|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
| **Huawei Ansible Plug-in** | | |
| **User Guide** | | |
| **Issue** | **02** | |
| **Date** | **2018-09-17** | |
|  | | | | |
|  |  | |  |  |

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, may result in minor or moderate injury. |
|  | Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results.  NOTICE is used to address practices not related to personal injury. |
|  | Calls attention to important information, best practices and tips.  NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration. |

Change History

| Issue | Date | Description |
| --- | --- | --- |
| 02 | 2018-06-29 | Modified 4.10 Configuring the RAID. |
| 01 | 2018-04-26 | This issue is the first official release. |

Contents

[Preface ii](#_Toc525660229)

[1 Introduction 1](#_Toc525660230)

[2 Installing and Uninstalling the Huawei Ansible Plug-in 2](#_Toc525660231)

[2.1 Environment Dependency 2](#_Toc525660232)

[2.1.1 Software Dependency 2](#_Toc525660233)

[2.1.2 Hardware Dependency 2](#_Toc525660234)

[2.1.3 Version Mapping 2](#_Toc525660235)

[2.2 Installing the Huawei Ansible Plug-in 2](#_Toc525660236)

[2.3 Uninstalling the Huawei Ansible Plug-in 3](#_Toc525660237)

[3 Configuring the Huawei Ansible Plug-in 4](#_Toc525660238)

[3.1 Modifying the /etc/ansible/ansible.cfg File 4](#_Toc525660239)

[3.2 Configuring the /etc/ansible/hosts File 4](#_Toc525660240)

[3.3 Configuring the /group\_vars/myhosts File 4](#_Toc525660241)

[4 Using the Huawei Ansible Plug-in 6](#_Toc525660242)

[4.1 Querying Information (Generating a .csv File) 6](#_Toc525660243)

[4.2 Querying the Health Status (Generating a .csv File) 7](#_Toc525660244)

[4.3 Configuring the Boot Device 8](#_Toc525660245)

[4.4 Managing the Power Supply 9](#_Toc525660246)

[4.5 Configuring the iBMC 11](#_Toc525660247)

[4.6 Configuring the NTP 12](#_Toc525660248)

[4.7 Configuring the SNMP Trap 13](#_Toc525660249)

[4.8 Importing and Exporting the Profile File 15](#_Toc525660250)

[4.9 Upgrading the Firmware 16](#_Toc525660251)

[4.9.1 BMC Mode 16](#_Toc525660252)

[4.9.2 Smart Provisioning Mode 18](#_Toc525660253)

[4.10 Configuring the RAID 19](#_Toc525660254)

[4.10.1 Querying RAID Configuration 19](#_Toc525660255)

[4.10.2 Deleting RAID Group 22](#_Toc525660256)

[4.10.3 Creating RAID Group 23](#_Toc525660257)

[4.10.4 Modifying RAID Configuration 24](#_Toc525660258)

[4.11 Deploying an OS 27](#_Toc525660259)

[4.11.1 ServiceCD 2.0 Mode 27](#_Toc525660260)

[4.11.2 Smart Provisioning Mode 30](#_Toc525660261)

# 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 boot devices, PSUs, iBMCs, NTP, SNMP traps, and RAIDs of servers.
* 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

## 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 3.00 or later is supported.
* Only Smart Provisioning V110 or later is supported.
* Only ServiceCD 2.0 V133 or later, which can be downloaded at the [Support E](http://support.huawei.com/enterpriseproduct/docTypeNewOffering?lang=en&pid=21015513&from=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](http://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

## 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>.

----End

# Using the Huawei Ansible Plug-in

## 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.

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   
[root@localhost playbooks]#

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   
[root@localhost playbooks]#

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!   
[root@localhost playbooks]#

## 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   
[root@localhost playbooks]#

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!   
[root@localhost playbooks]#

## Configuring the iBMC

Function

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.

Parameter Configuration

Modify the **/etc/ansible/cfgBMCIP.yml** file, including the hosts (configuration object) and command (IP address;Subnet mask;Gateway) parameters. The IP address, subnet mask, and gateway must be separated by semicolons (;). If you need to configure only the subnet mask and gateway, you can set command to **;255.255.255.0;192.168.2.1** and you do not need to configure an IP address.

- 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='192.168.2.23;255.255.255.0;192.168.2.1' 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   
[root@localhost playbooks]#

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!   
[root@localhost playbooks]#

## Configuring the NTP

Function

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.

Parameter Configuration

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

[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

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   
   
[root@ansible playbooks]#

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!   
[root@ansible playbooks]#

## Configuring the SNMP Trap

Function

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.

Parameter Configuration

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

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

- hosts: myhosts   
 name: config snmp trap   
 gather\_facts: no   
 vars:   
 user: 'root'   
   
 tasks:   
   
 - name: config snmp   
 local\_action: >   
 ibmc category=CfgSnmpTrap command="{{ user }}" ibmcip={{ ibmcip }}   
 ibmcuser={{ ibmcuser }} ibmcpswd={{ ibmcpswd }}

Procedure

[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!   
[root@localhost playbooks]#

## 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!   
[root@localhost playbooks]#

## Upgrading the Firmware

### BMC Mode

Function

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

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!   
[root@localhost playbooks]#

### 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   
[root@localhost playbooks]#

### 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   
   
[root@localhost playbooks]#

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'}]   
[root@localhost playbooks]#

### 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   
   
[root@ansible playbooks]#

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'}]   
[root@localhost playbooks]#

## 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   
[root@localhost playbooks]#

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   
[root@localhost playbooks]#

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!   
[root@localhost playbooks]#

### 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=DeployOS 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   
 [root@localhost playbooks]#

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   
[root@localhost playbooks]#