|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
| **Huawei Ansible Plug-in**  **V2.0.5** | | |
| **User Guide** | | |
| **Issue** | **02** | |
| **Date** | **2020-11-23** | |
|  | | | | |
|  | HUAWEI TECHNOLOGIES CO., LTD. | |  |  |

|  |
| --- |
| **Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.**  No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.  **Trademarks and Permissions**  and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.  All other trademarks and trade names mentioned in this document are the property of their respective holders.  **Notice**  The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.  The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied. |

|  |  |
| --- | --- |
| Huawei Technologies Co., Ltd. | |
| Address: | Huawei Industrial Base  Bantian, Longgang  Shenzhen 518129  People's Republic of China |
| Website: | <https://e.huawei.com> |
|  |  |

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 |
| --- | --- | --- |
| 02 | 2020-11-23 | This issue is the second official release.  Added A FAQ. |
| 01 | 2020-07-30 | This issue is the first official release. |

Contents

[About This Document ii](#_Toc57025318)

[1 Introduction 1](#_Toc57025319)

[2 Installing and Uninstalling the Huawei Ansible Plug-in 3](#_Toc57025320)

[2.1 Installing the Huawei Ansible Plug-in 3](#_Toc57025321)

[2.2 Uninstalling the Huawei Ansible Plug-in 4](#_Toc57025322)

[3 Configuring the Huawei Ansible Plug-in 5](#_Toc57025323)

[3.1 Configuring /etc/ansible/hosts 5](#_Toc57025324)

[3.2 Configuring /group\_vars/myhosts 6](#_Toc57025325)

[3.3 Configuring the SSL Certificate Authentication and TLS 1.2 Communication Mode 7](#_Toc57025326)

[4 Using the Huawei Ansible Plug-in 9](#_Toc57025327)

[4.1 Querying Basic Server Information 11](#_Toc57025328)

[4.2 Configuring the Boot Device 12](#_Toc57025329)

[4.2.1 Querying Boot Option Information 12](#_Toc57025330)

[4.2.2 Setting the Boot Device 13](#_Toc57025331)

[4.3 Managing the Power Supply 14](#_Toc57025332)

[4.3.1 Querying PSU Status 14](#_Toc57025333)

[4.3.2 Setting the Power Supply Status 15](#_Toc57025334)

[4.4 Managing iBMC Users 16](#_Toc57025335)

[4.4.1 Querying iBMC User Information (JSON File Generated) 16](#_Toc57025336)

[4.4.2 Creating an iBMC User 16](#_Toc57025337)

[4.4.3 Modify iBMC User Information 18](#_Toc57025338)

[4.4.4 Deleting an iBMC User 20](#_Toc57025339)

[4.5 Configuring iBMC Network Information 21](#_Toc57025340)

[4.5.1 Querying iBMC Network Configuration (JSON File Generated) 21](#_Toc57025341)

[4.5.2 Configuring iBMC Network Information 21](#_Toc57025342)

[4.6 Managing the NTP Service 23](#_Toc57025343)

[4.6.1 Querying NTP Service Information 23](#_Toc57025344)

[4.6.2 Configure NTP Settings 24](#_Toc57025345)

[4.7 Managing the SNMP Trap Service 25](#_Toc57025346)

[4.7.1 Querying SNMP Service Information (JSON File Generated) 25](#_Toc57025347)

[4.7.2 Configuring SNMP Trap 26](#_Toc57025348)

[4.8 Importing or Exporting a Profile 28](#_Toc57025349)

[4.9 Upgrading Firmware 29](#_Toc57025350)

[4.9.1 Querying Firmware Version (JSON File Generated) 29](#_Toc57025351)

[4.9.2 Upgrading Firmware 30](#_Toc57025352)

[4.9.2.1 Out-of-Band Firmware Upgrade 30](#_Toc57025353)

[4.9.2.2 In-Band Firmware Upgrade 31](#_Toc57025354)

[4.10 Configuring RAID 33](#_Toc57025355)

[4.10.1 Querying RAID Configuration (JSON File Generated) 34](#_Toc57025356)

[4.10.2 Deleting a RAID Array 34](#_Toc57025357)

[4.10.3 Creating a RAID Array 35](#_Toc57025358)

[4.10.4 Modifying RAID Configuration 37](#_Toc57025359)

