

xFusion iBMC Ansible Module V2.0.9 用户指南

发布日期 2023-03-30

版权所有©版权所有人

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

前言.....	v
1 简介.....	7
2 安装和卸载 Ansible 插件	10
2.1 安装 Ansible 插件	10
2.2 卸载 Ansible 插件	11
3 配置 Ansible 插件	13
3.1 配置/etc/ansible/hosts 文件	13
3.2 配置/group_vars/myhosts 文件	14
3.3 配置 SSL 证书认证和 TLS 1.2 通讯方式.....	17
4 使用 Ansible 插件	19
4.1 查询服务器基本信息	23
4.2 管理设备启动项	23
4.2.1 查询启动项配置信息	24
4.2.2 配置启动设备	24
4.3 管理电源	26
4.3.1 查询电源状态	26
4.3.2 配置电源状态	26
4.4 管理 iBMC 用户	27
4.4.1 查询 iBMC 用户（生成 json 文件）	28
4.4.2 创建 iBMC 用户	28
4.4.3 修改 iBMC 用户	30
4.4.4 删除 iBMC 用户	32
4.5 iBMC 网络配置	33
4.5.1 查询 iBMC 网络信息（生成 json 文件）	33
4.5.2 配置 iBMC 网络信息	34
4.6 管理 NTP 服务.....	35
4.6.1 查询 NTP 服务信息.....	36
4.6.2 配置 NTP.....	36
4.7 管理 SNMP Trap 服务	38

4.7.1 查询 SNMP 服务信息（生成 json 文件）	38
4.7.2 配置 SNMP Trap	38
4.8 导入导出 Profile 文件	41
4.8.1 导入 Profile 文件	41
4.8.2 导出 Profile 文件	43
4.9 固件升级	45
4.9.1 查询固件版本信息（生成 json 文件）	45
4.9.2 升级固件	46
4.9.2.1 带外固件升级	46
4.9.2.2 带内固件升级	48
4.10 RAID 配置	50
4.10.1 查询 RAID 配置（生成 json 文件）	50
4.10.2 删除 RAID 组	51
4.10.3 创建 RAID 组	52
4.10.4 修改 RAID 配置	54
4.11 OS 部署	55
4.11.1 Smart Provisioning 方式	56
4.12 BIOS 管理	62
4.12.1 查询 BIOS 信息（生成 json 文件）	62
4.12.2 设置 BIOS 信息	63
4.12.3 恢复 BIOS 默认配置	65
4.13 日志管理	66
4.13.1 一键收集 iBMC 日志	66
4.13.2 收集 SEL 日志	68
4.13.3 清空 SEL 日志	69
4.14 通用接口	70
4.15 文件本地传输	71
4.15.1 上传本地文件	71
4.15.2 下载文件至本地	73
4.16 管理 HTTPS 服务器根证书	74
4.16.1 导入远程 HTTPS 服务器根证书	74
4.16.2 删除远程 HTTPS 服务器根证书	76
4.16.3 导入远程 HTTPS 服务器根证书的吊销列表	78
4.17 查询安全服务信息（生成 json 文件）	79
4.18 打开或关闭 HTTPS 文件服务器证书校验	80
A FAQ	82
B 获取技术支持	88
C 通讯矩阵	89

前言

概述

本文档详细的描述了如何安装和卸载 Ansible 插件、以及如何使用插件实现查看服务器的信息和健康状态查询、配置、部署、固件升级等功能。





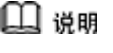
读者对象

本指南主要适用于以下工程师：

- 技术支持工程师
- 系统维护工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	用于警示紧急的危险情形，若不可避免，将会导致人员死亡或严重的人身伤害。
 警告	用于警示潜在的危险情形，若不可避免，可能会导致人员死亡或严重的人身伤害。
 注意	用于警示潜在的危险情形，若不可避免，可能会导致中度或轻微的人身伤害。
 须知	用于传递设备或环境安全警示信息，若不可避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “注意”不涉及人身伤害。
 说明	用于突出重要/关键信息、最佳实践和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。

修改记录

文档版本	发布日期	修改说明
01	2023-03-30	第一次正式发布。

1 简介

功能介绍

Ansible 插件是一个集成在 Ansible 软件中的管理服务器的插件，通过 Redfish 接口对接 iBMC，该插件可以实现对服务器查询、配置、部署、升级等功能。

