|  |  |  |
| --- | --- | --- |
|  |  | |
|  | | |
|  | |  |
|  | | |
| Huawei Ansible Plug-in | |  |
| **User Guide** | |  |
|  | | |
| V2.0.6 | |  |
|  | | |
| Issue:01  Date:2021-06-30 | |  |
|  | | |
|  | Copyright © Copyright Holder. All rights reserved. | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  | | | |
|  | **Notice** |  | |
|  | 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 the warranty of any kind, express or implied. |  |

Contents

[About This Document 1](#_Toc256000000)

[1 Introduction 1-1](#_Toc256000001)

[2 Installing and Uninstalling the Huawei Ansible Plug-in 2-1](#_Toc256000002)

[2.1 Installing the Huawei Ansible Plug-in 2-1](#_Toc256000003)

[2.2 Uninstalling the Huawei Ansible Plug-in 2-2](#_Toc256000004)

[3 Configuring the Huawei Ansible Plug-in 3-1](#_Toc256000005)

[3.1 Configuring /etc/ansible/hosts 3-1](#_Toc256000006)

[3.2 Configuring the /group\_vars/myhosts File 3-2](#_Toc256000007)

[3.3 Configuring the SSL Certificate Authentication and TLS 1.2 Communication Mode 3-4](#_Toc256000008)

[4 Using the Huawei Ansible Plug-in 4-1](#_Toc256000009)

[4.1 Querying Basic Server Information 4-5](#_Toc256000010)

[4.2 Configuring the Boot Device 4-5](#_Toc256000011)

[4.2.1 Querying Boot Option Information 4-6](#_Toc256000012)

[4.2.2 Setting the Boot Device 4-6](#_Toc256000013)

[4.3 Managing the Power Supply 4-7](#_Toc256000014)

[4.3.1 Querying PSU Status 4-8](#_Toc256000015)

[4.3.2 Setting the Power Supply Status 4-8](#_Toc256000016)

[4.4 Managing iBMC Users 4-9](#_Toc256000017)

[4.4.1 Querying iBMC User Information (JSON File Generated) 4-9](#_Toc256000018)

[4.4.2 Creating an iBMC User 4-10](#_Toc256000019)

[4.4.3 Modify iBMC User Information 4-11](#_Toc256000020)

[4.4.4 Deleting an iBMC User 4-13](#_Toc256000021)

[4.5 Configuring iBMC Network Information 4-14](#_Toc256000022)

[4.5.1 Querying iBMC Network Configuration (JSON File Generated) 4-15](#_Toc256000023)

[4.5.2 Configuring iBMC Network Information 4-15](#_Toc256000024)

[4.6 Managing the NTP Service 4-17](#_Toc256000025)

[4.6.1 Querying NTP Service Information 4-17](#_Toc256000026)

[4.6.2 Configure NTP Settings 4-18](#_Toc256000027)

[4.7 Managing the SNMP Trap Service 4-19](#_Toc256000028)

[4.7.1 Querying SNMP Service Information (JSON File Generated) 4-19](#_Toc256000029)

[4.7.2 Configuring SNMP Trap 4-20](#_Toc256000030)

[4.8 Importing or Exporting a Profile 4-22](#_Toc256000031)

[4.8.1 Importing a Profile 4-22](#_Toc256000032)

[4.8.2 Exporting the Profile 4-24](#_Toc256000033)

[4.9 Upgrading Firmware 4-26](#_Toc256000034)

[4.9.1 Querying Firmware Version (JSON File Generated) 4-26](#_Toc256000035)

[4.9.2 Upgrading Firmware 4-27](#_Toc256000036)

[4.9.2.1 Out-of-Band Firmware Upgrade 4-27](#_Toc256000037)

[4.9.2.2 In-Band Firmware Upgrade 4-29](#_Toc256000038)

[4.10 Configuring RAID 4-31](#_Toc256000039)

[4.10.1 Querying RAID Configuration (JSON File Generated) 4-31](#_Toc256000040)

[4.10.2 Deleting a RAID Array 4-32](#_Toc256000041)

[4.10.3 Creating a RAID Array 4-33](#_Toc256000042)

[4.10.4 Modifying RAID Configuration 4-35](#_Toc256000043)

[4.11 Deploying the OS 4-37](#_Toc256000044)

[4.11.1 Deploying the OS Using ServiceCD 2.0 4-37](#_Toc256000045)

[4.11.2 Deploying the OS Using Smart Provisioning 4-41](#_Toc256000046)

[4.12 BIOS Management 4-48](#_Toc256000047)

[4.12.1 Querying BIOS Information (JSON File Generated) 4-48](#_Toc256000048)

[4.12.2 Setting BIOS Information 4-49](#_Toc256000049)

[4.12.3 Restoring Default BIOS Settings 4-51](#_Toc256000050)

[4.13 Log Management 4-52](#_Toc256000051)

[4.13.1 Collecting iBMC Logs in One-Click Mode 4-52](#_Toc256000052)

[4.13.2 Collecting SELs 4-54](#_Toc256000053)

[4.13.3 Clearing SELs 4-55](#_Toc256000054)

[4.14 Common Interface 4-56](#_Toc256000055)

[4.15 Local File Transfer 4-57](#_Toc256000056)

[4.15.1 Uploading Local Files 4-57](#_Toc256000057)

[4.15.2 Downloading a File to a Local Computer 4-59](#_Toc256000058)

[A FAQ 1](#_Toc256000059)

[B Obtaining Technical Support 1](#_Toc256000060)