[4.11 Deploying the OS 39](#_Toc57025360)

[4.11.1 Deploying the OS Using ServiceCD 2.0 39](#_Toc57025361)

[4.11.2 Deploying the OS Using Smart Provisioning 43](#_Toc57025362)

[A FAQ 51](#_Toc57025363)

[A.1 How Do I Encrypt Files and View, Edit, and Execute Encrypted Files 51](#_Toc57025364)

[A.1.1 How Do I Encrypt Files 51](#_Toc57025365)

[A.1.2 How Do I View Encrypted Files 52](#_Toc57025366)

[A.1.3 How Do I Edit Encrypted Files 53](#_Toc57025367)

[A.1.4 How Do I Run Configuration Commands After Files Are Encrypted? 53](#_Toc57025368)

[B Getting Help 55](#_Toc57025369)

[B.1 Collecting Fault Information 55](#_Toc57025370)

[B.2 Preparing for Debugging 55](#_Toc57025371)

[B.3 Using Product Documentation 55](#_Toc57025372)

[B.4 Obtaining Technical Support 55](#_Toc57025373)

# Introduction

Function Description

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.

Supported Servers

| Architecture | Type | Model |
| --- | --- | --- |
| x86 | Rack server | RH2288H V3 |
| 2488 V5 |
| 2288H V5 |
| 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 5280) |
| TaiShan 200 server (model 5290) |
| TaiShan 200 server (model 2480) |

Version Mapping

| Version | Mapping Version |
| --- | --- |
| iBMC | * x86 V5 server: V325 or later * x86 V3 server: V323 or later * Arm server:   + - V556 or later (for version number in Vxxx format)     - 3.01.01.00 or later (for version number in 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 is recommended)
* Python: 2.7 or later, 3.7 or later

[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/tree/master/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\_Modul/**

Run the uninstallation command.

**python uninstall.py**

----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 /group\_vars/myhosts](#_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 /group\_vars/myhosts

Set parameters in the **myhosts** file in the **/home/ibmc\_ansible/examples/group\_vars** directory. The user names and passwords of the iBMC users and the SFTP/CIFS/SCP service, SNMP community name, and administrator password for OS deployment can be modified.

Procedure

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**



The **myhosts** file must be encrypted because the file contains sensitive data such as passwords.

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/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**.
* 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**

[root@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: True**   
 **verify: True**   
 **certify:**



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

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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 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.

Querying Help Information

1. Run the following command to view all command module names of the Ansible plug-in. The following uses Python 2 as an example.

**ansible-doc -l |grep ibmc**

[root@localhost ~]# ansible-doc -l |grep ibmc   
[WARNING]: template parsing did not produce documentation.   
[WARNING]: win\_template parsing did not produce documentation.   
ibmc\_ansible\_show\_version Huawei iBMC ansible plugin 2.0.5   
ibmc\_create\_account create a new ibmc account   
ibmc\_create\_raid Create volume   
ibmc\_delete\_account delete bmc account by username   
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\_get\_account get ibmc accounts info   
ibmc\_get\_basic\_info get ibmc basic 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 ibmc accounts info   
ibmc\_modify\_raid Modify volume   
ibmc\_outband\_fw\_update update outband firmware   
ibmc\_server\_profile export or import the server profile   
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\_snmp\_trap Set snmp trap info

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   
> IBMC\_GET\_ACCOUNT (/usr/lib/python2.7/site-packages/ansible-2.8.6-py2.7.egg/ansible/modules/ibmc/ibmc\_get\_account.py)   
   
 get ibmc accounts 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/ibmc\_ansible/examples** directory.

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

1. Run the following command to view the version:

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

[root@localhost examples]# ansible-playbook -v show\_ibmc\_ansible\_version.yml   
   
PLAY [show Huawei iBMC ansible plugin version] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [show Huawei iBMC ansible plugin version] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [127.0.0.1] => {"changed": false, "msg": "**Huawei iBMC ansible modules 2.0.5**"}   
   
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)

## 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/ibmc\_ansible/examples** directory.

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

1. Run the following command:

**ansible-playbook get\_basic\_info.yml**

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples** directory.

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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

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 06:41:10 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/ibmc\_ansible/examples/set\_boot\_device.yml** file.

[root@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"**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples** directory.

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

1. Run the following command:

**ansible-playbook get\_power\_status.yml**

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/set\_power.yml** file.

