftp_user"

* sftp_pswd: "sftp_pwd"
```

```
# input the cifs user and password when we need to use the cifs service
cifs_user: "cifs_user"
cifs_pswd: "cifs_pwd"

# input the scp user and password when we need to use the scp service
scp_user: "scp_user"
scp_pswd: "scp_pwd"

# if you select SNMP Trap mode as V1 or V2C, you can set the community name
community: "community_name"

# input the os password when you deploy the server os by sp
os pswd: "os pswd"
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the upgrade command:

ansible-playbook update_inband_fw.yml

The command execution is successful if the following information is displayed:

4.10 Configuring RAID

□ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

- Configure only the RAID controller cards that support out-of-band management.
- Support the scenario where multiple RAID controller cards are configured.
- The supported RAID controller cards: LSI SAS3108, Avago SAS3408iMR, Avago SAS3004iMR, Avago SAS3508, BROADCOM SAS3908, and Avago SAS9460.
- Supported functions:
 - Querying information of the RAID controller cards.
 - Creating, modifying, and deleting a RAID array.
 - Modifying RAID controller card modes and the status of RAID-managed drives.

4.10.1 Querying RAID Configuration (JSON File Generated)

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to query the RAID configuration:

ansible-playbook get_raid.yml

The JSON file (for example, **172.26.100.9_RAIDInfo.json**) generated after the query is saved in the **/home/plugin/ansible_ibmc/report/raid** directory by default. You are advised to export the JSON file before viewing it.

4.10.2 Deleting a RAID Array

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/delete_raid.yml file.

```
[plugin@localhost examples]$ vi delete_raid.yml
- hosts: myhosts
 connection: local
 name: delete raid
 gather_facts: False
# storage id: ID of the storage resource
 # 1.Delete one RAID storage, Format: RAIDStorage+Controller_ID
 # 2.Delete multiple RAID storage, Separated by commas, Format: RAIDStorage+Controller ID1,RAIDStorage
+Controller_ID2,
 # 3.Delete all RAID storage, Format: all
# volume id: Volume resource ID
 # 1.Delete one volume, Format: LogicalDrive+Volume ID
 # 2.Delete multiple volume, Separated by commas, Format: LogicalDrive+Volume ID1,LogicalDrive+Volume ID2,...
 #3.Delete all volume, Format: all
 tasks:
 - name: delete raid
  ibmc_delete raid:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc user: "{{ ibmc user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   storage_id: "RAIDStorage0,RAIDStorage1"
   volume_id: "LogicalDrive0,LogicalDrive1"
```

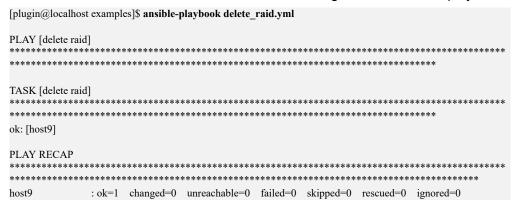
Commands

Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples

2. Run the following command to delete the RAID:

ansible-playbook delete_raid.yml

The command execution is successful if the following information is displayed:



4.10.3 Creating a RAID Array

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/create_raid.yml file.

NOTICE

Different RAID controller cards support different configurable parameters. For details, see the V2 and V3 Server RAID Controller Card User Guide, V5 Server RAID Controller Card User Guide, v6 Server RAID Controller Card User Guide, or V7 Server RAID Controller Card User Guide.

```
[plugin@localhost examples]$ vi create raid.yml
- hosts: myhosts
 connection: local
 name: create raid
 gather facts: False
# storage id: ID of the storage resource. Format: RAIDStorage+Controller ID
# capacity_mbyte: Volume capacity, must be an integer, the size unit is MB. It is an optional parameter
# stripe_size: Stripe size of a volume, must be an integer. It is an optional parameter. Available values: 65536, 131072,
262144, 524288, 1048576
# cachecade flag: Whether it is a CacheCade volume. It is an optional parameter, Available values: True, False
# drives: Member disk list number. It is a mandatory parameter. Format: "1,2...N"
# volume_raid_level: RAID level of the volume. It is a mandatory parameter. Available values: RAID0, RAID1, RAID5,
RAID6, RAID10, RAID50, RAID60
# volume_name: Volume name. It is an optional parameter. A string of up to 15 bytes. Value range: ASCII code
corresponding to 0x20 to 0x7E
# df_read_policy: Default read policy of the volume. It is an optional parameter. Available values: NoReadAhead,
ReadAhead
# df_write_policy: Default write policy of the volume. It is an optional parameter. Available values: WriteThrough,
WriteBackWithBBU, WriteBack
# df cache policy: Default cache policy of the volume. It is an optional parameter. Available values: CachedIO, DirectIO
# span_num: Number of spans of the volume, must be an integer. It is an optional parameter
 # 1.Set this parameter to 1 when creating a RAID0, RAID1, RAID5, or RAID6 array.
 # 2.Set this parameter to a value from 2 to 8 when creating a RAID10, RAID50, or RAID60 array.
# access_policy: Volume access policy. It is an optional parameter. Available values: ReadWrite, ReadOnly, Blocked
# disk cache policy: Cache policy for member disks. It is an optional parameter. Available values: Unchanged, Enabled,
Disabled
# init_mode: Volume initialization mode. It is an optional parameter. Available values: UnInit, QuickInit, FullInit
```

```
tasks:
- name: create raid
 ibmc create raid:
  ibmc_ip: "{{ ibmc_ip }}"
  ibmc_user: "{{ ibmc_user }}"
  ibmc_pswd: "{{ ibmc_pswd }}"
  volumes:
   - storage_id: "RAIDStorage0"
   capacity mbyte: 1000
   stripe_size: 65536
   cachecade flag: False
   drives: "0,1"
   volume_raid_level: "RAID0"
   volume name: "volume name"
   df_read_policy: "NoReadAhead"
   df write policy: "WriteThrough"
   df_cache_policy: "CachedIO"
   span_num: 1
   access_policy: "ReadWrite"
   disk_cache_policy: "Unchanged"
   init mode: "UnInit"
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to create a RAID:

ansible-playbook create_raid.yml

The command execution is successful if the following information is displayed:

4.10.4 Modifying RAID Array Configuration

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/modify_raid.yml file.

NOTICE

Different RAID controller cards support different parameters. For details, see the V2 and V3 Server RAID Controller Card User Guide, V5 Server RAID Controller Card User Guide, or V7 Server RAID Controller Card User Guide.