About This Document

Purpose

This document describes how to install and uninstall the Huawei 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 Huawei 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** |
| --- | --- |
|  | Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury. |
|  | Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. |
|  | Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury. |
|  | 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. |
|  | 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 | 2020-12-07 | This issue is the first official release. |

# Introduction

Function

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

Supported Servers

| Architecture | Type | Model |
| --- | --- | --- |
| x86 | Rack server | RH2288H V3 |
| 2488 V5 |
| 2288H V5 |
| 1288H V6 |
| 2288H V6 |
| 5288 V6 |
| Blade server | CH121 V3 |
| CH242 V3 DDR4 |
| CH121 V5 |
| CH242 V5 |
| CH121L V5 |
| MM921 |
| CX621 |
| CX320 |
| High-density server | XH622 V3 |
| XH321 V5 |
| Heterogeneous server | G560 V5 |
| Arm | Rack server | TaiShan 200 server (model 2280) |
| TaiShan 200 server (model 2280, VE) |
| TaiShan 200K server (model 2280K) |
| TaiShan 200 server (model 5280) |
| TaiShan 200 server (model 5280, VE) |
| TaiShan 200K server (model 5280K) |
| TaiShan 200 server (model 5290) |
| TaiShan 200 server (model 2480) |
| TaiShan 200K server (model 2480K) |
| TaiShan 200 servers (model 2180) |
| TaiShan 200 server (model 5180) |
| TaiShan 200 Pro server (model 2480) |

Matching Versions

| Software | Matching Versions |
| --- | --- |
| iBMC | * x86 V5 server: V325 or later * x86 V3 server: V323 or later * Arm server:   + - V556 or later (for version number in the Vxxx format)     - 3.01.01.00 or later (for version number in the x.xx.xx.xx format) |
| BIOS | * x86 V5 server: V119 or later * x86 V3 server: V513 or later * Arm server: V112 or later |
| Smart Provisioning | * x86 server: V118 or later, which can be downloaded from [Smart Provisioning](https://support.huawei.com/enterprise/en/management-software/smart-provisioning-pid-23143793/software) * Arm server: V129 or later, which can be downloaded from [Smart Provisioning](https://support.huawei.com/enterprise/en/management-software/smart-provisioning-pid-23143793/software) |
| ServiceCD2.0 | V139 or later, which can be downloaded from [FusionServer Tools](https://support.huawei.com/enterprise/en/server/fusionserver-tools-pid-21015513/software) |

# Installing and Uninstalling the Huawei 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



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 Huawei Ansible Plug-in](#_EN-US_TOPIC_0102786743)

[2.2 Uninstalling the Huawei Ansible Plug-in](#_EN-US_TOPIC_0102786567)

## Installing the Huawei Ansible Plug-in

Obtain the installation package of the Huawei Ansible plug-in from [GitHub](https://github.com/Huawei/Server_Management_Plugin_Ansible/releases).

Log in to the Ansible server as the **root** user.

Upload the installation package to the **root** user directory on the Ansible server.

Go to the directory where the installation package of the Huawei Ansible plug-in is stored.

Run the following command to decompress the software package of the Huawei Ansible plug-in:

**unzip Huawei\_iBMC\_Ansible\_Module\_*x.x*.zip**

Run the following command to go to the **Huawei\_iBMC\_Ansible\_Module** directory generated after the decompression:

**cd Huawei\_iBMC\_Ansible\_Module**

Run the following command to install the Huawei Ansible plug-in:

**python install.py**



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 Huawei Ansible plug-in, and the **/home/ibmc\_ansible/examples** folder stores the YML sample file that needs to be configured for executing the Huawei Ansible plug-in.

----End

## Uninstalling the Huawei Ansible Plug-in

Log in to the Ansible server as the **root** user.

Go to the directory where the installation package of the Huawei Ansible plug-in is stored.

**cd Huawei\_iBMC\_Ansible\_Module/**

Run the uninstallation command.

**python uninstall.py**

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 Huawei\_iBMC\_Ansible\_Module]# python uninstall.py   
start uninstalling Huawei\_ibmc\_ansible module   
do you want to keep the yml files?(y/n)   
**n**

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 Huawei\_iBMC\_Ansible\_Module]# python uninstall.py   
start uninstalling Huawei\_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 Huawei\_iBMC\_Ansible\_Module]# python uninstall.py   
start uninstalling Huawei\_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 Huawei\_ibmc\_ansible successfully!**

----End

# Configuring the Huawei Ansible Plug-in



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](#_EN-US_TOPIC_0102786570)

[3.2 Configuring the /group\_vars/myhosts File](#_EN-US_TOPIC_0102786571)

[3.3 Configuring the SSL Certificate Authentication and TLS 1.2 Communication Mode](#_EN-US_TOPIC_0214681103)

## Configuring /etc/ansible/hosts

Run the following command to create the **/etc/ansible** directory:

**mkdir /etc/ansible**

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

**cd /etc/ansible**

Run the following command to create the **hosts** file:

**vi hosts**

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

[myhosts]   
**host0 ibmc\_ip=192.168.2.20 host=huaweiserver0**   
**host1 ibmc\_ip=192.168.2.21 host=huaweiserver1**

----End



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.

## Configuring the /group\_vars/myhosts File



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.



**/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)

Run the following command to switch to the **plugin** user:

**su plugin**

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/plugin/ibmc\_ansible/**

Run the following command to create the **/home/plugin/ibmc\_ansible/examples/group\_vars** directory:

**mkdir /home/plugin/ibmc\_ansible/examples/group\_vars**

Run the following command to go to the **/home/plugin/ibmc\_ansible/examples/group\_vars** directory:

**cd /home/plugin/ibmc\_ansible/examples/group\_vars**

Run the encryption command to create the **myhosts** file.