具体主要支持以下功能：

- 查询服务器基本信息及健康状态
- 配置服务器的启动设备
- 管理服务器的电源
- 管理 iBMC 用户
- 查询和配置 iBMC 网络信息
- 查询和配置 NTP 服务
- 查询和配置 SNMP 服务
- 导入或导出服务器的 Profile 文件
- 升级服务器带外和带内固件
- 查询和配置 RAID
- 部署服务器的 OS（支持 ServiceCD2.0 和 Smart Provisioning 方式）
- 管理 BIOS（包括查询和设置 BIOS 信息、恢复 BIOS 默认配置）
- 管理日志（包括一键收集 iBMC 日志、收集 SEL 日志、清空 SEL 日志）
- 提供通用公共接口
- 上传本地文件
- 下载文件至本地
- 管理 HTTPS 服务器根证书
- 查询安全服务信息
- 打开或关闭 HTTPS 文件服务器证书校验

须知

Ansible 插件不涉及用户个人数据的采集和处理。

Ansible 插件支持的服务器

架构	类型	型号
x86	机架服务器	RH2288H V3
		2488 V5
		2288H V5
		1288H V6
		2288H V6
		5288 V6
x86	刀片服务器	CH121 V3
		CH242 V3 DDR4
		CH121 V5
		CH242 V5
		CH121L V5
		MM921
		CX621
		CX320
x86	高密服务器	XH622 V3
		XH321 V5
x86	异构服务器	G560 V5

版本配套关系

版本名称	配套版本
iBMC	<ul style="list-style-type: none">• V6 服务器：V3.01.12.23 及以上版本• V5 服务器：V325 及以上版本• V3 服务器：V323 及以上版本
BIOS	<ul style="list-style-type: none">• V6 服务器：V66 及以上版本• V5 服务器：V119 及以上版本• V3 服务器：V513 及以上版本

版本名称	配套版本
Smart Provisioning	V118 及以上版本，可访问 Smart Provisioning 下载
ServiceCD2.0	V139 及以上版本，可访问 FusionServer Tools 下载

2 安装和卸载 Ansible 插件

软件要求

- Ansible: 2.5.0 及以上（推荐使用 2.10 以上版本）
- Python: 2.7 及以上、3.7 及以上

说明

需要在 Python 环境上安装 requests-toolbelt 依赖包（0.9.1 及以上版本），以支撑部分功能的使用。

2.1 安装 Ansible 插件

2.2 卸载 Ansible 插件

2.1 安装 Ansible 插件

步骤 1 下载软件包并检验软件包的完整性。

1. 从 [GitHub](#) 网站获取 Ansible 插件的安装包（如“xFusion_iBMC_Ansible_Module_v2.0.8.zip”）和其对应的 sha256 校验文件（如“xFusion_Ansible.sha256.sum”）。
2. 检验 Ansible 插件软件包的完整性（Linux 操作系统下）。
 - a. 进入插件安装包和 sha256 校验文件的存放目录。
 - b. 执行 `sha256sum -c <(grep 软件包名称 sha256 校验文件名称)` 命令进行校验。
如：`sha256sum -c <(grep xFusion_iBMC_Ansible_Module_v2.0.8.zip xFusion_Ansible.sha256.sum)`
 - c. 查看校验结果是否为“OK”。
 - 是：软件包未被篡改，可使用。
 - 否：软件包已被篡改，请勿使用，请获取新软件包。

步骤 2 以 **root** 用户登录 Ansible 服务器。

步骤 3 将安装包上传到 Ansible 服务器的 root 用户目录。

步骤 4 进入 Ansible 插件安装包的所在目录。

步骤 5 执行以下命令，解压 Ansible 插件的软件包。

```
unzip xFusion_iBMC_Ansible_Module_x.x.zip
```

步骤 6 执行以下命令，进入解压后的“xFusion_iBMC_Ansible_Module”目录。

```
cd xFusion_iBMC_Ansible_Module
```

步骤 7 执行以下命令，安装 Ansible 插件。

```
python install.py
```

说明

安装成功后，在/home 目录下会增加一个 ibmc_ansible 的文件夹，该文件夹下保存了 Ansible 插件的 SSL 配置文件，以及在/home/ibmc_ansible/examples 文件夹下保存了执行 Ansible 插件时需要配置的 yml 样例文件。