```
[plugin@localhost examples]$ vi modify_raid.yml
- hosts: myhosts
 connection: local
 name: modify raid
 gather_facts: False
# storage_id: ID of the storage resource. Format: RAIDStorage+Controller_ID
# volume_id: Volume resource ID. Format: LogicalDrive+Volume_ID
# volume_name: Volume name. It is an optional parameter. A string of up to 15 bytes. Value range: ASCII code
corresponding to 0x20 to 0x7E
# df read policy: Default read policy of the volume. It is an optional parameter. Available values: NoReadAhead,
# df_write_policy: Default write policy of the volume. It is an optional parameter. Available values: WriteThrough,
WriteBackWithBBU, WriteBack
# df_cache_policy: Default cache policy of the volume. It is an optional parameter. Available values: CachedIO, DirectIO
# boot enable: Whether it is the boot device. Available values: True.
# bgi_enable: Whether background initialization is enabled. Available values: True, False.
# access policy: Volume access policy. It is an optional parameter. Available values: ReadWrite, ReadOnly, Blocked
# ssd_cache_enable: Whether the CacheCade volume is used as the cache. Available values: True, False.
# disk_cache_policy: Cache policy for member disks. It is an optional parameter. Available values: Unchanged, Enabled,
Disabled
 tasks:
 - name: modify raid
  ibmc_modify_raid:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   volumes:
    - storage id: "RAIDStorage0"
    volume_id: "LogicalDrive0"
    volume_name: "volume name"
    df_read_policy: "NoReadAhead"
    df_write_policy: "WriteThrough"
    df_cache_policy: "CachedIO"
    boot enable: True
    bgi enable: False
    access_policy: "ReadWrite"
    ssd cache enable: False
    disk_cache_policy: "Unchanged"
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to modify the RAID configuration:

ansible-playbook modify_raid.yml

The command execution is successful if the following information is displayed:

4.10.5 Modifying RAID Mode

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/modify_storage_controller.yml file.

NOTICE

Different RAID controller cards support different parameters. For details, see *V2 and V3 Servers RAID Controller Card User Guide*, *V5 Server RAID Controller Card User Guide*, or *V7 Server RAID Controller Card User Guide*.

```
[plugin@localhost examples]$ vi modify_storage_controller.yml
- hosts: myhosts
connection: local
 name: modify storage controller
 gather facts: False
# storager_id: ID of the storage resource. Format: RAIDStorage+Controller_ID
# mode: Working mode of the RAID controller. Available values: RAID, JBOD.
# Setting the working mode may require restarting the OS to take effect, please check the echo information.
# Cannot set with JBOD_state at the same time.
# JBOD state: Specifies whether to enable the hard disk pass-through function. Available values: True, False.
# This property can only be set when the working mode of the RAID controller is RAID.
# Cannot set with mode at the same time.
 tasks:
 - name: modify storage controller
  ibmc_modify_storage_controller:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   storage_controllers:
    - storage id: "RAIDStorage0"
    JBOD state: True
    - storage_id: "RAIDStorage1"
     mode: "JBOD"
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- 2. Run the following command to modify the RAID configuration: ansible-playbook -v modify_storage_controller.yml

The command execution is successful if the following information is displayed. Then, reboot the OS so that the configuration takes effect.

4.10.6 Modifying Status of RAID-managed Drives

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/modify_drive.yml file.

NOTICE

Different RAID controller cards support different parameters. For details, see *V2 and V3 Servers RAID Controller Card User Guide*, *V5 Server RAID Controller Card User Guide*, or *V7 Server RAID Controller Card User Guide*, or *V7 Server RAID Controller Card User Guide*.

```
[plugin@localhost examples]$ vi modify_drive.yml
- hosts: myhosts
 connection: local
 name: modify drive
 gather_facts: False
# drive id: The ID of the drive resource
# firmware_status: Drive status, The drive status can switch between the following states:
# After creating a logical drive, the drive status can be switched between "Online" and "Offline"
# When the controller mode is RAID and JBOD State is true, the drive state can switch between "UnconfiguredGood" and
# When the controller EpdEnable is true, the drive status can switch between "UnconfiguredGood" and "Online"
#"UnconfiguredBad" to "UnconfiguredGood"
 tasks:
 - name: modify drive
  ibmc modify drive:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   drives:
   - drive id: "HDDPlaneDisk0"
    firmware_status: "UnconfiguredGood"
   - drive_id: "HDDPlaneDisk1"
    firmware status: "JBOD"
   - drive id: "HDDPlaneDisk2"
    firmware status: "Online"
   - drive_id: "HDDPlaneDisk3"
    firmware_status: "Offline"
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- 2. Run the following command to modify the RAID configuration:

ansible-playbook -v modify_drive.yml

The command execution is successful if the following information is displayed:

```
[plugin@localhost examples]$ ansible-playbook -v modify drive.yml
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/
bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported
by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it.
utils.CryptographyDeprecationWarning,
Using /etc/ansible/ansible.cfg as config file
PLAY [modify drive]
*******************************
TASK [modify drive]
*************************
ok: [host0] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"}, "changed": false, "msg":
"Modify drive configuration successfully, the result is: [{'result': True, 'msg': 'Modify drive: HDDPlaneDisk0
configuration successfully.'}]"}
PLAY RECAP
**********************************
*************************
             : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

4.11 Deploying the OS

- The MM921 management module and CX320/CX621 switch modules do not support this function.
- The RAID configuration is complete for the server, on which the OS is to be deployed.
- If ServiceCD2.0 is used, the logical drive on the server where the OS is to be deployed cannot exceed 2 TB. Otherwise, ServiceCD2.0 cannot identify the logical drive.

4.11.1 Deploying the OS Using Smart Provisioning

For details about the servers supported by Smart Provisioning, see the **Smart Provisioning User Guide**.

◯ NOTE

The commands deploy_centos7u3_by_sp.yml, deploy_esxi65_by_sp.yml, and deploy_win2012r2_by_sp.yml can be used. The following uses deploy_centos7u3_by_sp.yml as an example.

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/deploy_centos7u3_by_sp.yml file.

For details about the parameters, see **Table 4-1**.

 $[plugin@localhost\ examples] \$\ vi\ deploy_centos 7u3_by_sp.yml$