**ansible-vault create myhosts**



* 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.

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 huawei 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)

Run the following command to create the **/home/ibmc\_ansible/examples/group\_vars** directory:

**mkdir /home/ibmc\_ansible/examples/group\_vars**

Run the following command to go to the **/home/ibmc\_ansible/examples/group\_vars** directory:

**cd /home/ibmc\_ansible/examples/group\_vars**

Run the encryption command to create the **myhosts** file.

**ansible-vault create myhosts**



* 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.

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 huawei 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

## 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**.



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 **certifi** 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**



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.



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.   
 tasks:   
 - 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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command:

**ansible-playbook set\_request\_cfg.yml**



This command can be executed only by the **root** user.

The operation is successful if the following information is displayed:

[plugin@localhost examples]# ansible-playbook set\_request\_cfg.yml   
   
PLAY [set request config] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [set request config] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
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

# Using the Huawei Ansible Plug-in



* 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.
* 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**

[root@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 Huawei iBMC ansible modules version   
ibmc\_clear\_sel\_logs Clear SEL logs   
ibmc\_collect\_logs Collect iBMC logs   
ibmc\_collect\_sel\_logs Collect iBMC SEL logs   
ibmc\_common\_api Common api   
ibmc\_create\_account Create an ibmc user   
ibmc\_create\_raid Create volume   
ibmc\_delete\_account Delete an ibmc user   
ibmc\_delete\_raid Delete volume   
ibmc\_deploy\_os\_by\_service\_cd deploy os by service cd   
ibmc\_deploy\_os\_by\_sp deploy os by sp   
ibmc\_download\_file Download files   
ibmc\_get\_account Get ibmc user info   
ibmc\_get\_basic\_info Get server information   
ibmc\_get\_bios Get bios info   
ibmc\_get\_boot\_device get boot device   
ibmc\_get\_firmware\_info\_by\_sp get firmware info info   
ibmc\_get\_ip Get ibmc ip info   
ibmc\_get\_ntp Get ntp info   
ibmc\_get\_power\_status get ibmc accounts info   
ibmc\_get\_raid Get raid info   
ibmc\_get\_snmp\_trap Get snmp trap resource info   
ibmc\_inband\_fw\_update update inband firmware   
ibmc\_modify\_account modify an ibmc user   
ibmc\_modify\_raid Modify volume   
ibmc\_outband\_fw\_update update outband firmware   
ibmc\_profile\_export export the server profile   
ibmc\_profile\_import import the server profile   
ibmc\_reset\_bios Reset BIOS resource attributes   
ibmc\_set\_bios Set bios info   
ibmc\_set\_boot\_device Set boot device   
ibmc\_set\_ip Set ibmc ip info   
ibmc\_set\_ntp Set ntp info   
ibmc\_set\_power manager server power   
ibmc\_set\_redfish\_request\_cfg set request config   
ibmc\_set\_snmp\_trap Set snmp trap info   
ibmc\_upload\_file upload file

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

**ansible-doc *Command module name***

Example: **ansible-doc ibmc\_get\_account**

[root@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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to view the version:

**ansible-playbook -v show\_ibmc\_ansible\_version.yml**

[plugin@localhost examples]# ansible-playbook -v show\_ibmc\_ansible\_version.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 [show Huawei iBMC ansible modules version] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [show Huawei iBMC ansible modules version] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [127.0.0.1] => {"changed": false, "msg": "**Huawei iBMC ansible modules version is 2.0.6**"}   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
127.0.0.1 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[4.1 Querying Basic Server Information](#_EN-US_TOPIC_0102786712)

[4.2 Configuring the Boot Device](#_EN-US_TOPIC_0206987987)

[4.3 Managing the Power Supply](#_EN-US_TOPIC_0207371483)

[4.4 Managing iBMC Users](#_EN-US_TOPIC_0193957031)

[4.5 Configuring iBMC Network Information](#_EN-US_TOPIC_0193957036)

[4.6 Managing the NTP Service](#_EN-US_TOPIC_0193957037)

[4.7 Managing the SNMP Trap Service](#_EN-US_TOPIC_0193957039)

[4.8 Importing or Exporting a Profile](#_EN-US_TOPIC_0102786736)

[4.9 Upgrading Firmware](#_EN-US_TOPIC_0102786721)

[4.10 Configuring RAID](#_EN-US_TOPIC_0102786725)

[4.11 Deploying the OS](#_EN-US_TOPIC_0102786703)

[4.12 BIOS Management](#_EN-US_TOPIC_0000001167612627)

[4.13 Log Management](#_EN-US_TOPIC_0000001167612631)

[4.14 Common Interface](#_EN-US_TOPIC_0000001167772605)

[4.15 Local File Transfer](#_EN-US_TOPIC_0000001120895452)

## 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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

The file (for example, **172.26.100.9\_BasicInfo.json**) generated after the query is saved in the **/var/log/ansible/ibmc/report/basic\_info** directory by default. You are advised to export the file before viewing it.

## Configuring the Boot Device



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.

### Querying Boot Option Information

Procedure

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to query the configuration of boot options:

**ansible-playbook get\_boot\_device.yml**



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:

[plugin@localhost examples]# ansible-playbook get\_boot\_device.yml   
   
PLAY [get boot device] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get boot device] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

You can view the query result in the **ansibleibmc.report** file in the **/var/log/ansible/ibmc/report** directory, as shown in [3](#d0e1614).

1. Run the following command to view the information:

**cd /var/log/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']}}

### 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/plugin/ibmc\_ansible/examples**

1. Run the following command to configure the boot device.

**ansible-playbook set\_boot\_device.yml**

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

[plugin@localhost examples]# ansible-playbook set\_boot\_device.yml   
   
PLAY [set boot device] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [set boot device] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Managing the Power Supply



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

Function

Query and set the power supply status.

