|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
| **Huawei Ansible Plug-in V1.3.2** | | |
| **User Guide** | | |
| **Issue** | **03** | |
| **Date** | **2019-09-25** | |
|  | | | | |
|  | HUAWEI TECHNOLOGIES CO., LTD. | |  |  |

|  |
| --- |
| **Copyright © Huawei Technologies Co., Ltd. 2019. 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: | <http://e.huawei.com> |
|  |  |

Preface

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 |
| --- | --- | --- |
| 03 | 2019-09-25 | Added 4.5 Managing iBMC Users. |
| 02 | 2018-06-29 | Modified 4.11 Configuring the RAID. |
| 01 | 2018-04-26 | This issue is the first official release. |

Contents

[Preface ii](#_Toc20757100)

[1 Introduction 1](#_Toc20757101)

[2 Installing and Uninstalling the Huawei Ansible Plug-in 2](#_Toc20757102)

[2.1 Environment Dependency 2](#_Toc20757103)

[2.1.1 Software Dependency 2](#_Toc20757104)

[2.1.2 Hardware Dependency 2](#_Toc20757105)

[2.1.3 Version Mapping 2](#_Toc20757106)

[2.2 Installing the Huawei Ansible Plug-in 3](#_Toc20757107)

[2.3 Uninstalling the Huawei Ansible Plug-in 3](#_Toc20757108)

[3 Configuring the Huawei Ansible Plug-in 4](#_Toc20757109)

[3.1 Modifying the /etc/ansible/ansible.cfg File 4](#_Toc20757110)

[3.2 Configuring the /etc/ansible/hosts File 4](#_Toc20757111)

[3.3 Configuring the /group\_vars/myhosts File 4](#_Toc20757112)

[4 Using the Huawei Ansible Plug-in 6](#_Toc20757113)

[4.1 Querying Information (Generating a .csv File) 6](#_Toc20757114)

[4.2 Querying the Health Status (Generating a .csv File) 8](#_Toc20757115)

[4.3 Configuring the Boot Device 9](#_Toc20757116)

[4.4 Managing the Power Supply 10](#_Toc20757117)

[4.5 Managing iBMC Users 11](#_Toc20757118)

[4.5.1 Querying an iBMC User (Generating a .json File) 11](#_Toc20757119)

[4.5.2 Creating an iBMC User 12](#_Toc20757120)

[4.5.3 Modifying an iBMC User 13](#_Toc20757121)

[4.5.4 Deleting an iBMC User 15](#_Toc20757122)

[4.6 Configuring the iBMC 16](#_Toc20757123)

[4.6.1 Querying the iBMC Information 16](#_Toc20757124)

[4.6.2 Configuring the iBMC 17](#_Toc20757125)

[4.7 Managing the NTP Service 18](#_Toc20757126)

[4.7.1 Querying the NTP Service Information 19](#_Toc20757127)

[4.7.2 Configuring the NTP 20](#_Toc20757128)

[4.8 Managing the SNMP Service 21](#_Toc20757129)

[4.8.1 Querying the SNMP Service Information (Generating a .json File) 21](#_Toc20757130)

[4.8.2 Configuring the SNMP Trap 22](#_Toc20757131)

[4.9 Importing and Exporting the Profile File 24](#_Toc20757132)

[4.10 Upgrading the Firmware 25](#_Toc20757133)

[4.10.1 Querying the Firmware Information (Generating a .json File) 25](#_Toc20757134)

[4.10.2 Upgrading the Firmware 26](#_Toc20757135)

[4.10.2.1 BMC Mode 26](#_Toc20757136)

[4.10.2.2 Smart Provisioning Mode 28](#_Toc20757137)

[4.11 Configuring the RAID 29](#_Toc20757138)

[4.11.1 Querying RAID Configuration 29](#_Toc20757139)

[4.11.2 Deleting RAID Group 32](#_Toc20757140)

[4.11.3 Creating RAID Group 33](#_Toc20757141)

[4.11.4 Modifying RAID Configuration 34](#_Toc20757142)

[4.12 Deploying an OS 37](#_Toc20757143)

[4.12.1 ServiceCD 2.0 Mode 37](#_Toc20757144)

[4.12.2 Smart Provisioning Mode 40](#_Toc20757145)

[5 FAQ 45](#_Toc20757146)

[5.1 How Do I Configure Ansible So That Multiple Users in the System Can Use It? 45](#_Toc20757147)

# Introduction

The Huawei Ansible plug-in is a plug-in integrated in the Ansible software and used for Huawei server management. By using Redfish interfaces, the Huawei Ansible plug-in can implement the information and health status query, configuration deployment, and firmware upgrade functions for Huawei servers.

You can implement the following functions by using this plug-in:

* Query the basic information and health status of servers.
* Configure the boot device of servers.
* Manage the power supply of servers.
* Manage iBMC users.
* Query and configure iBMC information.
* Query and configure the NTP service.
* Query and configure the SNMP service.
* Import or export profiles of servers.
* Upgrade server firmware (in BMC or Smart Provisioning mode).
* Deploy server OSs (in ServiceCD 2.0 or Smart Provisioning mode).

# Installing and Uninstalling the Huawei Ansible Plug-in

[2.1 Environment Dependency](#_EN-US_TOPIC_0102786727)

[2.2 Installing the Huawei Ansible Plug-in](#_EN-US_TOPIC_0102786743)

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

## Environment Dependency

### Software Dependency

* sshpass 1.06 or later
* Python 2.7 and Python-request 2.6.0 or later
* jq 1.5 or later
* File server, which must support the CIFS or NFS protocol

### Hardware Dependency

* Configuration of a single RAID controller card is supported: Out-of-band management is supported, such as LSI SAS3108 (Currently, only the LSI SAS3108 RAID controller card is tested.)
* The capacity of the logical disk of the server where the OS is deployed cannot exceed 2 TB; otherwise, it cannot be identified by ServiceCD 2.0.

### Version Mapping

* Only Ansible 2.0.0 or later is supported.
* Only BIOS 0.51 or later is supported.
* Only iBMC 3.20 or later is supported.
* Only Smart Provisioning V116 or later is supported.
* Only ServiceCD 2.0 V133 or later, which can be downloaded at the [Support E](https://support.huawei.com/enterprise/en/server/fusionserver-tools-pid-21015513/software) website.
* Only iBMA V1.30.211 or later (which needs to be installed for in-band query), which can be downloaded at the [Support E](https://support.huawei.com/enterprise/en/server/ibma-pid-21099187/software) website

## Installing the Huawei Ansible Plug-in

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

Upload the installation package to any directory of the Ansible server (such as the **/home** directory).

Log in to the Ansible server as user root.

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

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

**tar -xvf Huawei\_iBMC\_Ansible\_Modules\_x.x.tar**

Run the following command to go to the **/Huawei-iBMC-Ansible\_Modules** directory generated after the decompression:

**cd /Huawei-iBMC\_Ansible\_Modules**

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

**python install.py**

----End

## Uninstalling the Huawei Ansible Plug-in

Log in to the Ansible server as user root.

Go to the /ansible\_ibmc directory, and run the uninstallation command.

**cd /etc/ansible/ansible\_ibmc**

**python uninstall.py**

----End

# Configuring the Huawei Ansible Plug-in

[3.1 Modifying the /etc/ansible/ansible.cfg File](#_EN-US_TOPIC_0102786569)

[3.2 Configuring the /etc/ansible/hosts File](#_EN-US_TOPIC_0102786570)

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

## Modifying the /etc/ansible/ansible.cfg File

[defaults]   
# some basic default values...   
#inventory = /etc/ansible/hosts   
library = /etc/ansible/ansible\_ibmc/module/

## Configuring the /etc/ansible/hosts File

[myhosts]   
host0.domain.com ibmcip=192.168.2.20 host=huaweiserver0   
host1.domain.com ibmcip=192.168.2.21 host=huaweiserver1

## Configuring the /group\_vars/myhosts File

Set related parameters in the **myhosts** file in the **/etc/ansible/ansible\_ibmc/playbooks/group\_vars/** directory, such as the log storage path, iBMC user name and password, and SFTP/CIFS service user name and password.

[root@ansible group\_vars]# cd /etc/ansible/ansible\_ibmc/playbooks/group\_vars/   
[root@ansible group\_vars]# vim 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   
   
# input local directory where all results are placed   
rootdir: /etc/ansible/ansible\_ibmc/report   
reportdir: /etc/ansible/ansible\_ibmc/report   
   
# input True if we want the inventory query result to be comma-separated   
csvformat: True   
   
# input the huawei ibmc user and password   
ibmcuser: "ibmc\_user"   
ibmcpswd: "ibmc\_pwd"   
   
# input the scp user and password when we need to use the scp service   
ibmcuser: "scp\_user"   
ibmcpswd: "scp\_pwd"   
   
# input the sftp user and password when we need to use the sftp service   
sftpuser: "sftp\_user"   
sftppswd: "sftp\_pwd"   
   
# input the cifs user and password when we need to use the cifs service   
cifsuser: "cifs\_user"   
cifspswd: "cifs\_pwd"   
   
   
# if you select SNMP Trap mode as V1 or V2C, you need to set the community name   
community: "community\_name"   
   
 ~



If sensitive data such as passwords are involved, you are advised to encrypt the myhosts file by using the ansible-vault encryption method. For details, see <http://docs.ansible.com/ansible/latest/vault.html>.