```
- hosts: myhosts
 connection: local
 name: ibmc deploy centos7u3 by sp
 gather_facts: False
#os img: The os image path; Format: protocol://Username:Password@IPaddress/Folder/image file; Available protocol:
nfs,cifs,https
#OSType:Os type; Available values:RHEL6U9, RHEL6U10, RHEL7U3, RHEL7U4, RHEL7U5, RHEL7U6, CentOS6U9,
CentOS6U10, CentOS7U3, CentOS7U4, CentOS7U5, CentOS7U6, ESXi6.0, ESXi6.5, ESXi6.7
# SLES11SP4, SLES12SP2, SLES12SP3, Ubuntu16.04, Ubuntu16.04.1, Ubuntu16.04.2, Win2016, Win2016 Standard
Desktop, Win2016 Standard Core, Win2016 Datacenter Desktop
# Win2016 Datacenter Core, Win2012_R2, Win2012_R2 Standard Desktop, Win2012_R2 Standard Core, Win2012_R2
Datacenter Desktop, Win2012 R2 Datacenter Core
# For more details about the servers supported by Smart Provisioning, see the Smart Provisioning User Guide.
#InstallMode: OS Installation mode; Available values:Recommended, Customized
#Language: Available values: Please refer to the installation guide of the OS.
#TimeZone: Available values: Please refer to the installation guide of the OS.
#Keyboard: Available values: Please refer to the installation guide of the OS.
#BootType: Bios boot mode, This parameter is optional; Available values: UEFIBoot, LegacyBoot, SecureBoot
#CDKey: key of the OS Installation, This parameter is optional
#RootPwd: Root user password, this parameter is mandatory;
# Windows: a sting of at least 6 characters.SUSE: a sting of at least 6 characters.
# Centos/Redhat/ EulerOS: a sting of at least 6 characters excluding #,$, and space.
# Ubuntu: a sting of at least 8 characters excluding #, $, and space.
# Vmware: a string of 7 to 40 characters. For the ESXi 6.7, the password must consist of letters, digits, and special
# (NOTE: Smart Provisioning supports special characters #, $ and spaces from V119.)
#HostName: Host Name, This parameter is optional Installation
#CheckFirmware: Whether to verify firmware. This parameter is optional; Available values: True, False
#Partition: Partition information. This parameter is optional. Please refer to the installation guide of the OS
#Software: Software list. This parameter is mandatory.Format: { "FileName": "iBMA" }
#Autopart: Whether auto-partitioning is supported. Linux/VMware: true Window: false
#MediaType:Type of the media where the OS can be deployed. This parameter is optional; Available values: SANBoot,
Disk, USB
#AutoPosition: Whether the installation drive is automatically selected; Available values:True
#NetCfg: Network config
 tasks:
 - name: ibmc deploy centos7u3 by sp
  ibmc_deploy_os_by_sp:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   os_img: "nfs://172.26.200.11/data/centeros7u3.iso"
   os config:
    InstallMode: "Recommended"
    OSType: "CentOS7U3"
    BootType: "UEFIBoot"
    CDKey: ""
    RootPwd: "{{ os_pswd }}"
    HostName: "test"
    Language: "en_US.UTF-8"
    TimeZone: "America/New_York"
    Keyboard: "us"
    CheckFirmware: False
    Partition: []
    Autopart: True
    AutoPosition: True
    Software: []
    NetCfg:
      - Device:
         Name: "eth10086"
         MAC: "**:**:**:**:**
       IPv4Addresses:
         - Address: "192.168.2.44"
          SubnetMask: "255.255.0.0"
          Gateway: "192.168.2.1"
          AddressOrigin: "Static"
       IPv6Addresses:
         - Address: ""
```

PrefixLength: ""
Gateway: ""
AddressOrigin: "Static"

NameServers:
- DNS: "192.168.2.1"
- DNS: "192.168.2.2"

Table 4-1 Parameters

Parameter	Description	Value	
os_img	OS image path (mandatory).	Format: File Transfer Protocol:// User name:Password@Server IP address/Directory/File name	
		The file transfer protocols SFTP, HTTPS, NFS, CIFS, and SCP are supported.	
InstallMode	Installation mode (mandatory).	Recommended Customized	
OSType	Type of the OS to be installed (mandatory).	Value: RHEL6U9, RHEL6U10, RHEL7U3, RHEL7U4, RHEL7U5, RHEL7U6, CentOS6U9, CentOS6U10, CentOS7U3, CentOS7U4, CentOS7U5, CentOS7U6, ESXi6.0, and ESXi6.5 ESXi6.7, SLES11SP4, SLES12SP2, SLES12SP3, Ubuntu16.04, Ubuntu16.04.1, Ubuntu16.04.2, Win2016, Win2016 Standard Desktop, Win2016 Datacenter Desktop, Win2016 Datacenter Core, Win2012_R2, and Win2012_R2 Standard Desktop, Win2012_R2 Standard Core, Win2012_R2 Datacenter Desktop, Win2012_R2 Datacenter Core NOTE For more details about the servers supported by Smart Provisioning, see the Smart Provisioning User Guide.	
BootType	BIOS boot mode (optional).	UEFIBootLegacyBootSecureBoot	

Parameter	Description	Value	
CDKey	OS installation key (optional).	 For the Windows or VMware OS, this parameter is optional and can be set to a 25-digit value with every five digits connected by a hyphen (-). The value can contain uppercase letters (A to Z), lowercase letters (a to z), and digits (0 to 9). For Linux, this parameter is left empty. 	
RootPwd	Initial password of the administrator (mandatory).	 Set this parameter in the myhosts file based on the following rules: Windows: The parameter value must contain at least six digits. SUSE: The parameter value must contain at least six digits. CentOS, Red Hat and EulerOS: The parameter value must contain at least six digits and cannot contain a "#", "\$", or space. Ubuntu: The password must contain at least 8 characters and cannot contain #, \$, or spaces. VMware OS: The parameter value must contain at least seven digits. For VMware ESXi 6.7, the parameter value must contain at least three types of characters, including letters, digits, and special characters and cannot exceed 40 digits. NOTE Smart Provisioning V119 and later versions support the following special characters: #\$. 	

Parameter	Description	Value
HostName	Host name (optional).	The value contains a maximum of 15 characters, including uppercase letters (A to Z), lowercase letters (a to z), digits (0 to 9), and hyphens (-). • For Linux, this parameter is optional and takes effect only after the network is configured. • For Windows, this parameter is optional. • For VMware OS, this parameter is optional and takes effect only after the network is configured.
Language	System language (mandatory).	The parameter is a string of characters. For details, see the installation guide of the OS. • For Linux, this parameter is mandatory. • For Windows, this parameter is mandatory. • For VMware OS, leave it blank.
TimeZone	System time zone (mandatory).	The parameter is a string of characters. For details, see the installation guide of the OS. • For Linux, this parameter is mandatory. • For Windows, this parameter is mandatory. • For VMware OS, leave it blank.
Keyboard	System keyboard type (mandatory).	 The parameter is a string of characters. For details, see the installation guide of the OS. For Linux, this parameter is mandatory. For Windows, this parameter is mandatory. For VMware OS, leave it blank.
CheckFirmware	Specifies whether to verify firmware. This parameter is mandatory.	true false

Parameter	Description	Value		
Partition	Partition information (optional). The format is as follows: Partition: - Size: "64" FileSystem: "NTFS" Name: "C"	 Windows: The value of Name is a string of characters from C to Z. Set FileSystem to NTFS. The value of Size is greater than 32. If the value is max, the entire disk is used as the data disk. Linux: The value of Name cannot contain <> :& or spaces, for example, I, Ihome, and swap. The value of FileSystem can be ext4, ext3, ext2, or xfs. The value of Size is greater than 0. The root partition size must be greater than 10, and the swap partition size must be greater than 1. If the value is max, the remaining space is allocated. The VMware OS does not support this function. 		
Autopart	Specifies whether automatic partitioning is supported (mandatory).	 For Linux and VMware OS, the value is true. For Windows, the value is false. 		
MediaType	Media type that supports system deployment (optional). If this parameter is not specified, the system is deployed on the local drive.	 SANBoot: Deploy the OS on the SANBoot drive (only VMware 6.5.1 and VMware 6.7 are supported). Disk: Deploy the OS on the drive. USB: Deploy the OS on the USB device (only VMware 6.5 is supported). 		
AutoPosition	Specifies whether auto- selection of the installation drive is supported (mandatory).	Value: true (The installation drive can only be automatically selected now.)		
Software	List of software to be installed (mandatory). The format is as follows: Software: - FileName: "iBMA"	iBMA		

Parameter	Description	Value	
NetCfg	Network configuration (optional).	[] or configure the following parameters: NOTE [] indicates that no device is specified and batch deployment is supported.	
		Device: device network information.	
		 Name: name of the network port on the NIC of the server to be deployed. 	
		 MAC: device MAC address. 	
		IPv4Addresses: IPv4 address information of the network port.	
		 Address: IPv4 address. 	
		 SubnetMash: subnet mask. 	
		 AddressOrigin: mode for obtaining the IPv4 address. It can be Static or DHCP. 	
		 Gateway: IPv4 gateway address. 	
		 IPv6Addresses: IPv6 address information of the network port. 	
		 Address: IPv6 address. 	
		 PrefixLength: prefix length of the IPv6 address. 	
		 AddressOrigin: mode for obtaining the IPv6 address. It can be Static or DHCP. 	
		 Gateway: IPv6 gateway address. 	
		NOTE This option is not supported by Ubuntu and VMware OS.	
		NameServers: IP address of the DNS server. It can be an IPv4 or IPv6 address.	

The example values of **OSType**, **Language**, **TimeZone**, and **Keyboard** are as follows:

ОЅТуре	Language	TimeZone	Keyboard
RHEL/CentOS/ EulerOS/Ubuntu	en_US.UTF-8	America/New_York	us

ОЅТуре	Language	TimeZone	Keyboard
SLES	en_US	America/New_York	english-us
Windows	en-US	Eastern Standard Time	0x00000409

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- 2. Run the following deployment commands:

ansible-playbook deploy_centos7u3_by_sp.yml

The command execution is successful if the following information is displayed:

4.12 BIOS Management

NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Query and set BIOS information and restore the default BIOS settings.

4.12.1 Querying BIOS Information (JSON File Generated)

(Optional) Parameter Settings

Modify the **bios_attribute** parameter in the **/home/plugin/ibmc_ansible/examples/get_bios.yml** file.

bios_attribute: Specifies the BIOS parameters to be queried. This parameter is optional. If this parameter is not specified, information about all BIOS parameters is displayed.

The *IP address_BIOSInfo.json* file is generated and saved in the */home/plugin/ansible_ibmc/report/bios/* folder.

- IP address_BIOSInfo.json: Set the IP address to the configured server IP address.
- /home/plugin/ansible_ibmc/report/bios/: Replace plugin with the actual user name.