### Querying PSU Status

Procedure

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook get\_power\_status.yml   
   
PLAY [get ibmc os power status] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get ibmc os power status] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

The query result is in the **ansibleibmc.report** file in **/var/log/ansible/ibmc/report**.

1. Run the following command to view the information:

**cd /var/log/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

### 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\_user }}"   
 ibmc\_pswd: "{{ ibmc\_pswd }}"   
 **power\_cmd: "poweron"**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook set\_power.yml   
   
PLAY [power manager] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [power manager] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Managing iBMC Users

Function

TQuery, create, modify, or delete iBMC users.

### Querying iBMC User Information (JSON File Generated)

Procedure

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to query the iBMC user.

**ansible-playbook get\_account.yml**

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

[plugin@localhost examples]# ansible-playbook get\_account.yml   
   
PLAY [get ibmc Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get ibmc Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

The JSON file (for example, **172.26.100.9\_AccountInfo.json**) generated after the query is saved in the **/var/log/ansible/ibmc/report/account\_info** directory by default. You are advised to export the JSON file before viewing it.

### 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.

[root@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 huawei 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/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook create\_account.yml   
   
PLAY [create ibmc Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [create ibmc Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### 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/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 huawei 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 :   
 ibmc\_ip: "{{ ibmc\_ip }}"   
 ibmc\_user: "{{ ibmc\_user }}"   
 ibmc\_pswd: "{{ ibmc\_pswd }}"   
 old\_account\_user: "test"   
 new\_account\_user: "{{ account\_user }}"   
 new\_account\_pswd: "{{ account\_pswd }}"   
 **roleid: "Administrator"**   
 **locked: False**   
 **enable: True**   
 **login\_interface:**   
  **- Web**   
 **login\_rule:**   
 **- Rule1**   
 **account\_insecure\_prompt\_enabled: True**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook modify\_account.yml   
   
PLAY [modify ibmc Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [modify ibmc Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### 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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook delete\_account.yml   
   
PLAY [delete ibmc Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [delete ibmc Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Configuring iBMC Network Information



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.



* 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.

### Querying iBMC Network Configuration (JSON File Generated)

Procedure

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to query the iBMC network information.

**ansible-playbook get\_ibmc\_ip.yml**

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

[plugin@localhost examples]# ansible-playbook get\_ibmc\_ip.yml   
   
PLAY [get ibmc ip] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get ibmc ip] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

The JSON file (for example, **172.26.100.9\_iBMCIPInfo.json**) generated after the query is saved in the **/var/log/ansible/ibmc/report/ibmc\_ip** directory by default. You are advised to export the JSON file before viewing it.

### Configuring iBMC Network Information



The Ansible plug-in supports only IPv4 connections to servers. Do not to disable the IPv4 mode.

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   
   
# 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.   
   
 tasks:   
 - name: set ibmc ip   
 ibmc\_set\_ip:   
 ibmc\_ip: "{{ ibmc\_ip }}"   
 ibmc\_user: "{{ ibmc\_user }}"   
 ibmc\_pswd: "{{ ibmc\_pswd }}"   
 **ip\_version: "IPv4AndIPv6"**   
 **ipv4\_addr:**   
 **- address: "172.26.100.100"**   
  **subnet\_mask: "255.255.0.0"**   
 **gateway: "172.26.0.1"**   
 **address\_origin: "Static"**   
 **ipv6\_addr:**   
  **- address: "fc00:172::100"**   
  **prefix\_length: 10**   
 **address\_origin: "Static"**   
 **ipv6\_gateway: "fc00:172::1"**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook set\_ibmc\_ip.yml   
   
PLAY [set ibmc ip] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [set ibmc ip] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## 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.

### Querying NTP Service Information

Procedure

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to query the NTP service information.

**ansible-playbook get\_ntp.yml**

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

[plugin@localhost examples]# ansible-playbook get\_ntp.yml   
   
PLAY [get ntp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get ntp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

The query result is in the **ansibleibmc.report** file in **/var/log/ansible/ibmc/report**.

1. Run the following command to view the information:

**cd /var/log/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}

### 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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to configure the NTP:

**ansible-playbook set\_ntp.yml**

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

[plugin@localhost examples]# ansible-playbook set\_ntp.yml   
   
PLAY [set ntp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [set ntp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Managing the SNMP Trap Service



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.

### Querying SNMP Service Information (JSON File Generated)

Procedure

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to query the SNMP service information.

**ansible-playbook get\_snmp\_trap.yml**

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

[plugin@localhost examples]# ansible-playbook get\_snmp\_trap.yml   
   
PLAY [get snmp trap] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get snmp trap] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

The JSON file (for example, **172.26.100.9\_SNMPTrapInfo.json**) generated after the query is saved in the **/var/log/ansible/ibmc/report/snmp\_trap** directory by default. You are advised to export the JSON file before viewing it.

### 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.   
   
 tasks:   
 - 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.



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 huawei 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"

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook set\_snmp\_trap.yml   
   
PLAY [set snmp trap] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [set snmp trap] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## 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.

### 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.
* Folder in the directory of the remote file server. The format of the file path is *File transfer protocol***://***IP 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.