# Using the Huawei Ansible Plug-in

[4.1 Querying Information (Generating a .csv File)](#_EN-US_TOPIC_0102786712)

[4.2 Querying the Health Status (Generating a .csv File)](#_EN-US_TOPIC_0102786717)

[4.3 Configuring the Boot Device](#_EN-US_TOPIC_0102786708)

[4.4 Managing the Power Supply](#_EN-US_TOPIC_0102786699)

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

[4.6 Configuring the iBMC](#_EN-US_TOPIC_0193957036)

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

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

[4.9 Importing and Exporting the Profile File](#_EN-US_TOPIC_0102786736)

[4.10 Upgrading the Firmware](#_EN-US_TOPIC_0102786721)

[4.11 Configuring the RAID](#_EN-US_TOPIC_0102786725)

[4.12 Deploying an OS](#_EN-US_TOPIC_0102786703)

## Querying Information (Generating a .csv File)

Function

This plug-in can be used to query the server status, server model, BIOS version, CPLD version, RAID controller card version, NIC version, asset label, serial number, memory size, CPU model, CPU quantity, power supply status, power consumption statistics, BMC version, and boot device.



To query versions of the RAID controller card and NIC, you need to install iBMA V1.30.211 or later, which can be downloaded from [Support E](https://support.huawei.com/enterprise/en/server/ibma-pid-21099187/software).

Parameter Configuration

Fields of the server information to be queried are configured in the **inventory.yml** file. You do not need to modify the file before the query.

After information is queried, the generated **.csv** file is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the **.csv** file to view the information.

Procedure

Go to the **playbooks** directory.

**cd /etc/ansible/ansible\_ibmc/playbooks**

Run query commands.

root@ubuntu:/etc/ansible/ansible\_ibmc/playbooks# ansible-playbook inventory.yml   
   
TASK [Add header row to system information CSV file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
changed: [host3.domain.com -> localhost] => (item=ServerStatus)   
changed: [host3.domain.com -> localhost] => (item=ServerModel)   
changed: [host3.domain.com -> localhost] => (item=BiosVersion)   
changed: [host3.domain.com -> localhost] => (item=AssetTag)   
changed: [host3.domain.com -> localhost] => (item=SerialNumber)   
changed: [host3.domain.com -> localhost] => (item=MemoryGiB)   
changed: [host3.domain.com -> localhost] => (item=MemoryHealth)   
changed: [host3.domain.com -> localhost] => (item=CPUModel)   
changed: [host3.domain.com -> localhost] => (item=CPUHealth)   
changed: [host3.domain.com -> localhost] => (item=CPUCount)   
changed: [host3.domain.com -> localhost] => (item=PowerState)   
changed: [host3.domain.com -> localhost] => (item=ConsumedWatts)   
changed: [host3.domain.com -> localhost] => (item=iBMCFirmwareVersion)   
changed: [host3.domain.com -> localhost] => (item=iBMCHealth)   
changed: [host3.domain.com -> localhost] => (item=BootSourceOverrideMode)   
   
TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
changed: [host3.domain.com -> localhost]   
   
TASK [Combine all host outputs to one CSV file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
changed: [host3.domain.com -> localhost]   
   
TASK [Remove temporary results file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
changed: [host3.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host3.domain.com : ok=88 changed=38 unreachable=0 failed=0

After information is queried, the generated **.csv** file (such as **allhosts\_inventory\_20180507\_220419.csv**) is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the **.csv** file to view the information, as shown in Figure 4-1.

Query results



----End

## Querying the Health Status (Generating a .csv File)

Function

This plug-in can be used to query the server, fan, memory, BMC, power supply, hard disk, and CPU health status.

Parameter Configuration

Fields of the server component health status to be queried are configured in the **healthStatus.yml** file. You do not need to modify the file before the query.

After the server component status is queried, the generated **.csv** file is saved in the **/etc/ansible/ansible\_ibmc/playbooks/report** directory by default. You are advised to export the **.csv** file to view the information.

Procedure

Run query commands.

root@ubuntu:/etc/ansible/ansible\_ibmc/playbooks# ansible-playbook healthStatus.yml   
   
TASK [Add header row to system information CSV file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
changed: [host7.domain.com -> localhost] => (item=ServerStatus)   
changed: [host7.domain.com -> localhost] => (item=CPUHealth)   
changed: [host7.domain.com -> localhost] => (item=MemoryHealth)   
changed: [host7.domain.com -> localhost] => (item=iBMCHealth)   
changed: [host7.domain.com -> localhost] => (item=FansHealth)   
changed: [host7.domain.com -> localhost] => (item=PowerSuppliesHealth)   
changed: [host7.domain.com -> localhost] => (item=DiskHealth)   
   
TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
changed: [host7.domain.com -> localhost]   
TASK [Combine all host outputs to one CSV file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
changed: [host7.domain.com -> localhost]   
TASK [Remove temporary results file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
changed: [host7.domain.com -> localhost]   
TASK [Remove temporary results file] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host7.domain.com -> localhost]   
TASK [command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
skipping: [host7.domain.com]   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host7.domain.com : ok=48 changed=22 unreachable=0 failed=0

After the server component status is queried, the generated **.csv** file (such as **allhosts\_health\_20180507\_223830.csv**) is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the .csv file to view the information, as shown in Figure 4-2.

Query results



## Configuring the Boot Device

Function

This plug-in can be used to set the boot device to PXE, CD, HDD, Floppy, or BIOSSetup. By default, the configuration takes effect only once.

Parameter Configuration

Modify the **/etc/ansible/ansible\_ibmc/setBootDevice.yml** file, including hosts (configuration object) and command (which can be set to Cd, Pxe, Floppy, Hdd, BiosSetup, or None).

- hosts: myhosts   
 name: set boot device   
 gather\_facts: False   
 # boot device:   
 # Cd CD ROM   
 # Pxe pxe   
 # Floppy Floppy   
 # Hdd hard disk device   
 # BiosSetup   
 # None   
 tasks:   
 - name: set boot order   
 local\_action: >   
 ibmc category=SetBootDevice command='Cd' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run configuration commands.

[root@localhost playbooks]# ansible-playbook setBootDevice.yml   
PLAY [set boot device] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
TASK [set boot order] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host7.domain.com -> localhost]   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host7.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical configuration records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgBmc.log   
[2018-01-25 06:00:12 INFO ] (cfgBmc.py:186)- 192.168.3.114 -- config bmc successful!   
[2018-01-25 06:11:10 INFO ] (cfgBmc.py:101)- 192.168.3.114 -- set boot device as:Hdd successful!

## Managing the Power Supply

Function

This plug-in can be used to power on, power off, forcibly restart, or gracefully shut down a server.

Parameter Configuration

Modify the **/etc/ ansible/ansible\_ibmc/playbooks/powerManage.yml** file, including **hosts** (configuration object) and **command**. You can set **command** to **PowerOn** (power-on), **PowerOff** (power-off), **ForceRestart** (forcible restart), **GracefulShutdown** (graceful shutdown), **ForcePowerCycle** (forcible power-off and restart), or **Nmi**.

- hosts: myhosts   
 name: iBMC Manage System Power   
 gather\_facts: False   
   
# Commands available:   
# PowerOn Turn system power on   
# PowerOff Turn system power off, please confirm it again before run the command!   
# ForceRestart Reset OS gracefully, is invalid for server in power-off state   
# GracefulShutdown Shutdown OS gracefully   
# ForcePowerCycle Forcibly power off and then power on   
# Nmi generate a non-maskable interrupt only when the OS is abnormal   
   
 tasks:   
   
 # Except for PowerSate, tasks are mutually exclusive, do only one at a time   
   
 - name: Turn system power on   
 local\_action: >   
 ibmc category=Power command=PowerOn ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run power supply management commands.

[root@localhost playbooks]# ansible-playbook powerManage.yml   
PLAY [iBMC Manage System Power] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
TASK [Force restart] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host7.domain.com -> localhost]   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host7.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical operation records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/pmReport.log   
[2018-01-25 06:03:23 INFO ] (powerManage.py:120)- 192.168.3.114 -- set system force restart successful!

## Managing iBMC Users

Function

This plug-in can be used to query, create, modify, or delete an iBMC user.

### Querying an iBMC User (Generating a .json File)

Parameter Configuration

After information is queried, the generated **.json** file is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the **.json** file to view the information.

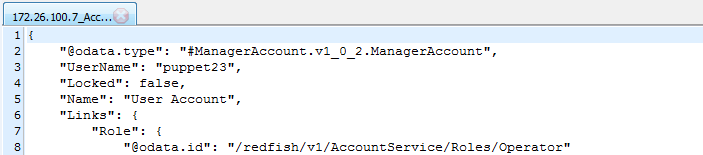
Procedure

Run query commands.

[root@localhost playbooks]# ansible-playbook getAccount.yml   
   
PLAY [Get Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [Get Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

After information is queried, the generated **.json** file (such as **172.26.100.7\_AccountInfo.json**) is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the **.json** file to view the information. See Figure 4-3.

Query result



### Creating an iBMC User

Parameter Configuration

* Modify the **newuser** (user name) and **newpswd** (user password) parameters in the **/etc/ansible/ansible\_ibmc/playbooks/group\_vars/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   
   
# input local directory where all results are placed   
rootdir: /etc/ansible/ansible\_ibmc/report   
reportdir: /etc/ansible/ansible\_ibmc/report   
   
# input True if we want the inventory query result to be comma-separated   
csvformat: True   
   
#used for create new ibmc account   
newuser: "newuser"   
newpswd: "newpswd"   
   
# input the huawei ibmc user and password   
ibmcuser: "ibmc\_user"   
ibmcpswd: "ibmc\_pwd"   
   
# input the sftp user and password when we need to use the sftp service   
sftpuser: "sftp\_user"   
sftppswd: "sftp\_pwd"   
   
# input the cifs user and password when we need to use the cifs service   
cifsuser: "cifs\_user"   
cifspswd: "cifs\_pwd"   
   
   
# if you select SNMP Trap mode as V1 or V2C, you need to set the community name   
community: "community\_name"   
# input the scp user and password when we need to use the scp service   
scpuser: "scp\_user"   
scppswd: "scp\_pwd"

* Modify the **hosts** (configuration object) and **roleid** (user role) parameters in the **/etc/ansible/ansible\_ibmc/playbooks/createAccount.yml** file.