----结束

2.2 卸载 Ansible 插件

步骤 1 以 **root** 用户登录 Ansible 服务器。

步骤 2 进入 Ansible 插件安装包的所在目录。

```
cd xFusion_iBMC_Ansible_Module/
```

步骤 3 执行卸载命令。

```
python uninstall.py
```

步骤 4 在提示是否保存 yml 文件时，根据实际需要输入“n”（不保存）或者“y”（保存），此处以不保存为例。

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

步骤 5 在提示是否保存日志文件和插件生成的文件时，根据实际需要输入“n”（不保存）或者“y”（保存），此处以不保存为例。

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

插件卸载成功后会提示卸载成功。

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

```
do you want to keep the log files and plug-in generation file? (y/n)
n
rm ibmc_ansible log successfully!
uninstalling xFusion_ibmc_ansible successfully!
```

----结束

3 配置 Ansible 插件

须知

本章节中的文件使用加密方式创建。加密文件、查看或设置已加密文件、以及文件加密后，如何执行配置命令的具体步骤可参见 [A.1 如何加密文件以及如何查看、编辑和执行已加密的文件](#)。

[3.1 配置/etc/ansible/hosts 文件](#)

[3.2 配置/group_vars/myhosts 文件](#)

[3.3 配置 SSL 证书认证和 TLS 1.2 通讯方式](#)

3.1 配置/etc/ansible/hosts 文件

步骤 1 执行以下命令，创建/etc/ansible。

```
mkdir /etc/ansible
```

步骤 2 执行以下命令，进入/etc/ansible 目录。

```
cd /etc/ansible
```

步骤 3 使用以下命令创建“hosts”文件。

```
vi hosts
```

步骤 4 将 myhost 信息写入“hosts”文件。

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

----结束

须知

第一列的名称（如 “host0”、“host1”）不能设置为相同的，否则执行命令时只会针对最后一台服务器执行操作。

3.2 配置/group_vars/myhosts 文件

须知

可使用 root 用户或者非 root 用户进行配置，因涉及到密码等敏感数据，推荐使用非 root 用户进行配置。

设置 “/home/用户/ibmc_ansible/examples/group_vars”（非 root 用户）或者 “/home/ibmc_ansible/examples/group_vars”（root 用户）目录下的 “myhosts” 文件中的相关参数，如 iBMC 用户名密码、SFTP/CIFS/SCP 服务用户名密码、SNMP 团体名、OS 部署时管理员用户密码等。

说明

“/home/用户/ibmc_ansible/examples/group_vars”：“用户” 为实际的非 root 用户名称，本文档以用户 “plugin” 为例。

操作步骤（非 root 用户）

步骤 1 执行以下命令切换用户权限。

```
su plugin
```

步骤 2 执行以下命令，拷贝/home 目录下的 ibmc_ansible 文件夹至/home/plugin 目录下。

```
cp -r /home/ibmc_ansible/ /home/plugin/ibmc_ansible/
```

步骤 3 执行以下命令，创建/home/plugin/ibmc_ansible/examples/group_vars 目录。

```
mkdir /home/plugin/ibmc_ansible/examples/group_vars
```

步骤 4 执行以下命令，进入/home/plugin/ibmc_ansible/examples/group_vars 目录。

```
cd /home/plugin/ibmc_ansible/examples/group_vars
```

步骤 5 使用加密命令创建 “myhosts” 文件。

```
ansible-vault create myhosts
```

须知

- 因涉及到密码等敏感数据，推荐对“myhosts”文件进行加密创建。加密创建后，执行命令时需要使用--ask-vault-pass 进行解密，具体请参见 [A.1 如何加密文件以及如何查看、编辑和执行已加密的文件](#)。
- 非加密方式创建命令为 **create myhosts**，该方式可能会导致密码等敏感数据的泄露，请谨慎使用。

步骤 6 在“myhosts”文件中写入以下内容。

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

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

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

# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
sftp_pswd: "sftp_pswd"

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

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

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

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

----结束

操作步骤（root 用户）

步骤 1 执行以下命令，创建/home/ibmc_ansible/examples/group_vars 目录。

```
mkdir /home/ibmc_ansible/examples/group_vars
```