**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   
   
 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\_import: "sftp://192.168.1.1/data/"**   
 **file\_server\_user: "{{sftp\_user}}"**   
 **file\_server\_pswd: "{{sftp\_pswd}}"**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to import the profile:

**ansible-playbook profile\_import.yml**

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

[plugin@localhost examples]# ansible-playbook profile\_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,   
   
PLAY [import profile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [import profile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### 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***://***IP 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   
   
 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\_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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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 by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it.   
 utils.CryptographyDeprecationWarning,   
   
PLAY [export profile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [export profile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Upgrading Firmware

Function

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

### Querying Firmware Version (JSON File Generated)



* 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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to query the firmware version.

**ansible-playbook get\_firmware\_info\_by\_sp.yml**

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

[plugin@localhost examples]# ansible-playbook get\_firmware\_info\_by\_sp.yml   
   
PLAY [get firmware info by sp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get firmware info by sp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

The JSON file (for example, **172.26.100.9\_fwInfo.json**) generated after the query is saved in the **/var/log/ansible/ibmc/report/inband\_fw\_info** directory by default. You are advised to export the JSON file before viewing it.

### Upgrading Firmware

#### 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***://***IP 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 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.



* 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:
* Visit [**Support > Intelligent Servers**](https://support.huawei.com/enterprise/en/category/intelligent-servers-pid-1548148142425?submodel=15791).

Click a server model.

Click the **Software Download** tab.

Select the patch version.

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   
   
 tasks:   
 - 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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the upgrade command:

**ansible-playbook update\_outband\_fw.yml**

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

[plugin@localhost examples]# ansible-playbook update\_outband\_fw.yml   
   
PLAY [update outband fw] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [update outband fw] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

#### In-Band Firmware Upgrade



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 following documents:

* x86 server: [Smart Provisioning User Guide](https://support.huawei.com/enterprise/en/doc/EDOC1100034052)
* Arm server: [Smart Provisioning User Guide (aarch64)](https://support.huawei.com/enterprise/en/doc/EDOC1100108318)

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](https://support.huawei.com/enterprise/en/management-software/fusionserver-idriver-pid-21588909/software).

[plugin@localhost examples]# vi update\_inband\_fw.yml   
---   
- hosts: myhosts   
 connection: local   
 name: update inband fw   
 gather\_facts: False   
   
 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 huawei 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"

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the upgrade command:

**ansible-playbook update\_inband\_fw.yml**

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

[plugin@localhost examples]# ansible-playbook update\_inband\_fw.yml   
   
PLAY [update inband fw] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [update inband fw] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Configuring RAID



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.
* Query, configure, modify, and delete the information of the LSI SAS3108, Avago SAS3408iMR, Avago SAS3004iMR, and Avago SAS3508 RAID controller cards.

### Querying RAID Configuration (JSON File Generated)

Procedure

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to query the RAID configuration:

**ansible-playbook get\_raid.yml**

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

[plugin@localhost examples]# ansible-playbook get\_raid.yml   
   
PLAY [get raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

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

### 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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to delete the RAID:

**ansible-playbook delete\_raid.yml**

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

[plugin@localhost examples]# ansible-playbook delete\_raid.yml   
   
PLAY [delete raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [delete raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### Creating a RAID Array

Parameter Configuration

Modify the **/home/plugin/ibmc\_ansible/examples/create\_raid.yml** file.



Different RAID controller cards support different configurable parameters. For details, see the [*Huawei V2 and V3 Server RAID Controller Card User Guide*](https://support.huawei.com/enterprise/en/doc/EDOC1000004186), [*Huawei V5 Server RAID Controller Card User Guide*](http://support.huawei.com/enterprise/en/doc/EDOC1000163569), or [*TaiShan Server RAID Controller Card User Guide*](https://support.huawei.com/enterprise/en/doc/EDOC1100048773).

[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"**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to create a RAID:

**ansible-playbook create\_raid.yml**

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

[plugin@localhost examples]# ansible-playbook create\_raid.yml   
   
PLAY [create raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [create raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### Modifying RAID Configuration

Parameter Configuration

Modify the **/home/plugin/ibmc\_ansible/examples/modify\_raid.yml** file.



Different RAID controller cards support different parameters. For details, see the [*Huawei V2 and V3 Server RAID Controller Card User Guide*](https://support.huawei.com/enterprise/en/doc/EDOC1000004186), [*Huawei V5 Server RAID Controller Card User Guide*](http://support.huawei.com/enterprise/en/doc/EDOC1000163569), or [*TaiShan Server RAID Controller Card User Guide*](https://support.huawei.com/enterprise/en/doc/EDOC1100048773).

[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, 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   
# 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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook modify\_raid.yml   
   
PLAY [modify raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [modify raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## 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.

### Deploying the OS Using ServiceCD 2.0



* This method applies to x86 V2 and V3 servers.
* The ServiceCD2.0 mode depends on the ServiceCD2.0 image package. To obtain the ServiceCD2.0 image package, visit [FusionServer Tools](https://support.huawei.com/enterprise/en/management-software/fusionserver-tools-pid-21015513/software/).

Parameter Configuration

Modify the **/home/plugin/ibmc\_ansible/examples/deploy\_os\_by\_service\_cd.yml** file.