For details about the parameters, see section "Creating a User" in the [*Huawei Server iBMC Redfish API Description*](https://support.huawei.com/enterprise/en/doc/EDOC1000126992).

- hosts: myhosts   
 name: created Account   
 gather\_facts: no   
   
   
 tasks:   
   
   
 - name: create account   
 local\_action: >   
 ibmc category=ManageAccount command="createAccount" useraccount={{ newuser }} newpassword={{ newpswd }} roleid='Administrator' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run query commands.

[root@localhost playbooks]# ansible-playbook createAccount.yml   
   
PLAY [created Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [create account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical creation records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/accountManage.log   
[2019-09-29 06:43:07 INFO ] (accountManage.py:194)- 172.26.100.7 -- create account successful!

### Modifying an iBMC User

Parameter Configuration

Modify the **modifyAccount.json** file in the **/etc/ansible/ansible\_ibmc/configFile/accountCfg/** directory.

For details about the parameters, see section "Modifying User Information" in the [*Huawei Server iBMC Redfish API Description*](https://support.huawei.com/enterprise/en/doc/EDOC1000126992).

{   
"test9": {   
"RoleId":"Operator",   
"Enabled":true,   
"Oem":{   
 "Huawei":{   
 "AccountInsecurePromptEnabled":false,   
 "LoginInterface":["Web","SNMP","IPMI","SSH","SFTP","Local","Redfish"]   
 }   
 }   
 }   
}

Procedure

Run query commands.

[root@localhost playbooks]# ansible-playbook modifyAccount.yml   
   
PLAY [modify Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [modify Account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical modification records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/accountManage.log   
[2019-09-29 06:53:10 INFO ] (accountManage.py:247)- 172.26.100.7 -- modify account successful! respon json is :{u'@odata.type': u'#ManagerAccount.v1\_0\_2.ManagerAccount', u'UserName': u'test5', u'Locked': False, u'Name': u'User Account', u'Links': {u'Role': {u'@odata.id': u'/redfish/v1/AccountService/Roles/Operator'}}, u'RoleId': u'Operator', u'@Message.ExtendedInfo': [{u'Severity': u'Warning', u'MessageId': u'iBMC.1.0.AccountInsecurePromptModificationRestricted', u'RelatedProperties': [], u'Message': u'Only the account itself can modify the status of the account insecure prompt.', u'Resolution': u'None', u'MessageArgs': []}], u'Enabled': True, u'@odata.id': u'/redfish/v1/AccountService/Accounts/6', u'@odata.context': u'/redfish/v1/$metadata#AccountService/Accounts/Members/$entity', u'Oem': {u'Huawei': {u'LoginRule': [], u'PasswordValidityDays': None, u'SSHPublicKeyHash': None, u'LoginInterface': [u'Web', u'SNMP', u'IPMI', u'SSH', u'SFTP', u'Local', u'Redfish'], u'Deleteable': True, u'AccountInsecurePromptEnabled': True, u'Actions': {u'#Account.ImportSSHPublicKey': {u'target': u'/redfish/v1/AccountService/Accounts/6/Oem/Huawei/Actions/Account.ImportSSHPublicKey', u'@Redfish.ActionInfo': u'/redfish/v1/AccountService/Accounts/6/ImportSSHPublicKeyActionInfo'}, u'#Account.DeleteMutualAuthClientCert': {u'target': u'/redfish/v1/AccountService/Accounts/6/Oem/Huawei/Actions/Account.DeleteMutualAuthClientCert', u'@Redfish.ActionInfo': u'/redfish/v1/AccountService/Accounts/6/DeleteMutualAuthClientCertActionInfo'}, u'#Account.DeleteSSHPublicKey': {u'target': u'/redfish/v1/AccountService/Accounts/6/Oem/Huawei/Actions/Account.DeleteSSHPublicKey', u'@Redfish.ActionInfo': u'/redfish/v1/AccountService/Accounts/6/DeleteSSHPublicKeyActionInfo'}, u'#Account.ImportMutualAuthClientCert': {u'target': u'/redfish/v1/AccountService/Accounts/6/Oem/Huawei/Actions/Account.ImportMutualAuthClientCert', u'@Redfish.ActionInfo': u'/redfish/v1/AccountService/Accounts/6/ImportMutualAuthClientCertActionInfo'}}, u'MutualAuthClientCert': None}}, u'Password': None, u'Id': u'6'}

### Deleting an iBMC User

Parameter Configuration

Modify the **hosts** (configuration object) and **useraccount** (name of the user to be deleted) parameters in the **/etc/ansible/ansible\_ibmc/playbooks/deleteAccount.yml** file.

- hosts: myhosts   
 name: delete account   
 gather\_facts: no   
   
   
 tasks:   
   
   
 - name: delete account   
 local\_action: >   
 ibmc category=ManageAccount command="deleteAccount" useraccount="test" ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run query commands.

[root@localhost playbooks]# ansible-playbook deleteAccount.yml   
   
PLAY [delete account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [delete account] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical deletion records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/accountManage.log   
[2019-09-29 06:55:50 INFO ] (accountManage.py:147)- 172.26.100.7 -- delete account successful!

## Configuring the iBMC

Function

* This plug-in can be used to query the iBMC configuration.
* This plug-in can be used to configure the iBMC IP address of a single server or subnet masks and gateways of multiple servers in batches.

### Querying the iBMC Information

Parameter Configuration

Change the value of the **command** parameter in the **/etc/ansible/ansible\_ibmc/playbooks/cfgBMCIP.yml** file to **getinfo**.

- hosts: myhosts   
 name: config bmc ip   
 gather\_facts: False   
   
 tasks:   
   
 # config ibmc ip including IP address;Subnet mask;Gateway, must be separated by semicolons(;)!   
   
 - name: config bmc ip   
 local\_action: >   
 ibmc category=ConfigBMC command='getinfo' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run query commands.

[root@localhost playbooks]# ansible-playbook cfgBMCIP.yml   
   
PLAY [config bmc ip] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [config bmc ip] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

Query the detailed iBMC configuration.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgBmc.log   
[2019-09-25 16:11:22 INFO ] (cfgBmc.py:259)- 172.26.100.7 -- get bmc successful! bmc json is:{u'@odata.type': u'#EthernetInterface.v1\_2\_0.EthernetInterface', u'VLAN': {u'VLANEnable': False, u'VLANId': 0}, u'Name': u'Manager Ethernet Interface', u'IPv6StaticAddresses': [{u'PrefixLength': 64, u'Address': u'2017:5::4'}], u'HostName': u'server7', u'@odata.id': u'/redfish/v1/Managers/1/EthernetInterfaces/485702ab0d57', u'@odata.context': u'/redfish/v1/$metadata#Managers/Members/1/EthernetInterfaces/Members/$entity', u'FQDN': u'server7.', u'NameServers': [u'', u''], u'IPv4Addresses': [{u'SubnetMask': u'255.255.0.0', u'AddressOrigin': u'Static', u'Gateway': u'172.26.0.1', u'Address': u'172.26.100.7'}], u'Link': {u'Oem': {u'Huawei': {u'EthernetInterface': {u'@odata.id': u'/redfish/v1/Systems/1/EthernetInterfaces'}}}}, u'IPv6DefaultGateway': u'2017:5::1', u'PermanentMACAddress': u'48:57:02:ab:0d:57', u'IPv6Addresses': [{u'AddressOrigin': u'Static', u'PrefixLength': 64, u'Address': u'2017:5::4'}, {u'AddressOrigin': u'LinkLocal', u'PrefixLength': 64, u'Address': u'fe80::4a57:2ff:feab:d57'}], u'Id': u'485702ab0d57', u'Oem': {u'Huawei': {u'ManagementNetworkPort': {u'PortNumber': 1, u'Type': u'Dedicated'}, u'NetworkPortMode': u'Fixed', u'DNSAddressOrigin': u'Static', u'IPVersion': u'IPv4AndIPv6', u'ManagementNetworkPort@Redfish.AllowableValues': [{u'PortNumber': 1, u'Type': u'Dedicated', u'LinkStatus': u'Connected'}, {u'PortNumber': 1, u'Type': u'LOM', u'LinkStatus': u'Connected'}, {u'PortNumber': 2, u'Type': u'LOM', u'LinkStatus': u'Disconnected'}, {u'PortNumber': 3, u'Type': u'LOM', u'LinkStatus': u'Disconnected'}, {u'PortNumber': 4, u'Type': u'LOM', u'LinkStatus': u'Disconnected'}], u'AdaptivePort': []}}}

### Configuring the iBMC

Parameter Configuration

1. Set the parameters in the **/etc/ansible/ansible\_ibmc/configFile/ibmcCfg/ibmcIPcfg.json** file.

You can set **AddressOrigin** to **Static** or **DHCPv6**, **IPVersion** to **IPv4**, **IPv6**, or **IPv4AndIPv6**.

For details about the parameters, see section "Modifying iBMC Network Port Information" in the [*Huawei Server iBMC Redfish API Description*](https://support.huawei.com/enterprise/en/doc/EDOC1000126992).