```
[plugin@localhost examples]$ vi get_bios.yml
---
- hosts: myhosts
connection: local
name: get bios
gather_facts: False

# bios_attribute: User-specified BIOS attributes to be queried

tasks:
- name: get bios
ibmc_get_bios:
ibmc_ip: "{{ ibmc_ip}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_pswd: "{{ ibmc_pswd}}"
bios_attribute:
- QuickBoot
- QuietBoot
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to query the BIOS information.

ansible-playbook get_bios.yml

4.12.2 Setting BIOS Information

Parameter Configuration

Modify the values of **Immediately** and **bios_attribute** in the **/home/**plugin/**ibmc_ansible/examples/set_bios.yml** file.

Immediately: Specifies the effective time. The options are True and False. The
default value is False. True indicates that the server is restarted immediately for
the settings to take effect. False indicates that you need to restart the server for
the settings to take effect.

NOTICE

Automatic server restart affects services. Exercise caution when performing this operation.

• **bios_attribute**: BIOS parameters to be set. This parameter is mandatory.

Ⅲ NOTE

The following example describes how to set **QuickBoot** and **QuietBoot**. For details about configurable BIOS parameters, see the **Server Purley Platform BIOS Parameter Reference**.

```
[plugin@localhost examples]$ vi set_bios.yml
- hosts: myhosts
 connection: local
 name: set ibmc bios
 gather_facts: False
# Immediately: Whether to restart the system immediately for the configuration to take effect: True, False
# bios_attribute: BIOS attributes set by the user
 tasks:
 - name: set ibmc bios
  ibmc set bios:
   ibmc ip: "{{ ibmc ip }}"
   ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
   Immediately: False
   bios attribute:
      QuickBoot: Disabled
      QuietBoot: Enabled
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to set the BIOS information:

ansible-playbook set_bios.yml

The command execution is successful if the following information is displayed:

4.12.3 Restoring Default BIOS Settings

Parameter Configuration

Modify the **Immediately** parameter in the **/home/plugin/ibmc_ansible/examples/reset_bios.yml** file.

Immediately: Specifies the effective time. The options are **True** and **False**. The default value is **False**.

- **True**: The server automatically restarts for the upgrade to take effect.
- False: You need to restart the server for the settings to take effect.

NOTICE

Automatic server restart affects services. Exercise caution when performing this operation.

```
[plugin@localhost examples]$ vi reset_bios.yml
---
-hosts: myhosts
connection: local
name: reset ibmc bios
gather_facts: False

# Immediately: Whether to restart the system immediately for the configuration to take effect: True, False

tasks:
- name: reset ibmc bios
ibmc_reset_bios:
ibmc_ip: "{{ ibmc_ip; "{{ ibmc_user}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_pswd: "{{ ibmc_pswd}}"
Immediately: False
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to reset the BIOS information:

ansible-playbook reset_bios.yml

The command execution is successful if the following information is displayed:

```
[plugin@localhost examples]$ ansible-playbook reset_bios.yml
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/
bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported
by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it.
utils.CryptographyDeprecationWarning,

PLAY [reset ibmc bios]
```

4.13 Log Management

◯ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Collect iBMC logs in one-click mode, collect SEL logs, and clear SEL logs.

4.13.1 Collecting iBMC Logs in One-Click Mode

Parameter Configuration

Modify the /home/p/ugin/ibmc_ansible/examples/collect_ibmc_logs.yml file.