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

[plugin@localhost examples]# vi deploy\_os\_by\_service\_cd.yml   
   
---   
- hosts: myhosts   
 connection: local   
 name: deploy os by service cd   
 gather\_facts: False   
#os\_img: The os image path ; Format: protocol://Username:Password@IPaddress/Folder/image\_file; Available protocol: nfs,cifs,https   
#service\_cd\_img:The service cd image path; Format: protocol://Username:Password@IPaddress/Folder/image\_file; Available protocol: nfs,cifs,https   
#os\_Type: os type ; Available values:   
 # CentOS6U7\_x64, CentOS6U8\_x64, CentOS6U9\_x64,CentOS7U0\_x64, CentOS7U1\_x64, CentOS7U2\_x64, CentOS7U3\_x64, CentOS7U4\_x64, CentOS7U5\_x64   
 # RHEL6U7\_x64, RHEL6U8\_x64, RHEL6U9\_x64, RHEL7U0\_x64, RHEL7U1\_x64, RHEL7U2\_x64, RHEL7U3\_x64, RHEL7U4\_x64, RHEL7U5\_x64,   
 # SLES11SP4\_x64, SLES11SP3\_x64, SLES12\_x64, SLES12SP1\_x64, SLES12SP2\_x64, SLES12SP3\_x64,   
 # Ubuntu16.04\_x64, Ubuntu14.04\_x64   
 # ESXi5.5\_x64, ESXi6.0\_x64, ESXi6.5\_x64, ESXi6.7\_x64   
 # Win2008\_R2\_x64, Win2012\_x64, Win2012\_R2\_x64, Win2016\_x64,   
#cd\_key: key of the OS Installation   
#password: Available values: Please refer to the installation guide of the OS for more infomation. Tips: use a strong password otherwise may failed to install os   
#timezone: Available values: windows Format: (GMT-12:00) International Date Line West; centos,redhat,ubuntu Format:Africa/Abidjan;   
 # SLES Format: Africa/Abidjan ; Please refer to the installation guide of the OS for more infomation.   
#language: Available values: windows Format: en-US; centos,redhat,ubuntu Format:en\_US.UTF-8; SlES Format:en\_US   
 # Please refer to the installation guide of the OS for more infomation.   
#hostname: Host Name   
#owner\_name: Owner Name   
#org\_name: Organize Name   
#position: Position where the os install; Available values: disk, usb   
#partitions: Partition information; Available values: Please refer to the installation guide of the OS.   
#mode: mode to install Available values: 1, 2, 3, ;1 for standard, 2 for full , 3 for Customized   
#rpms: rpm packages you want to install.   
#script: install script   
#software: software you want to install; Available values: ibma   
#win\_os\_name: windows os name, only for windows. Available values:   
 # Windows Server 2016 ServerStandard, Windows Server 2016 ServerStandardCore, Windows Server 2016 ServerDataCenter, Windows Server 2016 ServerDataCenterCore   
 # Windows Server 2012 R2 ServerStandard, Windows Server 2012 R2 ServerStandardCore, Windows Server 2012 R2 ServerDataCenter, Windows Server 2012 R2 ServerDataCenterCore   
 # Windows Server 2012 ServerStandard, Windows Server 2012 ServerStandardCore, Windows Server 2012 ServerDataCenter,Windows Server 2012 ServerDataCenterCore   
 # Windows Server 2008 R2 ServerStandard, Windows Server 2008 R2 ServerStandardCore, Windows Server 2008 R2 ServerEnterprise,   
 # Windows Server 2008 R2 ServerEnterpriseCore, Windows Server 2008 R2 ServerDataCenter, Windows Server 2008 R2 ServerDataCenterCore   
 # Windows Server 2008 R2 ServerWeb,Windows Server 2008 R2 ServerWebCore   
 tasks:   
 - name: deploy os by service cd   
 ibmc\_deploy\_os\_by\_service\_cd:   
 ibmc\_ip: "{{ ibmc\_ip }}"   
 ibmc\_user: "{{ ibmc\_user }}"   
 ibmc\_pswd: "{{ ibmc\_pswd }}"   
 **service\_cd\_img: "nfs://172.26.200.11/data/serviceCD.iso"**   
 **os\_img: "nfs://172.26.200.11/data/CentOS-7.3-x86\_64-DVD-1611.iso"**   
 **os\_type: "CentOS7U3\_x64"**   
 **win\_os\_name:**   
 **cd\_key:**   
 **password: "{{ os\_pswd }}"**   
 **hostname:**   
 **owner\_name:**   
 **language: "en\_US.UTF-8"**   
 **org\_name:**   
 **position: "disk"**   
 **partitions:**   
  **- partition: "swap:swap:10000|/:ext3:1"**   
 **timezone: "America/New\_York"**   
 **mode:**   
 **rpms:**   
 **- rpm:**   
  **script:**   
 **software: "ibma"**