{   
 "IPv4Addresses" :   
 [   
 {   
 "Address":"172.26.100.10",   
 "SubnetMask":"255.255.0.0",   
 "AddressOrigin":"Static",   
 "Gateway":"172.26.0.1"   
 }   
 ],   
 "IPv6Addresses":[   
 {   
 "Address":"2017:5::9",   
 "PrefixLength":"32",   
 "AddressOrigin":"Static"   
 }   
   
 ],   
 "IPv6DefaultGateway": "2017:5::1",   
 "IPVersion": "IPv4AndIPv6"   
   
}

1. Modify the **hosts** (configuration object) and **command** (path of the **ibmcIPcfg.json** file) parameters in the **/etc/ansible/ansible\_ibmc/playbooks/cfgBMCIP.yml** file.

- hosts: myhosts   
 name: config bmc   
 gather\_facts: False   
   
 tasks:   
   
 # Except for PowerSate, tasks are mutually exclusive, do only one at a time   
   
 - name: config bmc ip   
 local\_action: >   
 ibmc category=ConfigBMC command='/etc/ansible/ansible\_ibmc/configFile/ibmcCfg/ibmcIPcfg.json' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run configuration commands.

[root@localhost playbooks]# ansible-playbook cfgBMCIP.yml   
PLAY [config bmc] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
TASK [config bmc ip] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host7.domain.com -> localhost]   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host7.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical configuration records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgBmc.log   
[2018-01-25 06:00:12 INFO ] (cfgBmc.py:186)- 192.168.3.114 -- config bmc successful!

## Managing the NTP Service

Function

* This plug-in can be used to query the NTP service information.
* This plug-in can be used to configure the NTP service enabling, primary and secondary NTP server addresses, server identification enabling, NTP address mode (IPv4/IPv6/Static), and minimum and maximum polling intervals.

### Querying the NTP Service Information

Parameter Configuration

Change the value of the **command** parameter in the **/etc/ansible/ansible\_ibmc/playbooks/cfgNTP.yml** file to **getinfo**.

- hosts: myhosts   
 name: config NTP   
 gather\_facts: False   
   
 tasks:   
   
 # Config NTP function as below:   
 #   
 # 1,"ServiceEnabled": ServiceEnabled, true/false   
 # 2,"AlternateNtpServer": AlternateNtpServer, ip/domain name   
 # 3,"PreferredNtpServer": PreferredNtpServer, ip/domain name   
 # 4,"ServerAuthenticationEnabled": ServerAuthenticationEnabled, true/false   
 # 5,"NtpAddressOrigin": NtpAddressOrigin, IPv4/IPv6/Static   
 # 6,"MinPollingInterval": MinValue, 3~17 and less than MaxValue   
 # 7,"MaxPollingInterval": MaxValue, 3~17 and bigger than MinValue   
 #   
   
 - name: config bmc ntp   
 local\_action: >   
 ibmc category=ConfigNTP command='getinfo' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run query commands.

[root@localhost playbooks]# ansible-playbook cfgNTP.yml   
   
PLAY [config NTP] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [config bmc ntp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

Query the detailed NTP service configuration.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgNTP.log   
[2019-09-25 16:41:35 INFO ] (cfgNTP.py:189)- 172.26.100.7 -- get NTP successful! NTPInfo is:ServiceEnabled: False PreferredNtpServer:172.26.100.101 AlternateNtpServer:172.26.100.100 NtpAddressOrigin:Static MinPollingInterval:5 MaxPollingInterval:10 ServerAuthenticationEnabled:False NTPKeyStatus:Uploaded

### Configuring the NTP

Parameter Configuration

1. Configure the **/etc/ansible/ansible\_ibmc/configFile/ntpCfg/ntp.ini** file.

For details about the parameters, see section "Modifying NTP Resource Properties" in the [*Huawei Server iBMC Redfish API Description*](https://support.huawei.com/enterprise/en/doc/EDOC1000126992).

[NTP]   
# This is an ini file for bmc ntp config.   
   
# 1,"ServiceEnabled": ServiceEnabled, true/false   
# 2,"AlternateNtpServer": AlternateNtpServer, ip/domain name   
# 3,"PreferredNtpServer": PreferredNtpServer, ip/domain name   
# 4,"ServerAuthenticationEnabled": ServerAuthenticationEnabled, true/false   
# 5,"NtpAddressOrigin": NtpAddressOrigin, IPv4/IPv6/Static   
# 6,"MinPollingInterval": MinValue, 3~17 and <= MaxValue   
# 7,"MaxPollingInterval": MaxValue, 3~17 and >= MinValue   
   
   
# Enable or disable bmc ntp service,you can select true or false.   
ServiceEnabled = true   
# Config alternate NtpServer, you can enter ip or domain name.   
AlternateNtpServer = 192.168.100.2   
# Config preferred NtpServer, you can enter ip or domain name.   
PreferredNtpServer = 192.168.100.1   
# Enable or disable Server Authentication service,you can select true or false.   
ServerAuthenticationEnabled = false   
# Config Ntp Address Origin, you can select from IPv4/IPv6/Static.   
NtpAddressOrigin = IPv4   
# Config Min Polling Interval time, in 3~17 and <= MaxValue   
MinPollingInterval = 3   
# Config Max Polling Interval time, in 3~17 and >= MinValue   
MaxPollingInterval = 17

1. Modify the **/etc/ansible/ansible\_ibmc/playbooks/cfgNTP.yml** file, including hosts (configuration object) and command (path of the **ntp.ini** file).

- hosts: myhosts   
 name: config NTP   
 gather\_facts: False   
   
 tasks:   
   
 # Config NTP function as below:   
 #   
   
 - name: config bmc ntp   
 local\_action: >   
 ibmc category=ConfigNTP command='/etc/ansible/ansible\_ibmc/configFile/ntpCfg/ntp.ini' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run configuration commands.

[root@ansible playbooks]# ansible-playbook cfgNTP.yml   
   
PLAY [config NTP] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [config bmc ntp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
ok: [host1.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0   
host1.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical configuration records.

[root@ansible playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgNTP.log   
[2018-03-22 19:38:20 INFO ] (cfgNTP.py:162)- 192.168.12.10 -- config bmc ntp successfully!   
[2018-03-22 19:38:20 INFO ] (cfgNTP.py:162)- 192.168.3.114 -- config bmc ntp successfully!

## Managing the SNMP Service

Function

* This plug-in can be used to query the SNMP service information.
* This plug-in can be used to configure the enabling status, target IP address (four groups), trap mode, trap identifier, and trap severity of the trap function.

### Querying the SNMP Service Information (Generating a .json File)

Parameter Configuration

Change the value of the **command** parameter in the **/etc/ansible/ansible\_ibmc/playbooks/cfgTrap.yml** file to **getinfo**.

- hosts: myhosts   
 name: config snmp trap   
 gather\_facts: no   
   
 tasks:   
   
   
 - name: config snmp   
 local\_action: >   
 ibmc category=CfgSnmpTrap command="getinfo" ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }} extraparam={{ community }}

After information is queried, the generated **.json** file is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the **.json** file to view the information.

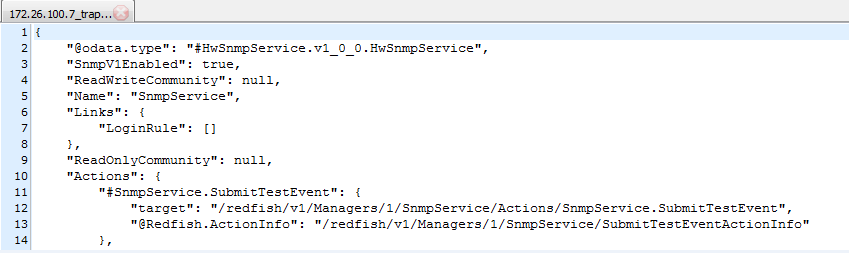
Procedure

Run query commands.

[root@localhost playbooks]# ansible-playbook cfgTrap.yml   
   
PLAY [config snmp trap] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [config snmp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

After information is queried, the generated **.json** file (such as **172.26.100.7\_trapInfo.json**) is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the **.json** file to view the information. See Figure 4-4.

Query result



### Configuring the SNMP Trap

Parameter Configuration

1. Set the related parameters in the **/etc/ansible/ansible\_ibmc/configFile/snmpCfg/snmpTrap.ini** file.



If **TrapVersion** is set to **V3**, **TrapV3User** must be configured. If **TrapVersion** is set to **V1** or **V2C**, **community** must be configured in the **/etc/ansible/ansible\_ibmc/playbooks/group\_vars/myhosts** file.

[snmpTrapNotification]   
# snmp trap config   
# enable or disable snmp trap service,you can choice Y or N   
ServiceEnabled = Y   
# you can choice V1/V2C/V3   
TrapVersion = V3   
TrapV3User = root   
# you can choice OID/EventCode/PreciseAlarm   
TrapMode = OID   
# you can choice BoardSN/ProductAssetTag/HostName   
TrapServerIdentity = HostName   
# you can choice Critical/Major/Minor/Normal   
AlarmServerity = Normal   
TrapDestNum = 4   
   
# Trap destination Server 1   
[trapDest1]   
TrapEnabled = Y   
TrapServerAddress = 192.168.2.196   
TrapServerPort = 161   
   
# Trap destination Server 2   
[trapDest2]   
TrapEnabled = Y   
TrapServerAddress = 192.168.2.6   
TrapServerPort = 1612   
   
# Trap destination Server 3   
[trapDest3]   
TrapEnabled = N   
TrapServerAddress = 192.168.2.7   
TrapServerPort = 1613   
   
# Trap destination Server 4   
[trapDest4]   
TrapEnabled = N   
TrapServerAddress = 192.168.2.8   
TrapServerPort = 1614

1. Modify the **/etc/ansible/ansible\_ibmc/playbooks/cfgTrap.yml** file, including hosts (configuration object) and command (path of the **snmpTrap.ini** file).