[root@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"**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples** directory.

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/group\_vars/myhosts** file.

[root@localhost examples]# vi /home/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/ibmc\_ansible/examples/create\_account.yml** file.

[root@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"**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/group\_vars/myhosts** file.

[root@localhost examples]# vi /home/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/ibmc\_ansible/examples/modify\_account.yml** file.

[root@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**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/delete\_account.yml** file.

[root@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"**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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.
* Set the IP address enabling mode and IPv4 and IPv6 addresses.

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

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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

Parameter Configuration

Modify the **/home/ibmc\_ansible/examples/set\_ibmc\_ip.yml** file.

[root@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"**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples** directory.

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/set\_ntp.yml** file.

[root@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**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples** directory.

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/set\_snmp\_trap.yml** file.

[root@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/ibmc\_ansible/examples/group\_vars/myhosts** file.



When **trap\_version** is set to **V1** or **V2C**, set **community** in the **/home/ibmc\_ansible/examples/group\_vars/myhosts** file.

[root@localhost examples]# vi /home/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"

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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

Import or export BIOS and iBMC profiles (configuration files). You can use SFTP to import a profile from the local client to the server or export a profile from the local client or remote server to the **/tmp** directory of iBMC.

Parameter Configuration

Modify the **/home/ibmc\_ansible/examples/server\_profile.yml** file.



If a local temporary directory of the iBMC is used, the directory must be **/tmp**.

[root@localhost examples]# vi server\_profile.yml   
---   
- hosts: myhosts   
 connection: local   
 name: server profile   
 gather\_facts: False   
#file\_name: the file name you want to import or export ;if the file name is empty ,ibmc\_ansible\_profile will used the default name; such as 172.26.201.2\_profile.xml   
#command: Available values:export, import   
#file\_path: Local export: /tmp Remote export: File protocol://Username:Password@IPaddress/Folder The file transfer   
# protocols: Available values: sftp,https,nfs,cifs,scp   
   
 tasks:   
 - name: server profile   
 ibmc\_server\_profile:   
 ibmc\_ip: "{{ ibmc\_ip }}"   
 ibmc\_user: "{{ ibmc\_user }}"   
 ibmc\_pswd: "{{ ibmc\_pswd }}"   
 **command: "export"**   
 **file\_path: "/tmp"**   
 **file\_name: "profile.xml"**

Procedure

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

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

1. Run the following command:

**ansible-playbook server\_profile.yml**

The operation is successful if the following information is displayed:

[root@localhost examples]# ansible-playbook server\_profile.yml   
   
PLAY [server profile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [server profile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host9]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host9 : 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/ibmc\_ansible/examples** directory.

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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 **image\_url** and **protocol** parameters in the **/home/ibmc\_ansible/examples/update\_outband\_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.
* **Protocol**: specifies the file transfer protocol to be used. It can be **SFTP**, **HTTPS**, **NFS**, **CIFS**, or **SCP**.



* 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.

[root@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 }}"   
 **image\_url: "nfs://172.26.200.11/tmp/package/cpldimage.hpm"**   
 **protocol: "NFS**"

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/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).

[root@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/ibmc\_ansible/examples/group\_vars**.

[root@localhost examples]# vi /home/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"

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples** directory.

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/delete\_raid.yml** file.

[root@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"**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/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).

[root@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. It is a mandatory parameter. Format: "disk1,disk2,.,diskN"   
# 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"**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/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).

[root@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"**

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/deploy\_os\_by\_service\_cd.yml** file.

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

[root@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 |

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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/ibmc\_ansible/examples/deploy\_centos7u3\_by\_sp.yml** file.

For details about the parameters, see Table 4-2.

[root@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 (optional). | 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 |

Procedure

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

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

1. Run the following command:

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

The operation is successful if the following information is displayed:

[root@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

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/ibmc\_ansible/examples** directory.

**cd /home/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.

[root@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.

[root@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.

[root@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/ibmc\_ansible/examples** directory.

**cd /home/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.

[root@localhost examples]# ansible-vault view get\_basic\_info.yml   
Vault password:

Enter the file password and view the file content.

[root@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/ibmc\_ansible/examples** directory.

**cd /home/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.

[root@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/ibmc\_ansible/examples** directory.