Parameters

| Parameter | Description | Value |
| --- | --- | --- |
| service\_cd\_img | ServiceCD 2.0 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. |
| os\_img | OS image path (mandatory). |
| os\_type | Type of the OS to be installed (mandatory). | Value:  CentOS6U7\_x64, CentOS6U8\_x64, CentOS6U9\_x64, CentOS7U0\_x64, CentOS7U1\_x64, CentOS7U2\_x64, CentOS7U3\_x64, and CentOS7U4\_x64, CentOS7U5\_x64  RHEL6U7\_x64, RHEL6U8\_x64, RHEL6U9\_x64, RHEL7U0\_x64, RHEL7U1\_x64, RHEL7U2\_x64, RHEL7U3\_x64, RHEL7U4\_x64, RHEL7U5\_x64  SLES11SP4\_x64, SLES11SP3\_x64, SLES12\_x64, SLES12SP1\_x64, SLES12SP2\_x64, SLES12SP3\_x64  Ubuntu16.04\_x64, Ubuntu14.04\_x64  ESXi5.5\_x64, ESXi6.0\_x64, ESXi6.5\_x64, ESXi6.7\_x64  Win2008\_R2\_x64, Win2012\_x64, Win2012\_R2\_x64, Win2016\_x64 |
| win\_os\_name | Windows system name (mandatory). | Only the Windows operating system is supported.  Value:  Windows Server 2016 Server Standard, Windows Server 2016 Server StandardCore, Windows Server 2016 Server DataCenter, Windows Server 2016 Server DataCenter Core  Windows Server 2012 R2 Server Standard, Windows Server 2012 R2 Server StandardCore, Windows Server 2012 R2 Server DataCenter, and Windows Server 2012 R2 Server DataCenter Core  Windows Server 2012 Server Standard, Windows Server 2012 Server StandardCore, Windows Server 2012 Server DataCenter, Windows Server 2012 Server DataCenter Core  Windows Server 2008 R2 Server Standard, Windows Server 2008 R2 Server StandardCore, Windows Server 2008 R2 Server Enterprise, Windows Server 2008 R2 Server Enterprise Core, Windows Server 2008 R2 Server DataCenter, Windows Server 2008 R2 ServerData CenterCore, Windows Server 2008 R2 Server Web, windows Server 2008 R2 Server Web Core  Other systems are empty. |
| cd\_key | OS installation key (optional). | Windows and VMware: optional |
| password | Initial password of the administrator (mandatory). | It is specified by the **os\_pswd** parameter in the **myhosts** file. |
| hostname | Host name (optional). | - |
| owner\_name | Owner name (optional). | - |
| language | System language (optional). | For details, see the installation guide of the corresponding system.   * Linux: mandatory (for example, RHEL/CentOS/Ubuntu:**en\_US.UTF-8**; SLES: **en\_US**). * Windows: mandatory (for example, **en-US**). * For VMware OS, leave it blank. |
| org\_name | Organization name (optional). | - |
| position | Location where the OS is installed (optional). | * disk * usb |
| partitions | System partition settings (optional). | For details, see the installation guide of the corresponding system.   * Linux: mandatory (for example, **swap:swap:10000|/:ext3:1**). * Windows: mandatory (for example, **C:NTFS:50000|D:NTFS:1**) * VMware: This parameter is left empty. |
| timezone | System time zone (optional). | For details, see the installation guide of the corresponding system.   * Linux: mandatory (for example, **America/New\_York**) * Windows: mandatory (for example, **(GMT-12:00) International Date Line West**). * VMware: This parameter is left empty. |
| mode | System installation mode (optional). | * 1 * 2 * 3   **1**: Standard  **2**: full  **3**: Customized |
| rpms | RPM package to be installed (optional). | - |
| script | Installation script (optional). | - |
| software | System software to be installed (optional). | ibma |

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following deployment commands:

**ansible-playbook deploy\_os\_by\_service\_cd.yml**

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

[plugin@localhost examples]# ansible-playbook deploy\_os\_by\_service\_cd.yml   
   
PLAY [deploy os by service cd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [deploy os by service cd] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### Deploying the OS Using Smart Provisioning

For details about the servers supported by Smart Provisioning, see the following documents:

* x86 server: [Smart Provisioning User Guide](https://support.huawei.com/enterprise/en/doc/EDOC1100034052)
* Arm server: [Smart Provisioning User Guide (aarch64)](https://support.huawei.com/enterprise/en/doc/EDOC1100108318)



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 4-2.

[plugin@localhost examples]# vi deploy\_centos7u3\_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   
# EulerOSV2SP3   
#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 characters.   
# (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"**

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 Standard Core, 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, EulerOSV2SP3 |
| BootType | BIOS boot mode (optional). | * UEFIBoot * LegacyBoot * SecureBoot |
| 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 #, $, or spaces. * VMware OS: The parameter value must contain at least seven digits. For VMware ESXi 6.7, the parameter value must contain at least three types of characters, including letters, digits, and special characters and cannot exceed 40 digits.   NOTE  Smart Provisioning V119 and later versions support the following special characters: #$. |
| 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 |
| 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, **/**, **/home**, 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 |
| 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:

| OSType | Language | TimeZone | Keyboard |
| --- | --- | --- | --- |
| RHEL/CentOS/EulerOS/Ubuntu | en\_US.UTF-8 | America/New\_York | us |
| SLES | en\_US | America/New\_York | english-us |
| Windows | en-US | Eastern Standard Time | 0x00000409 |

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following deployment commands:

**ansible-playbook deploy\_centos7u3\_by\_sp.yml**

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

[plugin@localhost examples]# ansible-playbook deploy\_centos7u3\_by\_sp.yml   
   
PLAY [ibmc deploy centos7u3 by sp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [ibmc deploy centos7u3 by sp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host1.domain.com]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host1.domain.com : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## BIOS Management

Function

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

### 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/***user***/ansible\_ibmc/report/bios/** folder.



* *IP address***\_BIOSInfo.json**: Set the *IP address* to the configured server IP address.
* **/home/*user*/ansible\_ibmc/report/bios/**: Replace *user* 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\_pswd: "{{ ibmc\_pswd }}"   
 **bios\_attribute:**   
 **- QuickBoot**   
 **- QuietBoot**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to query the BIOS information.

**ansible-playbook get\_bios.yml**

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

[plugin@localhost examples]# ansible-playbook get\_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 [get bios] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get bios] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### 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.



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

* **bios\_attribute**: BIOS parameters to be set. This parameter is mandatory.



The following example describes how to set **QuickBoot** and **QuietBoot**. For details about configurable BIOS parameters, see the [Huawei Server Purley Platform BIOS Parameter Reference](https://support.huawei.com/enterprise/en/doc/EDOC1000163372/426cffd9?idPath=23710424|251364409|21782478|21941608) or [BIOS Parameter Reference (Kunpeng 920 Processor)](https://support.huawei.com/enterprise/en/doc/EDOC1100088647/426cffd9?idPath=23710424|251364417|9856629|250783947).