- hosts: myhosts   
 name: config snmp trap   
 gather\_facts: no   
   
 tasks:   
   
   
 - name: config snmp   
 local\_action: >   
 ibmc category=CfgSnmpTrap command="/etc/ansible/ansible\_ibmc/configFile/snmpCfg/snmpTrap.ini" ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }} extraparam={{ community }}

Procedure

Run configuration commands.

[root@foreman playbooks]# ansible-playbook cfgTrap.yml   
   
PLAY [config snmp trap] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [config snmp] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host114.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host114.domain.com : ok=1 changed=0 unreachable=0 failed=0   
   
[root@foreman playbooks]#

View historical configuration records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgSnmpReport.log   
[2018-01-25 10:15:49 INFO ] (cfgTrap.py:159)- 192.168.3.114 -- config snmp trap successful!

## Importing and Exporting the Profile File

Function

This plug-in can be used to import and export the BIOS and iBMC Profile configuration file. The Profile file can be imported from a local path to the server by using the SFTP protocol or be exported from the server to the **/tmp** directory in the iBMC.

Parameter Configuration

Configure the command parameter in the **/etc/ansible/ansible\_ibmc/playbooks/serverProfile.yml** file.

hosts: myhosts   
 name: import or export server profile   
 gather\_facts: False   
 # default export profile as /tmp/profile.xml if cli has no paramters   
 # export example: 'import;sftp://{{ sftpuser }}:{{ sftppswd }}@192.168.2.156/etc/profile.xml'   
 vars:   
 command: 'export;/tmp/profile.xml'   
 tasks:   
 - name: import or export server profile   
 local\_action: >   
 ibmc category=Profile command={{ command }} ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Export the Profile file to the **/tmp** directory in the iBMC.

[root@foreman playbooks]# ansible-playbook serverProfile.yml   
   
PLAY [import or export server profile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [export server profile] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host114.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host114.domain.com : ok=1 changed=0 unreachable=0 failed=0   
   
[root@foreman playbooks]#

View historical operation records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/serverProfile.log   
[2018-01-25 10:17:33 INFO ] (serverProfile.py:170)- 192.168.3.114 -- EXPORT:/tmp/profile.xml successful!

## Upgrading the Firmware

Function

* This plug-in can be used to query the firmware information.
* This plug-in can be used to upgrade the BMC, BIOS, and CPLD.

### Querying the Firmware Information (Generating a .json File)

Parameter Configuration

After information is queried, the generated **.json** file is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the **.json** file to view the information.

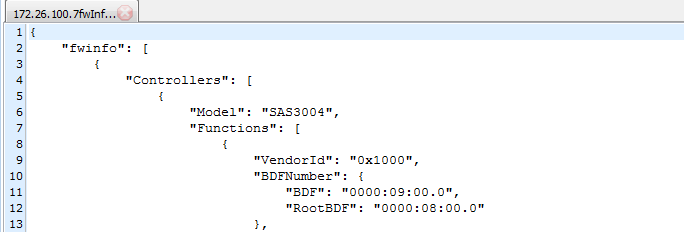
Procedure

Run query commands.

[root@localhost playbooks]# vi getFwInfo.yml   
[root@localhost playbooks]# ansible-playbook getFwInfo.yml   
   
PLAY [update firm ware] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [update frimware] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host0.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

After information is queried, the generated **.json** file (such as **172.26.100.7fwInfo.json**) is saved in the **/etc/ansible/ansible\_ibmc/report** directory by default. You are advised to export the **.json** file to view the information. See Figure 4-5.

Query result



### Upgrading the Firmware

#### BMC Mode

Parameter Configuration

Modify the **/etc/ansible/ansible\_ibmc/playbooks/upgradeFw.yml** file, including hosts (configuration object) and command (path of the image file of the firmware to be upgraded).

- hosts: myhosts   
 name: update firm ware   
 gather\_facts: no   
   
 tasks:   
   
 - name: update frimware   
 local\_action: >   
 ibmc category=UpdateFW command='nfs://192.168.2.241/NFSshare/2288V3.hpm' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run upgrade commands.

[root@localhost playbooks]# ansible-playbook updateFW.yml   
   
PLAY [update firm ware] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [update frimware] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host4.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host4.domain.com : ok=1 changed=0 unreachable=0 failed=0

View the upgrade progress.

[root@localhost log]# tail -f /etc/ansible/ansible\_ibmc/log/updateFwLog.log   
[2018-01-25 08:36:41 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:None   
[2018-01-25 08:36:43 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:0%   
[2018-01-25 08:36:45 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:3%   
[2018-01-25 08:36:48 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:4%   
[2018-01-25 08:36:50 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:5%   
[2018-01-25 08:36:53 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:6%   
[2018-01-25 08:36:55 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:9%   
[2018-01-25 08:36:58 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:10%   
[2018-01-25 08:37:01 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:11%   
[2018-01-25 08:37:03 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:13%   
[2018-01-25 08:37:05 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:16%   
[2018-01-25 08:37:08 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:17%   
[2018-01-25 08:37:10 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:18%   
…   
[2018-01-25 08:38:30 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:78%   
[2018-01-25 08:38:33 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:80%   
[2018-01-25 08:38:35 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:82%   
[2018-01-25 08:38:37 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:84%   
[2018-01-25 08:38:40 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:86%   
[2018-01-25 08:38:43 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:91%   
[2018-01-25 08:38:45 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:95%   
[2018-01-25 08:38:47 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:97%   
[2018-01-25 08:38:49 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Running percent:97%   
[2018-01-25 08:38:52 INFO ] (updateFW.py:123)- 192.168.3.114 -- status:Completed percent:100%   
[2018-01-25 08:38:52 INFO ] (updateFW.py:128)- 192.168.3.114 -- update 1288HV5260.hpm successful!

View historical upgrade records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/updateFwReport.log   
[2018-01-25 08:38:52 INFO ] (updateFW.py:129)- 192.168.3.114 -- update 1288HV5260.hpm successful!

#### Smart Provisioning Mode

Function

This plug-in can be used to upgrade the firmware of the RAID controller cards, NICs, and hard disks of V5 servers. For details, see the [*FusionServer Tools V200R002C10 Smart Provisioning User Guide*](http://support.huawei.com/enterprise/en/doc/EDOC1100002509?idPath=7919749|9856522|9856629|21015513).

Parameter Configuration

Configure the **imageurl** (path of the upgrade package) parameter in the **/etc/ansible/ansible\_ibmc/configFile/upgradeFwBySpCfg/upgradeFwBySp.ini** file. You need to upload upgrade package and upgrade package verification file to the same path. If multiple firmware needs to be upgraded at the same time, you need to enter the paths of each firmware upgrade package. You can use **fw1** and **fw2** to distinguish different firmware.

[fw1]   
imageurl=nfs://192.168.2.241/NFSshare/inbandfw/BC1SPSCA03-10GE BASE T-X722 FW-V102.zip   
[fw2]   
imageurl=nfs://192.168.2.241/NFSshare/inbandfw/ RAID-SR430C(3108)-FW-V108(4.650.00-6121).zip

Procedure

Run upgrade commands.

[root@ansible playbooks]# ansible-playbook upgradeFwBySp.yml   
   
PLAY [update firm ware] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [update frimware] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host4.domain.com : ok=1 changed=0 unreachable=0 failed=0

View the upgrade progress.

[root@localhost log]# tail -f /etc/ansible/ansible\_ibmc/log/upgradFwBySp.log   
[2018-09-05 21:39:18 INFO ] (upgradeFwBySp.py:69)- 192.168.3.10 -- ForceOff system successfully!   
[2018-09-05 21:39:18 INFO ] (upgradeFwBySp.py:78)- 192.168.3.10 -- GetFwUpdateId successfully!   
[2018-09-05 21:39:19 INFO ] (upgradeFwBySp.py:111)- 192.168.3.10 -- spApiSetFwUpgrade successfully!   
[2018-09-05 21:39:40 INFO ] (upgradeFwBySp.py:128)- 192.168.3.10 -- spApiGetFWSource successfully!   
[2018-09-05 21:39:44 INFO ] (upgradeFwBySp.py:149)- 192.168.3.10 -- spAPISetSpService successfully!   
[2018-09-05 21:39:46 INFO ] (upgradeFwBySp.py:164)- 192.168.3.10 -- poweron system successfully!   
…   
[2018-09-05 21:47:22 INFO ] (upgradeFwBySp.py:197)- have got no upgrade result   
[2018-09-05 21:47:33 INFO ] (upgradeFwBySp.py:197)- have got no upgrade result   
…   
[2018-09-05 21:59:39 INFO ] (upgradeFwBySp.py:185)- upgrade has not finished   
[2018-09-05 21:59:47 INFO ] (upgradeFwBySp.py:185)- upgrade has not finished   
[2018-09-05 21:59:56 INFO ] (upgradeFwBySp.py:207)- 192.168.3.10 -- upgrade successfully

View historical upgrade records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/updateFwReport.log   
[2018-09-05 21:59:56 INFO ] (upgradeFwBySp.py:208)- 192.168.3.10 -- upgrade successfully

## Configuring the RAID

Function

This plug-in can be used to configure RAID0/RAID10/RAID5/RAID50/RAID6/RAID60, query RAID configuration information, configure boot disks, create logical disks and configure related parameters, and delete RAID configuration.

### Querying RAID Configuration

Run query commands.