步骤 2 执行以下命令，进入/home/ibmc_ansible/examples/group_vars 目录。

```
cd /home/ibmc_ansible/examples/group_vars
```

步骤 3 使用加密命令创建“myhosts”文件。

ansible-vault create myhosts

须知

- 因涉及到密码等敏感数据，推荐对“myhosts”文件进行加密创建。加密创建后，执行命令时需要使用--ask-vault-pass 进行解密，具体请参见 [A.1 如何加密文件以及如何查看、编辑和执行已加密的文件](#)。
- 非加密方式创建命令为 **create myhosts**，该方式可能会导致密码等敏感数据的泄露，请谨慎使用。

步骤 4 在“myhosts”文件中写入以下内容。

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

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

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

# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
sftp_pswd: "sftp_pwd"

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

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

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

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

----结束

3.3 配置 SSL 证书认证和 TLS 1.2 通讯方式

参数配置

修改 “/home/plugin/ibmc_ansible/examples/set_request_cfg.yml” 文件。

- TLS 1.2 通讯方式通过 “force_tls1_2” 参数进行配置，其为强制使用，即 “force_tls1_2” 参数默认值为 “True”。当启用状态下执行命令后出现 “import ssl.PROTOCOL_TLSv1_2 exception” 告警时，需将 “force_tls1_2” 参数设置为 “False”。

须知

关闭 force_tls1_2（设置为 “False”）会存在安全风险，请谨慎操作。

- SSL 证书认证功能通过 “verify” 和 “certify” 参数进行配置。若不设置 “certify” 参数，会使用以下默认的证书库，需先将 CA 证书导入至对应证书库下。
 - 未安装 certifi 证书库时，默认使用系统的证书库，如：
/etc/pki/tls/certs/ca-bundle.crt
 - 使用 Python 安装 certifi 证书库时，默认使用此证书库，如：
Python2 环境下：/usr/lib/python2.7/site-packages/certifi-2019.11.28-py2.7.egg/certifi/cacert.pem
Python3 环境下：/usr/local/python3/lib/python3.7/site-packages/certifi-2020.6.20-py3.7.egg/certifi/cacert.pem

须知

关闭 SSL 证书认证（“verify” 设置为 “False”）会存在安全风险，请谨慎操作。

- ciphers: Ansible 插件作为客户端与服务器建立会话时使用的加密套件。

须知

建议使用安全的加密套件，使用不安全的加密套件存在安全风险，请谨慎使用。

```
[plugin@localhost examples]$ vi set_request_cfg.yml
---
- hosts: 127.0.0.1
  connection: local
  name: set request config
  gather_facts: False
  # verify: the requests module verify server certify or not; Available values:
  True, False;
  # certify: the certify use to verify the server, if this params do not set ,
```

```
requests module will used the certificate
#which is in the certifi module or the system default certificate. Format:
/etc/pki/tls/certs/ca-bundle.crt
# force_tls1_2: force to use tls1.2 , the default value is true.
tasks:
- name: set request config
  ibmc_set_redfish_request_cfg:
    force_tls1_2: False
    verify: False
    certify:
      ciphers: "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-
RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:DHE-
RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:DHE-DSS-AES256-GCM-SHA384:ECDHE-
RSA-CHACHA20-POLY1305"
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行配置命令。

```
ansible-playbook set_request_cfg.yml
```

说明

该命令只能使用 root 用户执行，使用非 root 用户时无权限执行。

如下返回信息表示执行成功。

```
[root@localhost examples]# ansible-playbook set_request_cfg.yml

PLAY [set request config]
*****
*****
*****

TASK [set request config]
*****
*****
*****

ok: [127.0.0.1]

PLAY RECAP
*****
*****
*****

127.0.0.1 : ok=1 changed=0 unreachable=0 failed=0
skipped=0 rescued=0 ignored=0
```

4 使用 Ansible 插件

不同 Ansible 版本执行结果显示有差异，本章节命令的回显信息仅供参考，具体以实际为准。

须知

- 本章中的文件均以未加密文件为例，建议为涉及密码等敏感数据的文件进行加密。加密文件、查看或设置已加密文件、以及文件加密后，如何执行配置命令的具体步骤请参见 [A.1 如何加密文件以及如何查看、编辑和执行已加密的文件](#)。
- 本文档使用非 root 用户 “plugin” 为例进行执行操作说明，“/home/plugin/**” 中的 “plugin” 请替换为实际登录用户。
- 当 Ansible 插件配套的 iBMC 版本为做了 HTTPS 证书校验的 iBMC 版本时，Ansible 插件无法使用 HTTPS 方式的远程传输方式，建议使用其他协议。

查看帮助信息

1. 执行以下命令查看 Ansible 插件的所有命令模块名，此处以 Python3 环境为例。