- save_mode: Files are stored on a remote file server or a local file server. To
 store files on a remote file server, set this parameter to sftp, https, nfs, cifs, or
 scp. For local storage, set this parameter to local.
- **file_server_ip**: IP address of the remote file server. This parameter is required when **save_mode** is set to **sftp**, **https**, **nfs**, **cifs**, or **scp**.
- file_server_user: user name for logging in to the remote file server. This
 parameter is required when save_mode is set to sftp, https, cifs, or scp.
- **file_server_pswd**: password for logging in to the remote file server. This parameter is required when **save_mode** is set to **sftp**, **https**, **cifs**, or **scp**
- file_name: log file name and path If only the file name is specified, the log file is saved in the /home/plugin/ansible_ibmc/report/collect_IBMC_log/ directory by default. In the directory, plugin indicates the actual user name.

```
[plugin@localhost examples]$ vi collect_ibmc_logs.yml
---
- hosts: myhosts
connection: local
name: collect logs
gather_facts: False

# save_mode: place to save logs: local, sftp, https, nfs, cifs, scp
# file_server_ip: ip address of file server, if save_mode is local, this parameter can be left blank.
# file_server_user: the user of file server
# file_server_pswd: the password of file server
# file_name: Log file storage path and file name

tasks:
- name: collect logs
ibmc_collect_logs:
ibmc_ip: "{{ ibmc_ip}}"
ibmc_user: "{{ ibmc_user}}"
```

```
ibmc_pswd: "{{ ibmc_pswd }}"
save_mode: "sftp"
file_server_ip: "sftp_server_ip"
file_server_user: "{{ sftp_user }}"
file_server_pswd: "{{ sftp_pswd }}"
file_name: "/usr/dump.tar.gz"
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- 2. Run the following command to collect iBMC logs in one-click mode:

ansible-playbook collect_ibmc_logs.yml

The command execution is successful if the following information is displayed:

4.13.2 Collecting SELs

Parameter Configuration

Modify the /home/p/ugin/ibmc_ansible/examples/collect_sel_logs.yml file.

- save_mode: Files are stored on a remote file server or a local file server. To
 store files on a remote file server, set this parameter to sftp, https, nfs, cifs, or
 scp. For local storage, set this parameter to local.
- **file_server_ip**: IP address of the remote file server. This parameter is required when **save_mode** is set to **sftp**, **https**, **nfs**, **cifs**, or **scp**.
- file_server_user: user name for logging in to the remote file server. This
 parameter is required when save_mode is set to sftp, https, cifs, or scp.
- file_server_pswd: password for logging in to the remote file server. This
 parameter is required when save_mode is set to sftp, https, cifs, or scp
- file_name: log file name and path If only the file name is specified, the log file is saved in the /home/plugin/ansible_ibmc/report/collect_SEL_log/ directory by default. In the directory, plugin indicates the actual user name.

```
[plugin@localhost examples]$ vi collect_sel_logs.yml
---
- hosts: myhosts
connection: local
```

```
name: collect sel logs
 gather_facts: False
# save mode: place to save logs: local, sftp, https, nfs, cifs, scp
# file_server_ip: ip address of file server, if save_mode is local, this parameter can be left blank.
# file_server_user: the user of file server
# file_server_pswd: the password of file server
# file_name: Log file storage path and file name
 tasks:
 - name: collect sel logs
  ibmc_collect_sel_logs:
   ibmc ip: "{{ ibmc ip }}"
   ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
   save mode: "local"
   file_server_ip: "sftp_server_ip"
   file_server_user: "{{ sftp_user }}"
   file_server_pswd: "{{ sftp_pswd }}"
   file_name: "/home/plugin/SEL_log.tar.gz"
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- Run the following command to collect SEL logs:

ansible-playbook collect_sel_logs.yml

The command execution is successful if the following information is displayed:

4.13.3 Clearing SELs

Procedure

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- Run the following command to clear SELs:

ansible-playbook clear_sel_logs.yml

The command execution is successful if the following information is displayed:

```
[plugin@localhost examples]$ ansible-playbook clear_sel_logs.yml /usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it. utils.CryptographyDeprecationWarning,
```

4.14 Common Interface

◯ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

A common public interface is provided. You can configure the URL and request body of the interface to implement all Redfish functions.

For details about Redfish, see the Server iBMC Redfish API Description.

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/common_api.yml file.

- url: request URL. The path must start with /redfish, for example, /redfish/v1/ EventService.
- request_method: request method. The value can be patch, post, get, or delete.
- **request_body**: request body. The request body is in JSON format. This parameter cannot be set when the **request_method** is set to **get** or **delete**.

```
[plugin@localhost examples]$ vi common_api.yml
- hosts: myhosts
connection: local
 name: common api
 gather_facts: False
# url: request resource
# request method: request method: GET, POST, PATCH, DELETE
# request_body: request body content
 tasks:
 - name: common api
  ibmc_common_api:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
   url: "/redfish/v1/EventService"
   request method: "GET"
   request_body: '{}'
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command:

ansible-playbook common_api.yml

The command execution is successful if the following information is displayed:

4.15 Local File Transfer

◯ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Upload files from a local end and download files to a local end.

4.15.1 Uploading Local Files

Parameter Configuration

Modify the **imgfile** parameter in the **/home/plugin/ibmc_ansible/examples/upload_file.yml** file.

imgfile: path and name of the local file to be uploaded. The format is *file path/file name*, for example, */home/plugin/SOO.keytab*. The uploaded file is saved in the */tmp/web* directory of the iBMC.

□ NOTE

For details about the file types supported by the server, see the API Description > Operation on Update Service Resources > Uploading a File in the Server iBMC Redfish API Description.

[plugin@localhost examples]\$ vi upload_file.yml

```
- hosts: myhosts
 connection: local
 name: file upload
 gather_facts: False
# imgfile: User-specified file to be uploaded, The format is file_path/file_name. After the upload is successful, the file is
placed in the /tmp/web on iBMC
      # The file types allowed by the V3 board are as follows:
{"hpm","cer","pem","cert","crt","pfx","p12","xml","keys","pub"}
# The file types allowed by the V5 board are as follows: {"hpm","zip","asc","cer","pem","cert","crt","pfx","p12","xml","keys","pub"," keytab"}
      # The maximum allowable hpm file of V3 single-board is 46M; the maximum allowable hpm, zip, asc file of v5
single-board is 90M.
      # The maximum allowable size of cer, pem, cert, crt, xml, p12, and keytab files is 1M.
      # The maximum allowable size of pfx and keys files is 2M, and the maximum allowable size of pub files is 2KB.
 tasks:
 - name: file upload
  ibmc_upload_file:
   ibmc_ip: "{{ ibmc_ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc pswd: "{{ ibmc pswd}}}"
   imgfile: "/home/plugin/SOO.keytab"
```

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to upload a local file:

ansible-playbook upload_file.yml

The command execution is successful if the following information is displayed:

4.15.2 Downloading a File to a Local Computer

Parameter Configuration

Modify the values of **file_name** and **local_path** in the **/home/plugin/ibmc_ansible/examples/download_file.yml** file.

• **file_name**: name of the file to be downloaded from the iBMC. The file must be in the /tmp/web directory or its subdirectory.

For example, if the file path is /tmp/web/111.txt, set this parameter to 111.txt. If the file path is /tmp/web/***/111.txt, set this parameter to /***/111.txt.

 local_path: local path for storing the file. If this parameter is not specified, the file is downloaded to the /home/plugin/ansible_ibmc/report/download/ directory by default.

◯ NOTE

- plugin: Use the actual user name.
- After the file is downloaded, the file name is automatically changed. The IP address and time are added to the beginning of the original file name.