[root@ansible playbooks]# ansible-playbook getRaidInfo.yml   
   
PLAY [get logic device info] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [get logic device info] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host3.domain.com -> localhost]   
ok: [host2.domain.com -> localhost]   
ok: [host0.domain.com -> localhost]   
ok: [host1.domain.com -> localhost]   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0   
host1.domain.com : ok=1 changed=0 unreachable=0 failed=0   
host2.domain.com : ok=1 changed=0 unreachable=0 failed=0   
host3.domain.com : ok=1 changed=0 unreachable=0 failed=0

View the brief information of RAID groups on all servers.

[root@ansible report]# cat /etc/ansible/ansible\_ibmc/report/getRaidInfoReport.log   
[2018-05-09 22:18:44 INFO ] (getRaidInfo.py:188)- 192.168.3.114 get raid info successfully, totalInfo:   
 ===========================================================   
/redfish/v1/Systems/1/Storages/RAIDStorage0:   
raidmodle:SAS3108   
-unconfigDrivers:   
--HDDPlaneDisk0   
--HDDPlaneDisk1   
--HDDPlaneDisk2   
--HDDPlaneDisk3   
for more info please refer to /etc/ansible/ansible\_ibmc/report/192.168.3.114\_raidInfo.json   
   
[2018-05-09 22:18:49 INFO ] (getRaidInfo.py:188)- 192.168.3.7 get raid info successfully, totalInfo:   
 ===========================================================   
/redfish/v1/Systems/1/Storages/RAIDStorage1:   
raidmodle:SAS3408   
-unconfigDrivers:   
===========================================================   
/redfish/v1/Systems/1/Storages/RAIDStorage0:   
raidmodle:SAS3108   
-LogicalDrive0:   
--raidLevel:RAID0   
--drivers:   
---HDDPlaneDisk3   
---HDDPlaneDisk4   
-unconfigDrivers:   
--HDDPlaneDisk0   
--HDDPlaneDisk1   
--HDDPlaneDisk2   
--HDDPlaneDisk5   
--HDDPlaneDisk6   
--HDDPlaneDisk7   
--HDDPlaneDisk40   
--HDDPlaneDisk41   
for more info please refer to /etc/ansible/ansible\_ibmc/report/192.168.3.7\_raidInfo.json

View the detailed information of the RAID group on a server.

[root@ansible]# cat /etc/ansible/ansible\_ibmc/report/192.168.3.114\_raidInfo.json   
 "/redfish/v1/Systems/1/Storages/RAIDStorage0": {   
 "@odata.type": "#Storage.v1\_1\_0.Storage",   
 "StorageControllers@odata.count": 1,   
 "Name": "RAIDStorage0",   
 "raidinfo": {   
 "/redfish/v1/Chassis/1/Boards/mainboardRAIDCard1--SAS3108": {   
 "@odata.type": "#HwBoard.v1\_0\_0.HwBoard",   
 "Status": {   
 "State": "Enabled",   
 "Health": "OK"   
 },   
 "PCBVersion": ".B",   
 "CPLDVersion": null,   
 "Name": "mainboardRAIDCard1",   
 "AssetTag": null,   
 "CardNo": 1,   
 "SerialNumber": null,   
 "@odata.id": "/redfish/v1/Chassis/1/Boards/mainboardRAIDCard1",   
 "@odata.context": "/redfish/v1/$metadata#Chassis/Members/1/Boards/Members/$entity",   
 "DeviceLocator": "RAIDCard1",   
 "ManufactureDate": "2016/11/09 Wed 07:35:00",   
 "BoardId": "0x0024",   
 "DeviceType": "RAIDCard",   
 "Location": "mainboard",   
 "BoardName": "BC61ESMLB",   
 "PartNumber": null,   
 "ProductName": "SR530C-M 1G",   
 "Id": "mainboardRAIDCard1",   
 "Manufacturer": "Huawei"   
 }   
 },   
 "@odata.id": "/redfish/v1/Systems/1/Storages/RAIDStorage0",   
 "Actions": {   
 "Oem": {   
 "Huawei": {   
 "#Storage.RestoreStorageControllerDefaultSettings": {   
 "target": "/redfish/v1/Systems/1/Storages/RAIDStorage0/Actions/Oem/Huawei/Storage.RestoreStorageControllerDefaultSettings",   
 "@Redfish.ActionInfo": "/redfish/v1/Systems/1/Storages/RAIDStorage0/RestoreStorageControllerDefaultSettingsActionInfo"   
 }   
 }   
 }   
 },   
 "Drives": [   
 {   
 "@odata.id": "/redfish/v1/Chassis/1/Drives/HDDPlaneDisk0"   
 },   
 {   
 "@odata.id": "/redfish/v1/Chassis/1/Drives/HDDPlaneDisk1"   
 },   
 {   
 "@odata.id": "/redfish/v1/Chassis/1/Drives/HDDPlaneDisk2"   
 },   
 {   
 "@odata.id": "/redfish/v1/Chassis/1/Drives/HDDPlaneDisk3"   
 }

### Deleting RAID Group

Parameter Configuration

Modify the **command** parameter in the **/etc/ansible/ansible\_ibmc/playbooks/deleteRaid.yml** file. You can delete a specified RAID group of a storage device (set **command** to **RAIDStorage0/LogicalDrive0**, and set **category** to **DelALD**) or all RAID groups of a storage device (set **command** to **RAIDStorage0**, and set **category** to **DelLD**). You can also delete all RAID groups of all storage devices (set **command** to **ALL**, and set **category** to **DelLD**).

---   
- hosts: myhosts   
 name: delete raid   
 gather\_facts: no   
   
   
 tasks:   
   
 - name: delete a raid in a storage   
 local\_action: >   
 ibmc category='DelALD' command='RAIDStorage0/LogicalDrive0' extraparam='' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}   
   
 #- name: delete ALL raid in a storage   
 # local\_action: >   
 # ibmc category='DelLD' command='RAIDStorage0' extraparam='' ibmcip={{ ibmcip }}   
 # ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}   
 #- name: delete ALL raid in each storage   
 # local\_action: >   
 # ibmc category='DelLD' command='ALL' extraparam='' ibmcip={{ ibmcip }}   
 # ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run deletion commands.

[root@ansible playbooks]# ansible-playbook deleteRaid.yml   
   
PLAY [delete raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [delete a raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host1.domain.com -> localhost]   
 to retry, use: --limit @/etc/ansible/ansible\_ibmc/playbooks/deleteRaid.retry   
   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host1.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical deletion records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgRaidReport.log   
[2018-05-09 23:52:03 INFO ] (cfgRaid.py:170)- 192.168.3.7 RAIDStorage0/LogicalDrive0 delete LD successful

### Creating RAID Group

Parameter Configuration

Modify the **createRaidtmp.json** file in the **/etc/ansible/ansible\_ibmc/configFile/raidCfg/** directory. You can configure one or multiple RAID groups. For details about how to set the parameters, see section "Creating a Logical Drive" in the [*Huawei Server iBMC Redfish API Description*](http://support.huawei.com/enterprise/en/doc/EDOC1000126992).



When RAID controller cards of different models are used to create logical disks, the parameter values supported by the RAID controller cards may be different. For details, see the parameters supported by the corresponding RAID controller card.

[root@ansible]# vim createRaidtmp.json   
{   
 "LDlist":[   
 {   
 "RAIDID" :"RAIDStorage0/Volumes" ,   
 "configDic":   
 {   
 "CapacityBytes":104857600000 ,   
 "OptimumIOSizeBytes" : 65536,   
 "Oem":{   
 "Huawei": {   
 "CreateCacheCadeFlag" :false,   
 "Drives" : [0,1],   
 "VolumeName": "testVol1",   
 "VolumeRaidLevel": "RAID0",   
 "DefaultReadPolicy": "ReadAhead",   
 "DefaultWritePolicy" : "WriteThrough",   
 "DefaultCachePolicy" : "CachedIO",   
 "SpanNumber": 1,   
 "AccessPolicy" : "ReadWrite",   
 "DriveCachePolicy" : "Disabled",   
 "InitializationMode" : "UnInit"   
 }   
 }   
 },   
{   
 "RAIDID" :"RAIDStorage0/Volumes" ,   
 "configDic":   
 {   
 "CapacityBytes":104857600000 ,   
 "OptimumIOSizeBytes" : 65536,   
 "Oem":{   
 "Huawei": {   
 "CreateCacheCadeFlag" :false,   
 "Drives" : [2,3],   
 "VolumeName": "testVol2",   
 "VolumeRaidLevel": "RAID1",   
 "DefaultReadPolicy": "ReadAhead",   
 "DefaultWritePolicy" : "WriteThrough",   
 "DefaultCachePolicy" : "CachedIO",   
 "SpanNumber": 1,   
 "AccessPolicy" : "ReadWrite",   
 "DriveCachePolicy" : "Disabled",   
 "InitializationMode" : "UnInit"   
 }   
 }   
 }   
 ]   
}

Procedure

Run creation commands.

[root@localhost playbooks]# ansible-playbook createRaid.yml   
PLAY [config raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
TASK [config raid1] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host7.domain.com -> localhost]   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host7.domain.com : ok=1 changed=0 unreachable=0 failed=0

View historical creation records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgRaidReport.log   
[2018-05-10 01:52:05 INFO ] (cfgRaid.py:549)- 192.168.3.7 cfgRaid successlly ,result :[{u'RAIDStorage0/Volumes': 'success'}]

### Modifying RAID Configuration

Parameter Configuration

Modify the **modifyRaid.json** file in the **/etc/ansible/ansible\_ibmc/configFile/raidCfg/** directory. For details about how to set the parameters, see section "Modifying Attributes of a Logical Drive Resource" in the [*Huawei Server iBMC Redfish API Description*](http://support.huawei.com/enterprise/en/doc/EDOC1000126992).