[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

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook set\_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 [set ibmc bios] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [set ibmc bios] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### 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.



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\_pswd: "{{ ibmc\_pswd }}"   
 **Immediately: False**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [reset ibmc bios] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Log Management

Function

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

### Collecting iBMC Logs in One-Click Mode

Parameter Configuration

Modify the **/home/plugin/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/***user***/ansible\_ibmc/report/collect\_IBMC\_log/** directory by default. In the directory, *user* 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"**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook collect\_ibmc\_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,   
   
PLAY [collect logs] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [collect logs] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### Collecting SELs

Parameter Configuration

Modify the **/home/plugin/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/***user***/ansible\_ibmc/report/collect\_SEL\_log/** directory by default. In the directory, *user* 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"**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook collect\_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,   
   
PLAY [collect sel logs] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [collect sel logs] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### Clearing SELs

Procedure

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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,   
   
PLAY [clear sel logs] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [clear sel logs] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Common Interface

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 [Huawei Server iBMC Redfish API Description](https://support.huawei.com/enterprise/en/doc/EDOC1000126992/6974c567?idPath=23710424|251364409|21782478|4275304).

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/Chassis/1/Oem/Huawei/Actions/Chassis.ControlIndicatorLED**.
* **request\_method**: request method. The value can be **patch**, **post**, **get**, or **delete**.
* **request\_body**: request body. The request body is in JSON format, for example, **'{"IndicatorLED":"Blinking", "Duration":50}'**. 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/Chassis/1/Oem/Huawei/Actions/Chassis.ControlIndicatorLED"**   
 **request\_method: "POST"**   
 **request\_body: '{"IndicatorLED":"Blinking", "Duration":50}'**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command:

**ansible-playbook common\_api.yml**

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

[plugin@localhost examples]# ansible-playbook common\_api.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 [common api] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [common api] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

## Local File Transfer

Function

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

### 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.

* x86 V3 servers support the following file types:

**.hpm**, **.cer**, **.pem**, **.cert**, **.crt**, **.pfx**, **.p12**, **.xml**, **.keys**, and **.pub**

* x86 V5 servers support the following file types:

**.hpm**, **.zip**, **.asc**, **.cer**, **.pem**, **.cert**, **.crt**, **.pfx**, **.p12**, **.xml**, **.keys**, **.pub**, and **.keytab**

* The maximum size of the **.hpm** file supported by an x86 V3 server is 46 MB. The maximum size of an **.hpm**, **.zip**, or **.asc** file supported by an x86 V5 server is 90 MB.
* The supported maximum size of the **.cer**, **.pem**, **.cert**, **.crt**, **.xml**, **.p12**, or **.keytab** file is 1 MB. The supported maximum size of a **.pfx** or **.keys** file is 2 MB, and the supported maximum size of a **.pub** file is 2 KB.

[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"**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. 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:

[plugin@localhost examples]# ansible-playbook upload\_file.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 [file upload] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [file upload] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

### 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/user/ansible\_ibmc/report/download/** directory by default.



* *user*: 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/USER/ansible\_ibmc/report/download/   
   
 tasks:   
 - name: file download   
 ibmc\_download\_file :   
 ibmc\_ip: "{{ ibmc\_ip }}"   
 ibmc\_user: "{{ ibmc\_user }}"   
 ibmc\_pswd: "{{ ibmc\_pswd }}"   
 **file\_name: "SOO.keytab"**   
 **local\_path: "/home/plugin/"**

Commands

1. Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

1. Run the following command to download the file:

**ansible-playbook download\_file.yml**

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

[plugin@localhost examples]# ansible-playbook download\_file.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 [file download] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [file download] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

1. FAQ

* 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   
 -h, --help 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   
 -v, --verbose verbose mode (-vvv for more, -vvvv to enable   
 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.

* + 1. How Do I Encrypt Files

Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

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:

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:

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

* + 1. How Do I View Encrypted Files

Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

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:

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\_pswd }}"   
 csv\_format: True

----End

* + 1. How Do I Edit Encrypted Files

Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

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:

Enter the file password and edit the file content.

----End

* + 1. How Do I Run Configuration Commands After Files Are Encrypted?

Go to the **/home/plugin/ibmc\_ansible/examples** directory.

**cd /home/plugin/ibmc\_ansible/examples**

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:

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 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
1 plays in get\_basic\_info.yml   
   
PLAY [get bmc basic info] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
META: ran handlers   
   
TASK [get bmc basic info] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
task path: /home/ibmc\_ansible/examples/get\_basic\_info.yml:10   
ok: [host0] => {"ansible\_facts": {"discovered\_interpreter\_python": "/usr/bin/python"}, "changed": false, "msg": "Get basic info successful! For more detail information,please refer the report log: /home/root/ansible\_ibmc/report/basic\_info/192.168.2.10\_BasicInfo.json"}   
META: ran handlers   
META: ran handlers   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

----End

1. Obtaining Technical Support

If a fault is difficult to locate and rectify after you refer to documents, contact technical support as follows:

* Contact customer service center.
* Enterprise customers in China can contact us in the following ways:
  1. Hotline: 400-822-9999
  2. Email: [support\_e@huawei.com](mailto:support_e@huawei.com)
* Enterprise customers outside China can obtain the customer service information from: [Global Service Hotline](https://e.huawei.com/en/service-hotline-query).
* Carrier customers in China can contact us in the following ways:
  1. Hotline: 400-830-2118
  2. Email: [support@huawei.com](mailto:support@huawei.com)
* Carrier customers outside China can obtain the customer service information from: [Global TAC Information](https://support.huawei.com/carrier/docview!docview?nid=IN0000034614).
* Contact the local technical support personnel.