```
[plugin@localhost examples]$ vi download_file.yml
---
- hosts: myhosts
connection: local
name: file download
gather_facts: False

# file_name: the name of the file to be downloaded, from /tmp/web of iBMC
# local_path: local path for storing files, The default file save path is /home/plugin/ansible_ibmc/report/download/

tasks:
- name: file download
ibmc_download_file:
ibmc_ip: "{{ ibmc_ip}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_pswd: "{{ ibmc_pswd}}"
file_name: "SOO.keytab"
local_path: "/home/plugin/"
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to download the file:

ansible-playbook download_file.yml

The command execution is successful if the following information is displayed:

4.16 Managing the HTTPS Server Root Certificate

□ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

You can import or delete the root certificate of the remote HTTPS server and import the revocation list of the root certificate of the remote HTTPS server.

∩ NOTE

Only the iBMC 3.01.12.20 or later versions support this function.

4.16.1 Importing the Root Certificate of a Remote HTTPS Server

Parameter Configuration

Modify the /home/plugin/ibmc ansible/examples/https ca import.yml file.

- certpath: the path of the root certificate of the remote HTTPS server. The local path, iBMC's tmp directory, and remote path (for example, sftp://user:password@ip/path) are supported. Currently, the supported transfer protocols include HTTPS, SCP, SFTP, CIFS, and NFS. The file name extension can be .crt, .cer, or .pem.
- certID: ID of the root certificate authenticated by the remote HTTPS server. The
 value is an integer that ranges from 5 to 8. Certificates 5 to 8 are used for
 remote file transfer over HTTPS.

NOTICE

If a certificate with the specified certificate ID has been imported, the new certificate will overwrite the old one. To prevent certificate conflicts, query the certificate ID before setting certID. For details, see **4.17 Querying Security Service Information (JSON File Generated)**.

- usage: certificate usage. The value is FileTransfer.
- import_location: the location of the certificate to be imported. The value can be tmp (/tmp directory of the iBMC), local (local executor), or sftp/https/nfs/ cifs/scp (remote file server).

NOTICE

Either CertID or usage must be set.

```
[plugin@localhost\ examples] \$\ vi\ https\_ca\_import.yml
```

- hosts: myhosts connection: local name: import https ca gather_facts: False

certpath: certificate to be imported (including the path and file name).

When the certificate is imported from a remote file server, the format is protocol://file_server_ip/folder/file_name # The file name extension must be in (".crt", ".cer", ".pem").

certID: ID of the root certificate used to authenticate the remote HTTPS server.

- Available values: [5, 6, 7, 8].

usage: certificate usage

```
# - Available values: "FileTransfer".
# import location: location of the certificate.
  # If the certificate is stored in the tmp directory of the BMC, the value is tmp.
  # If the certificate is stored in a local directory, the value is local.
  # If the certificate is stored on a remote file server, the value is the file server protocol.
  # - Available values: tmp, local, sftp, https, nfs, cifs, scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password
 tasks:
 - name: import https ca
  ibmc_https_ca_import:
   ibmc ip: "{{ ibmc ip }}"
   ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
   certpath: "/tmp/TestCA.crt"
   certID: 5
   # usage: "FileTransfer"
   import location: "tmp"
   # file_server_user: "{{sftp_user}}"
   # file_server_pswd: "{{sftp_pswd}}}"
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- Run the following command to import the root certificate of the remote HTTPS server:

ansible-playbook https_ca_import.yml

The command execution is successful if the following information is displayed:

4.16.2 Deleting the Root Certificate of a Remote HTTPS Server

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/delete_https_ca.yml file.

certID: ID of the root certificate authenticated by the remote HTTPS server. The value is an integer that ranges from 5 to 8. Certificates 5 to 8 are used for remote file transfer over HTTPS.

◯ NOTE

Before setting the certID, you can query the ID of the imported remote HTTPS server root certificate by referring to **4.17 Querying Security Service Information (JSON File Generated)**.

```
[plugin@localhost examples]$ vi delete_https_ca.yml
---
- hosts: myhosts
connection: local
name: delete https ca
gather_facts: False

# certID: ID of the root certificate used to authenticate the remote HTTPS server.
# - Available values: [5, 6, 7, 8].

tasks:
- name: delete https ca
ibmc_delete_https_ca:
ibmc_ip: "{{ ibmc_ip}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_pswd: "{{ ibmc_pswd}}"
certID: 6
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to delete the root certificate of the remote HTTPS server:

ansible-playbook delete_https_ca.yml

The command execution is successful if the following information is displayed:

4.16.3 Importing a CRL of a Remote HTTPS Server

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/https_crl_import.yml file.

certpath: the path of the CRL of the root certificate of the remote HTTPS server.
 The local path, iBMC's tmp directory, and remote path (for example, sftp://user:password@ip/path) are supported. Currently, the supported transfer

- protocols include HTTPS, SCP, SFTP, CIFS, and NFS. The file name extension must be .crl.
- certID: ID of the root certificate object that issues the CRL. The value must be
 the certID value of an array member in the RootCertificate object returned
 when the security service information is queried. For details about how to query
 security service information, see 4.17 Querying Security Service Information
 (JSON File Generated).
- usage: certificate usage. The value is FileTransfer.
- import_location: the location of the remote HTTPS server's root certificate CRL to be imported. The value can be tmp (/tmp directory of the iBMC), local (local executor), or sftp/https/nfs/cifs/scp (remote file server).