[root@ansible]# vim modifyRaid.json   
{   
 "ldlist":   
 [   
 {   
 "LDID":"RAIDStorage0/Volumes/LogicalDrive0",   
 "LDConfig":   
 {   
 "Oem":   
 {   
 "Huawei":   
 {   
 "VolumeName": "testName1",   
 "DefaultReadPolicy": "NoReadAhead",   
 "DefaultWritePolicy": "WriteThrough",   
 "DefaultCachePolicy": "CachedIO",   
 "BootEnable": true,   
 "BGIEnable": true,   
 "AccessPolicy": "ReadWrite",   
 "SSDCachingEnable": true,   
 "DriveCachePolicy": "Unchanged"   
 }   
 }   
 }   
 },   
 {   
   
 "LDID":"RAIDStorage0/Volumes/LogicalDrive1",   
 "LDConfig":   
 {   
 "Oem":   
 {   
 "Huawei":   
 {   
 "VolumeName": "testName2",   
 "DefaultReadPolicy": "NoReadAhead",   
 "DefaultWritePolicy": "WriteThrough",   
 "DefaultCachePolicy": "CachedIO",   
 "BootEnable": false,   
 "BGIEnable": true,   
 "AccessPolicy": "ReadWrite",   
 "SSDCachingEnable": true,   
 "DriveCachePolicy": "Unchanged"   
 }   
 }   
 }   
   
 }   
 ]   
}

Procedure

Run modification commands.

[root@ansible playbooks]# ansible-playbook modifyRaid.yml   
   
PLAY [modify raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
   
TASK [modify raid] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host0.domain.com : ok=1 changed=0 unreachable=0 failed=0

To view the modified RAID configuration information, run the **ansible-playbook getRaidInfo.yml** command to obtain the latest RAID configuration information, and then view the **/etc/ansible/ansible\_ibmc/report/*192.168.3.7*\_raidInfo.json** file.

[root@ansible report]# vim 192.168.3.7\_raidInfo.json   
{   
 "/redfish/v1/Systems/1/Storages/RAIDStorage1": {   
 "@odata.type": "#Storage.v1\_1\_0.Storage",   
 "StorageControllers@odata.count": 1,   
 "Name": "RAIDStorage1",   
 "raidinfo": {   
 "/redfish/v1/Chassis/1/Boards/mainboardPCIeCard4(RAID)--SAS3408": {   
 "@odata.type": "#HwBoard.v1\_0\_0.HwBoard",   
 "Status": {   
 "State": "Enabled",   
 "Health": "OK"   
 },   
 "PCBVersion": ".A",   
 "CPLDVersion": null,   
 "Name": "mainboardPCIeCard4(RAID)",   
 "AssetTag": null,   
 "CardNo": 4,   
 "SerialNumber": null,   
 "@odata.id": "/redfish/v1/Chassis/1/Boards/mainboardPCIeCard4(RAID)",   
 "@odata.context": "/redfish/v1/$metadata#Chassis/Members/1/Boards/Members/$entity",   
 "DeviceLocator": "PCIeCard4(RAID)",   
 "ManufactureDate": "2018/03/18 Sun 16:41:00",   
 "BoardId": "0x0059",   
 "DeviceType": "PCIeCard",   
 "Location": "mainboard",   
 "BoardName": null,   
 "PartNumber": null,   
 "ProductName": null,   
 "Id": "mainboardPCIeCard4(RAID)",   
 "Manufacturer": "LSI Logic / Symbios Logic"   
 }   
 },   
 "@odata.id": "/redfish/v1/Systems/1/Storages/RAIDStorage1",   
 "Actions": {   
 "Oem": {   
 "Huawei": {   
 "#Storage.RestoreStorageControllerDefaultSettings": {   
 "target": "/redfish/v1/Systems/1/Storages/RAIDStorage1/Actions/Oem/Huawei/Storage.RestoreStorageControllerDefaultSettings",   
 "@Redfish.ActionInfo": "/redfish/v1/Systems/1/Storages/RAIDStorage1/RestoreStorageControllerDefaultSettingsActionInfo"   
 }   
 }   
 }   
 },   
 "Drives": [],   
 "driverinfo": {},   
 "@odata.context": "/redfish/v1/$metadata#Systems/Members/1/Storages/Members/$entity",   
 "Volumes": {   
 "@odata.id": "/redfish/v1/Systems/1/Storages/RAIDStorage1/Volumes"   
 },   
 "StorageControllers": [   
 {   
 "Status": {   
 "State": "Enabled",   
 "Health": "OK"   
 },   
 "Description": "RAID Controller",   
"192.168.3.7\_raidInfo.json" [noeol] 876L, 39642C

View historical modification records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/cfgRaidReport.log   
[2018-05-11 01:04:55 INFO ] (cfgRaid.py:435)- 192.168.3.7 modify raid successlly ,result :[{u'RAIDStorage0/Volumes/LogicalDrive0': 'success'}, {u'RAIDStorage0/Volumes/LogicalDrive1': 'success'}]

## Deploying an OS

Function

The supported OSs are as follows:

* Windows: Windows 2012 R2 and Windows 2016
* Linux: CenteOS series, Red Hat series, and SUSE series
* VMware: VMware 5.5, VMware 6.0, and VMware 6.5



Ensure that the RAID configuration is complete on the server where the OS is to be installed

### ServiceCD 2.0 Mode

Parameter Configuration

* The **/etc/ansible/ansible\_ibmc/configFile/deployCfg/ServiceCD/deployConfig.ini** file is used to configure the OS type, OS image file path, and ServiceCD 2.0 image file path. The OS type must be in the following format (all supported OS types are listed):

;WINDOWS   
;Win2008\_R2\_x64   
;Win2012\_x64   
;Win2012\_R2\_x64   
;Win2016\_x64   
;UBUNTU   
;Ubuntu14.04\_x64   
;Ubuntu16.04\_x64   
   
[config]   
osType = Win2012\_R2\_x64   
osImg = nfs://192.168.2.241/NFSshare/Windows2012R2.iso   
serviceImg = nfs://192.168.2.241/NFSshare/ServiceCD133.iso

For details about the OS deployment templates in the **/etc/ansible/ansible\_ibmc/configFile/deployCfg/ServiceCD/** directory, see Table 4-1.



If sensitive data such as passwords are involved, you are advised to encrypt the ***OS*.xml** file by using the ansible-vault encryption method. For details, see <http://docs.ansible.com/ansible/latest/vault.html>.

OS deployment templates

| File | Supported OS Type | Supported Parameter |
| --- | --- | --- |
| CentOS.xml | CentOS series | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, whether to install the iBMA, and RPM package |
| RedHat.xml | Red Hat series | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, whether to install the iBMA, and RPM package |
| SLES11SP1\_64.xml | SLES 11 SP 1 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, whether to install the iBMA, and RPM package |
| SLES11SP3\_64.xml | SLES 11 SP 3 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, whether to install the iBMA, and RPM package |
| VM5.0.xml | VMware 5.0 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, and whether to install the iBMA |
| VM5.5.xml | VMware 5.5 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, and whether to install the iBMA |
| VM6.0.xml | VMware 6.0 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, and whether to install the iBMA |
| VM6.5.xml | VMware 6.5 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, and whether to install the iBMA |
| win2008r2.xml | Windows 2008 R2 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, and whether to install the iBMA |
| win2012r2.xml | Windows 2012 R2 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, and whether to install the iBMA |
| win2012.xml | Windows 2012 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, and whether to install the iBMA |
| win2016.xml | Windows 2016 | Initial password, host name, owner name, organization name, deployment location, partition, language, time zone, and whether to install the iBMA |

* Modify the **/etc/ansible/ansible\_ibmc/playbooks/deployOsByServiceCD.yml** file, including hosts (configuration object) and command (path of the **deployConfig.ini** file). Note that the file can be read by using only the CIFS and NFS protocols currently.

---   
   
- hosts: myhosts   
 name: deploy os   
 gather\_facts: no   
 tasks:   
 - name: deploy os   
 local\_action: >   
 ibmc category=DeployOS command='/etc/ansible/ansible\_ibmc/configFile/deployCfg/ServiceCD/deployConfig.ini' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run deployment commands.

[root@localhost playbooks]# ansible-playbook deployOsByServiceCD.yml   
PLAY [deploy os] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
TASK [deploy os] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
ok: [host7.domain.com -> localhost]   
PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
host7.domain.com : ok=1 changed=0 unreachable=0 failed=0

Query historical deployment process log.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/log/deployOsLog.log   
[2018-05-18 22:11:34 INFO ] (deployOs.py:344)- 192.168.4.7 -- ServiceCD133.iso;rhel73.iso;/etc/ansible/ansible\_ibmc/configFile/deployCfg/ServiceCD/RedHat.xml   
[2018-05-18 22:11:37 INFO ] (deployOs.py:366)- 192.168.4.7 -- read for bmc info:operator:eSight;osType:RHEL7U3\_x64   
[2018-05-18 22:11:37 INFO ] (deployOs.py:370)- 192.168.4.7 -- is connect:False   
[2018-05-18 22:11:37 INFO ] (deployOs.py:379)- 192.168.4.7 -- set boot device to CD!   
[2018-05-18 22:11:39 INFO ] (deployOs.py:386)- 192.168.4.7 -- bootdevice is:Cd   
[2018-05-18 22:11:39 INFO ] (deployOs.py:237)- 192.168.4.7 -- mount ServiceCD133.iso successful   
[2018-05-18 22:11:49 INFO ] (deployOs.py:400)- 192.168.4.7 -- reboot system successfully!   
[2018-05-18 22:12:09 INFO ] (deployOs.py:415)- 192.168.4.7 -- loop:1 ret: operator:eSight;osType:RHEL7U3\_x64   
[2018-05-18 22:12:30 INFO ] (deployOs.py:415)- 192.168.4.7 -- loop:2 ret: operator:eSight;osType:RHEL7U3\_x64

Query historical deployment records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/deployOsReport.log   
[2018-01-25 07:06:36 INFO ] (deployOs.py:786)- 192.168.3.114 -- install OS(Win2012\_R2\_x64) successfully!