**cd /home/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 2 as an example.

[root@localhost examples]# ansible-playbook -vv get\_basic\_info.yml --ask-vault-pass   
ansible-playbook 2.8.6   
 config file = /etc/ansible/ansible.cfg   
 configured module search path = [u'/etc/ansible/ansible\_ibmc/module']   
 ansible python module location = /usr/lib/python2.7/site-packages/ansible-2.8.6-py2.7.egg/ansible   
 executable location = /usr/bin/ansible-playbook   
 python version = 2.7.5 (default, Jun 20 2019, 20:27:34) [GCC 4.8.5 20150623 (Red Hat 4.8.5-36)]   
Using /etc/ansible/ansible.cfg as config file   
Vault password:

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

[root@localhost examples]# ansible-playbook -vv get\_basic\_info.yml --ask-vault-pass   
ansible-playbook 2.8.6   
 config file = /etc/ansible/ansible.cfg   
 configured module search path = [u'/etc/ansible/ansible\_ibmc/module']   
 ansible python module location = /usr/lib/python2.7/site-packages/ansible-2.8.6-py2.7.egg/ansible   
 executable location = /usr/bin/ansible-playbook   
 python version = 2.7.5 (default, Jun 20 2019, 20:27:34) [GCC 4.8.5 20150623 (Red Hat 4.8.5-36)]   
Using /etc/ansible/ansible.cfg as config file   
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: [host1] => {"ansible\_facts": {"discovered\_interpreter\_python": "/usr/bin/python"}, "changed": false, "msg": "Get basic info successful! For more detail information, please refer the report log: /var/log/ansible/ibmc/report/basic\_info/\*\*.\*\*.\*\*\*.\*\*\*\_BasicInfo.csv"}   
META: ran handlers   
META: ran handlers   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host1 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

----End

1. Getting Help

If you encounter any problems during routine maintenance or troubleshooting, contact Huawei technical support engineers.

* 1. Collecting Fault Information

Before troubleshooting, obtain the following information:

* Customer company and address
* Contact person and telephone number
* Time when the fault occurred
* Detailed fault symptom
* Device type and software version
* Any measures taken and effects
* Fault severity and expected rectification deadline
  1. Preparing for Debugging

When you seek Huawei technical support, Huawei technical support engineers may assist you in performing some operations to further collect fault information or rectify the fault.

Before contacting technical support engineers, prepare the spare parts for boards and port modules, screwdrivers, screws, serial cables, and network cables.

* 1. Using Product Documentation

Huawei provides the documents delivered with the equipment. This document provides guidance for you to solve common problems that occur during routine maintenance or troubleshooting.

To better rectify the fault, you are advised to use the guide before contacting Huawei technical support engineers.

* 1. Obtaining Technical Support

Huawei's timely and efficient response is available from:

* Local branch offices
* Secondary technical support system
* Telephone support
* Remote support
* Onsite support

Technical Support Website

Obtain technical documents at [Huawei Technical Support](https://support.huawei.com/enterprise/en/index.html) website.

Self-Service Platform and Community

Learn more about servers and communicate with experts at:

* [Computing Product Information Service Platform](https://support-it.huawei.com/server/#/home) for specific server product documentation.
* [Huawei Enterprise iKnow](https://support.huawei.com/iknow/?source=SupportE) for quick learning about products.
* [Huawei Enterprise Support Community (Servers)](https://forum.huawei.com/enterprise/en/Server/forum/895) for learning and discussion.

News

For notices about product life cycles, warnings, and updates, visit [Support > Bulletins > Product Bulletins](https://support.huawei.com/enterprise/en/bulletins-product).

Cases

Learn about server applications at [Computing Case Library](https://support-it.huawei.com/server-knowledgebase/#/home).



The Computing Case Library is available only to Huawei partners and Huawei engineers.

Huawei Technical Support

If a fault persists after taking the above measures, obtain technical support in the following ways:

* Contact Huawei customer service center.
* Enterprise customers

Send emails to [support\_e@huawei.com](mailto:support_e@huawei.com) or visit [Global Service Hotline](https://e.huawei.com/en/service-hotline-query).

* Carriers

Send emails to [support@huawei.com](mailto:support@huawei.com) or visit [Global TAC Information](https://support.huawei.com/carrier/docview!docview?nid=IN0000034614).

* Contact technical support at your local Huawei office.