NOTICE

Either CertID or usage must be set.

```
[plugin@localhost examples]$ vi https crl import.yml
- hosts: myhosts
 connection: local
 name: import https crl
 gather_facts: False
# certpath: the crl file to be imported (including the path and file name).
  # When the certificate is imported from a remote file server, the format is protocol://file_server_ip/folder/file_name
  # The file name extension must be .crl.
# certID: ID of the root certificate used to authenticate the remote HTTPS server.
  # - Available values: [5, 6, 7, 8].
# usage: certificate usage
  # - Available values: "FileTransfer".
# import location: location of the crl.
  # If the crl file is stored in the tmp directory of the BMC, the value is tmp.
  # If the crl file is stored in a local directory, the value is local.
  # If the crl file is stored on a remote file server, the value is the file server protocol.
  # - Available values: tmp, sftp, https, nfs, cifs, scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password
 tasks:
 - name: import https crl
  ibmc_https_crl_import:
   ibmc ip: "{{ ibmc ip }}"
   ibmc_user: "{{ ibmc_user }}"
   ibmc_pswd: "{{ ibmc_pswd }}"
   certpath: "/tmp/TestCA.crl"
   certID: 5
   # usage: "FileTransfer"
   import_location: "tmp"
   # file server user: "{{sftp user}}"
   # file_server_pswd: "{{sftp_pswd}}'
```

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/p/ugin/ibmc_ansible/examples
- Run the following command to import the root certificate CRL of the remote HTTPS server:

ansible-playbook https_crl_import.yml

The command execution is successful if the following information is displayed:

4.17 Querying Security Service Information (JSON File Generated)

■ NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Query information about the security service supported by the server.

Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to query the security service information.

ansible-playbook get_security_service_information.yml

The command execution is successful if the following information is displayed:

[plugin@localhost examples]\$ ansible-playbook get_security_service_information.yml /usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it. utils.CryptographyDeprecationWarning,

The *IP address_*SecurityServiceInfo.json file is generated in the *I*home/plugin/ansible_ibmc/report/security_service/ directory and can be exported.

4.18 Enabling or Disabling HTTPS File Server Certificate Verification

NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Enable or disable HTTPS file server certificate verification.

◯ NOTE

Only the iBMC 3.01.12.20 or later versions support this function.

Parameter Configuration

Modify the /home/plugin/ibmc_ansible/examples/set_https_cert_verification.yml file.

verify_cmd: indicates whether to verify the certificate of the remote HTTPS file server. The options are **True** (enable HTTPS file server certificate verification) and **False** (disable HTTPS file server certificate verification).

```
[plugin@localhost examples]$ vi set_https_cert_verification.yml
---
- hosts: myhosts
connection: local
name: set https cert verification
gather_facts: False

# verify_cmd: Configure the switch for enabling or disabling certificate verification for the HTTPS remote file server.
# Available values: True/False/off/on/yes/no/1/0

tasks:
- name: set https_cert_verification
ibmc_set_https_cert_verification:
ibmc_ip: "{{ ibmc_ip}}"
ibme_user: "{{ ibmc_user}}"
ibme_user: "{{ ibmc_pswd}}"
verify_cmd: True
```

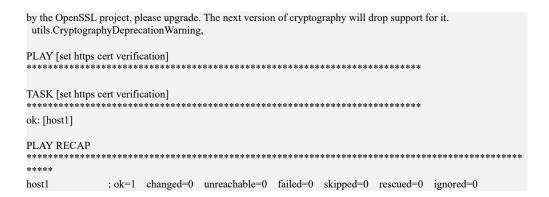
Commands

- Go to the /home/plugin/ibmc_ansible/examples directory.
 cd /home/plugin/ibmc_ansible/examples
- 2. Run the following command to enable or disable HTTPS file server certificate verification:

ansible-playbook set_https_cert_verification.yml

The command execution is successful if the following information is displayed:

[plugin@localhost examples]\$ ansible-playbook set_https_cert_verification.yml /usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported



4.19 Setting the Web Timeout Period

NOTE

The MM921 management module and CX320/CX621 switch modules do not support this function.

Function

Query and configure the web session timeout period.

4.19.1 Querying the Web Session Timeout Period (JSON File Generated)

Procedure

- Step 1 Go to the /home/plugin/ibmc_ansible/examples directory.

 cd /home/plugin/ibmc_ansible/examples
- **Step 2** Run the following command to query the web session timeout period.

ansible-playbook get_web_session_timeout.yml

The command execution is successful if the following information is displayed:

The JSON file (for example, 172.26.100.9_ web_session_timeout.json) generated after information query is saved in the /home/plugin/ansible_ibmc/report/ web_session_timeout directory by default. You are advised to export the JSON file and view the timeout period.

Step 3 Indicates the query result.

ansible-playbook -v get_web_session_timeout.yml

4.19.2 Configuring the Web Session Timeout Period

Parameter Configuration

----End

Modify the /home/plugin/ibmc_ansible/examples/set_web_session_timeout.yml file.

```
[plugin@localhost examples]$ vi set_web_session_timeout.yml
---
- hosts: myhosts
connection: local
name: set web session timeout
gather_facts: False

# web_session_timeout: The timeout of a web session in minutes, The value range is 5-480.

tasks:
- name: set web session timeout
ibmc_set_web_session_timeout:
ibmc_ip: "{{ ibmc_ip}}"
ibmc_user: "{{ ibmc_ip}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_pswd: "{{ ibmc_pswd}}"
web_session_timeout: 480
```

Procedure

Step 1 Go to the "/home/plugin/ibmc ansible/examples" directory.

cd /home/plugin/ibmc_ansible/examples

Step 2 Run the following command to configure the web session timeout period.

ansible-playbook set_web_session_timeout.yml

The command execution is successful if the following information is displayed:

----End



A.1 How Do I Encrypt Files and View, Edit, and Execute Encrypted Files

Before running the encryption command **ansible-vault**, you can run the **ansible-vault** -h command to view the help information about the command.

```
[root@localhost ~]# ansible-vault -h
Usage: ansible-vault [create|decrypt|edit|encrypt|encrypt_string|rekey|view] [options] [vaultfile.yml]
encryption/decryption utility for Ansible data files
Options:
 --ask-vault-pass ask for vault password
                show this help message and exit
 --new-vault-id=NEW_VAULT_ID
             the new vault identity to use for rekey
 --new-vault-password-file=NEW VAULT PASSWORD FILE
             new vault password file for rekey
 --vault-id=VAULT IDS the vault identity to use
 --vault-password-file=VAULT_PASSWORD_FILES
             vault password file
                 verbose mode (-vvv for more, -vvvv to enable
 -v, --verbose
             connection debugging)
 --version
                show program's version number, config file location,
             configured module search path, module location,
             executable location and exit
See 'ansible-vault <command> --help' for more information on a specific
command.
```

A.1.1 How Do I Encrypt Files

Step 1 Go to the /home/plugin/ibmc_ansible/examples directory.

cd /home/plugin/ibmc_ansible/examples

Step 2 Run the following command to encrypt a file:

ansible-vault encrypt File name

For example, to encrypt the **get_basic_info.yml** file:

Run the **ansible-vault encrypt get_basic_info.yml** command and enter the password as prompted.