### Smart Provisioning Mode

This mode supports only V5 servers. For details about OSs that can be deployed, see the [*FusionServer Tools V200R002C10 Smart Provisioning User Guide*](http://support.huawei.com/enterprise/en/doc/EDOC1100002509?idPath=7919749|9856522|9856629|21015513).

Parameter Configuration

The **/etc/ansible/ansible\_ibmc/configFile/deployCfg/SP/deployConfig.ini** file is used to configure the OS image file path and path of the **OS.json** file.

; This is the config file of OS deployment for Huawei Smart Provision Tools.   
   
[ConfigOS]   
# Input the os image file path on NFS server   
osImg = nfs://192.168.2.241/NFSshare/Windows2012R2.iso   
   
# Input the os config json file path on "/etc/ansible/ansible\_ibmc/configFile/SP"   
osConfig = /etc/ansible/ansible\_ibmc/configFile/deployCfg/SP/Win2012\_R2.json

For details about the OS deployment **.json** templates in the **/etc/ansible/ansible\_ibmc/configFile/deployCfg/SP/** directory, see Table 4-2. All supported OS types are listed



If sensitive data such as passwords are involved, you are advised to encrypt the ***OS*.json** file by using the ansible-vault encryption method. For details, see <http://docs.ansible.com/ansible/latest/vault.html>.

OS deployment templates

| File | Supported OS Type | Supported Parameter |
| --- | --- | --- |
| CentOS7U3.json | CentOS series | Time zone, keyboard language, whether to check firmware, manual partitioning, automatic partitioning, automatic installation position specifying, built-in software installation, and network configuration (device: NIC name or MAC address matching; IPv4 address setting; IPv6 address setting; DNS server address setting) |
| RHEL7U3.json | Red Hat series | Time zone, keyboard language, whether to check firmware, manual partitioning, automatic partitioning, automatic installation position specifying, built-in software installation, and network configuration (device: NIC name or MAC address matching; IPv4 address setting; IPv6 address setting; DNS server address setting) |
| SLES12SP3.json | SLES series | Time zone, keyboard language, whether to check firmware, manual partitioning, automatic partitioning, automatic installation position specifying, built-in software installation, and network configuration (device: NIC name or MAC address matching; IPv4 address setting; IPv6 address setting; DNS server address setting) |
| ESXi6.5.json | VMware ESXi series | Time zone, keyboard language, whether to check firmware, manual partitioning, automatic partitioning, automatic installation position specifying, built-in software installation, and network configuration (device: NIC name or MAC address matching; IPv4 address setting; IPv6 address setting; DNS server address setting) |
| Win2012\_R2.json | Windows series | Time zone, keyboard language, whether to check firmware, manual partitioning, automatic partitioning, automatic installation position specifying, built-in software installation, and network configuration (device: NIC name or MAC address matching; IPv4 address setting; IPv6 address setting; DNS server address setting) |

Modify the **/etc/ansible/ansible\_ibmc/playbooks/deployOsBySp.yml** file, including hosts (configuration object) and command (path of the **deployConfig.ini** file). Note that the file can be read by using only the CIFS and NFS protocols currently.

---   
   
- hosts: myhosts   
 name: deploy os   
 gather\_facts: no   
 tasks:   
 - name: deploy os   
 local\_action: >   
 ibmc category=DeployOSBySp command='/etc/ansible/ansible\_ibmc/configFile/deployCfg/SP/deployConfig.ini' ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

Run deployment commands.

[root@localhost playbooks]# ansible-playbook deployOsBySp.yml   
 PLAY [deploy os] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
 TASK [deploy os] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
 ok: [host7.domain.com -> localhost]   
 PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
 host7.domain.com : ok=1 changed=0 unreachable=0 failed=0

Query historical deployment process log.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/log/deploySPOsLog.log   
[2018-05-18 21:50:10 INFO ] (deployOsBySp.py:57)- 192.168.3.114 -- the iBMC version is greater than 3.00, match the redfish interface requirement, continue!   
   
[2018-05-18 21:50:10 INFO ] (deployOsBySp.py:98)- 192.168.3.114 -- the SP version is greater than 1.09, match the redfish interface requirement!   
   
[2018-05-18 21:50:10 INFO ] (deployOsBySp.py:57)- 192.168.3.7 -- the iBMC version is greater than 3.00, match the redfish interface requirement, continue!   
   
[2018-05-18 21:50:11 INFO ] (deployOsBySp.py:423)- 192.168.3.114 --vmm is connected:False   
[2018-05-18 21:50:11 INFO ] (deployOsBySp.py:98)- 192.168.3.7 -- the SP version is greater than 1.09, match the redfish interface requirement!   
   
   
==> pmLog.log <==   
[2018-05-18 21:50:11 INFO ] (powerManage.py:82)- 192.168.3.114 -- set system power off successful!   
   
   
==> deploySPOsLog.log <==   
[2018-05-18 21:50:11 INFO ] (deployOsBySp.py:433)- 192.168.3.114 -- Power off Operation System successfully!   
[2018-05-18 21:50:11 INFO ] (deployOsBySp.py:423)- 192.168.3.7 --vmm is connected:False   
   
==> pmLog.log <==   
[2018-05-18 21:50:12 INFO ] (powerManage.py:82)- 192.168.3.7 -- set system power off successful!   
   
   
==> deploySPOsLog.log <==   
[2018-05-18 21:50:12 INFO ] (deployOsBySp.py:433)- 192.168.3.7 -- Power off Operation System successfully!   
[2018-05-18 21:50:22 INFO ] (deployOsBySp.py:275)- 192.168.3.114 -- post os config parament successfully!   
   
[2018-05-18 21:50:22 INFO ] (deployOsBySp.py:275)- 192.168.3.7 -- post os config parament successfully!   
   
[2018-05-18 21:50:23 INFO ] (deployOsBySp.py:314)- 192.168.3.114 -- set SP result finished successful!   
   
[2018-05-18 21:50:23 INFO ] (deployOsBySp.py:314)- 192.168.3.7 -- set SP result finished successful!   
   
[2018-05-18 21:50:24 INFO ] (deployOsBySp.py:353)- 192.168.3.114 -- set boot device as SP successful!   
   
[2018-05-18 21:50:24 INFO ] (deployOsBySp.py:353)- 192.168.3.7 -- set boot device as SP successful!   
   
[2018-05-18 21:50:24 INFO ] (deployOsBySp.py:192)- 192.168.3.114 -- mount rhel73.iso successful   
[2018-05-18 21:50:25 INFO ] (deployOsBySp.py:192)- 192.168.3.7 -- mount rhel73.iso successful   
   
==> pmLog.log <==   
[2018-05-18 21:50:34 INFO ] (powerManage.py:60)- 192.168.3.114 -- set system power on successful!   
   
   
==> deploySPOsLog.log <==   
[2018-05-18 21:50:34 INFO ] (deployOsBySp.py:481)- 192.168.3.114 -- reboot system successfully!   
   
==> pmLog.log <==   
[2018-05-18 21:50:36 INFO ] (powerManage.py:60)- 192.168.3.7 -- set system power on successful!   
   
   
==> deploySPOsLog.log <==   
[2018-05-18 21:50:36 INFO ] (deployOsBySp.py:481)- 192.168.3.7 -- reboot system successfully!   
[2018-05-18 21:50:40 INFO ] (deployOsBySp.py:385)- 192.168.3.114: SP Status is Init   
   
[2018-05-18 21:50:40 INFO ] (deployOsBySp.py:501)- 192.168.3.114 -- loopInstall:1 SPStatus:Init, OSProgress:, OSStatus:, OSStep:, OSErrorInfo:   
   
[2018-05-18 21:50:40 INFO ] (deployOsBySp.py:503)- 192.168.3.114 -- SP is initial, please wait!   
[2018-05-18 21:50:45 INFO ] (deployOsBySp.py:385)- 192.168.3.7: SP Status is Init

Query historical deployment records.

[root@localhost playbooks]# cat /etc/ansible/ansible\_ibmc/report/deploySPOsReport.log   
[2018-05-11 18:16:04 INFO ] (deployOsBySp.py:274)- 192.168.3.7 -- post os config parament successfully!   
[2018-05-11 18:16:05 INFO ] (deployOsBySp.py:274)- 192.168.3.114 -- post os config parament successfully!   
[2018-05-11 18:53:08 INFO ] (deployOsBySp.py:509)- 192.168.3.114 -- os install successfully   
[2018-05-11 19:01:03 INFO ] (deployOsBySp.py:509)- 192.168.3.7 -- os install successfully

# FAQ

[5.1 How Do I Configure Ansible So That Multiple Users in the System Can Use It?](#_EN-US_TOPIC_0195360757)

## How Do I Configure Ansible So That Multiple Users in the System Can Use It?

Create an Ansible user and group.

**useradd Ansible**

Modify the owner of the Ansible directory.

**cd /etc/ansible**

**Chown -R Ansible:Ansible ansible\_ibmc**

Modify the permission on the Ansible directory.

**cd /etc/ansible**

**chmod -R 775 ansible\_ibmc**

Add another user (for example, testansible) to group Ansible.

**useradd -g Ansible testansible**

----End