[plugin@localhost examples]\$ ansible-vault encrypt get_basic_info.yml
New Vault password:

Step 3 Enter the password to be set and press **Enter**. The system prompts you to enter the password again.

```
[plugin@localhost examples]$ ansible-vault encrypt get_basic_info.yml
New Vault password:
Confirm New Vault password:
```

Step 4 Enter the password again and press **Enter**. The system displays a message indicating that the file is encrypted successfully.

```
[plugin@localhost examples]$ ansible-vault encrypt get_basic_info.yml
New Vault password:
Confirm New Vault password:
Encryption successful
```

----End

A.1.2 How Do I View Encrypted Files

Step 1 Go to the /home/plugin/ibmc_ansible/examples directory.

cd /home/plugin/ibmc_ansible/examples

Step 2 Run the following command to view a file:

ansible-vault view File name

For example, to view the encrypted **get_basic_info.yml** file:

Run the **ansible-vault view get_basic_info.yml** command and enter the file password as prompted.

```
[plugin@localhost examples]$ ansible-vault view get_basic_info.yml Vault password:
```

Step 3 Enter the file password and view the file content.

```
[plugin@localhost examples]$ ansible-vault view get_basic_info.yml

Vault password:
---
- hosts: myhosts
connection: local
name: get bmc basic info
gather_facts: False

# cvs_format: Whether to write the result to a CSV file. It is a mandatory parameter. Available values: True, False

tasks:
- name: get bmc basic info
ibmc_get_basic_info:
ibmc_ip: "{{ ibmc_ip}}"
ibmc_user: "{{ ibmc_user}}"
ibmc_pswd: "{{ ibmc_user}}"
csv_format: True
```

----End

A.1.3 How Do I Edit Encrypted Files

Step 1 Go to the /home/plugin/ibmc_ansible/examples directory.

cd /home/plugin/ibmc_ansible/examples

Step 2 Run the following command to edit a file:

ansible-vault edit File name

For example, to edit the encrypted **get_basic_info.yml** file:

Run the **ansible-vault edit get_basic_info.yml** command and enter the file password as prompted.

[plugin@localhost examples]\$ ansible-vault edit get_basic_info.yml Vault password:

Step 3 Enter the file password and edit the file content.

----End

A.1.4 How Do I Run Configuration Commands After Files Are Encrypted?

Step 1 Go to the /home/plugin/ibmc ansible/examples directory.

cd /home/p/ugin/ibmc_ansible/examples

Step 2 Run the configuration command.

ansible-playbook -vv File name --ask-vault-pass

For example, to query the basic information about a server:

Run the **ansible-playbook -vv get_basic_info.yml --ask-vault-pass** command and enter the file password as prompted. The following uses Python 3 as an example.

```
[plugin@localhost examples]$ ansible-playbook -vv get_basic_info.yml --ask-vault-pass
```

/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it.

utils.CryptographyDeprecationWarning,

ansible-playbook 2.9.9

config file = None

configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules'] ansible python module location = /usr/local/python3/lib/python3.7/site-packages/ansible-2.9.9-py3.7.egg/ansible executable location = /usr/local/python3/bin/ansible-playbook

python version = 3.7.5 (default, Nov 16 2020, 23:36:26) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

No config file found; using defaults

Vault password:

Step 3 Enter the file password to query the basic information about the server.

 $[plugin@localhost\ examples] \$\ ansible-playbook\ -vv\ get_basic_info.yml\ --ask-vault-pass$

/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it.

utils.CryptographyDeprecationWarning,

ansible-playbook 2.9.9

config file = None

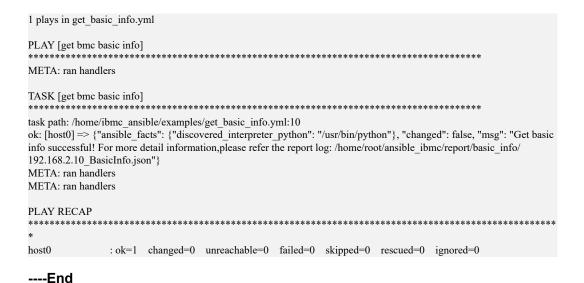
configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules'] ansible python module location = /usr/local/python3/lib/python3.7/site-packages/ansible-2.9.9-py3.7.egg/ansible executable location = /usr/local/python3/bin/ansible-playbook

python version = 3.7.5 (default, Nov 16 2020, 23:36:26) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]

No config file found; using defaults

Vault password:

PLAYBOOK: get_basic_info.yml



A.2 Message invalid upload file Is Displayed When the Root Certificate or CRL of the Local HTTPS Server Is Imported

Symptom

The root certificate or CRL of the local HTTPS server fails to be imported, and the message **invalid upload file** is displayed, as shown in the following figure:

```
[plugin@localhost examples]$ ansible-playbook -v https_ca_import.yml
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/
openssl/binding.py:177: CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by the
OpenSSL project, please upgrade. The next version of cryptography will drop support for it.
utils.CryptographyDeprecationWarning,
No config file found; using defaults
PLAY [import https ca]
TASK [import https ca]
fatal: [host0]: FAILED! => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"}, "changed": false,
"msg": "Import remote https server root ca failed! The detailed information is as follows: Send request to upload the file
failed! The error code is: 400, The error info is: {'error': {'code': 'Base.1.0.GeneralError', 'Message': 'A general error has
occurred. See ExtendedInfo for more information.', '@Message.ExtendedInfo': [{'@odata.type': '#MessageRegistry.
1.0.0.MessageRegistry', 'MessageId': 'iBMC.0.1.0.FirmwareUploadError', 'RelatedProperties': [], 'Message': 'An error
occurred during the firmware upload process. Details: invalid upload file.', 'MessageArgs': [], 'Severity': 'Warning',
'Resolution': 'Locate the cause based on error information, rectify the fault, and submit the request again.'}]}} "}
PLAY RECAP
          : ok=0 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
```

Solution

Step 1 Check whether the certificate or CRL file is correct.

- If yes, go to Step 2.
- If no, use the correct certificate or CRL file and run the import command again.
- **Step 2** Upgrade the iBMC of the server to the latest version and run the import command again.
 - ----End
 - **◯** NOTE

If the fault persists, contact technical support.

B Obtaining Technical Support

To obtain assistance, contact technical support as follows:

- Contact customer service center at support@xfusion.com.
- Contact technical support personnel.

C Communication Matrix

Sou rce Dev ice	Sour ce IP Addr ess	Sou rce Por t Nu mb er	Desti natio n Devic e	Destin ation IP Addres s	Desti natio n Port Numb er	Prot oco I	Port Description	Destin ation Port Config urable	Authent ication Mode	Encr yptio n Mod e
Devi ce to whic h Ansi ble belo ngs.	The IP addre ss of the Ansib le virtua I ether net port.	Ran	iBMC	IP address of the iBMC virtual network port veth.	22	SS H	When the file transfer service is conducted, enable the SSH standard protocol port. Ansbile, as a client, accesses the device on the iBMC.	Not involve d.	User name/ passwor d	SSH