```
ansible-doc -l |grep ibmc

[plugin@localhost ~]$ ansible-doc -l |grep ibmc
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-
x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177:
CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by
the OpenSSL project, please upgrade. The next version of cryptography will drop
support for it.
  utils.CryptographyDeprecationWarning,
ibmc_ansible_show_version          Show xFusion iBMC ansible modules ver...
ibmc_clear_sel_logs               Clear SEL ...
ibmc_collect_logs                 Collect iBMC ...
ibmc_collect_sel_logs             Collect iBMC SEL ...
ibmc_common_api                   Common...
ibmc_create_account               Create an ibmc ...
ibmc_create_raid                  Create vo...
ibmc_delete_account               Delete an ibmc ...
```

ibmc_delete_https_ca	delete http...
ibmc_delete_raid	Delete vo...
ibmc_deploy_os_by_service_cd	deploy os by servic...
ibmc_deploy_os_by_sp	deploy os b...
ibmc_download_file	Download f...
ibmc_get_account	Get ibmc user ...
ibmc_get_basic_info	Get server informa...
ibmc_get_bios	Get bios ...
ibmc_get_boot_device	get boot de...
ibmc_get_firmware_info_by_sp	get firmware ...
ibmc_get_ip	Get ibmc ip ...
ibmc_get_ntp	Get ntp ...
ibmc_get_power_status	get ibmc power ...
ibmc_get_raid	Get raid ...
ibmc_get_security_service_information	get security service informa...
ibmc_get_snmp_trap	Get snmp trap resource ...
ibmc_https_ca_import	import http...
ibmc_https_crl_import	import https...
ibmc_inband_fw_update	update inband firm...
ibmc_modify_account	modify an ibmc ...
ibmc_modify_raid	Modify vo...
ibmc_outband_fw_update	update outband firm...
ibmc_profile_export	export the server pro...
ibmc_profile_import	import the server pro...
ibmc_reset_bios	Reset BIOS resource attrib...
ibmc_set_bios	Set bios ...
ibmc_set_boot_device	Set boot de...
ibmc_set_https_cert_verification	set https cert verifica...
ibmc_set_ip	Set ibmc ip ...
ibmc_set_ntp	Set ntp ...
ibmc_set_power	manager server p...
ibmc_set_redfish_request_cfg	set request co...
ibmc_set_snmp_trap	Set snmp trap ...
ibmc_upload_file	upload ...

2. 执行以下命令查看各个命令模块的帮助信息。

ansible-doc 命令模块名

如 **ansible-doc ibmc_get_account**

```
[plugin@localhost ~]$ ansible-doc ibmc_get_account
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-
x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177:
CryptodgraphyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by
the OpenSSL project, please upgrade. The next version of cryptography will drop
support for it.
  utils.CryptographyDeprecationWarning,
> IBMC_GET_ACCOUNT (/usr/local/python3/lib/python3.7/site-packages/ansible-
2.9.9-py3.7.egg/ansible/modules/i

Get ibmc user info

* This module is maintained by The Ansible Community
OPTIONS (= is mandatory):

= ibmc_ip
  iBMC IP address
```

```
[Default: None]

= ibmc_pswd
    iBMC user password used for authentication
    [Default: None]

= ibmc_user
    iBMC user name used for authentication
    [Default: None]

METADATA:
    status:
    - preview
    supported_by: community

EXAMPLES:

...skipping...
> IBMC_GET_ACCOUNT    (/usr/local/python3/lib/python3.7/site-packages/ansible-
2.9.9-py3.7.egg/ansible/modules/i

    Get ibmc user info

    * This module is maintained by The Ansible Community
OPTIONS (= is mandatory):

= ibmc_ip
    iBMC IP address
    [Default: None]

= ibmc_pswd
    iBMC user password used for authentication
    [Default: None]

= ibmc_user
    iBMC user name used for authentication
    [Default: None]

METADATA:
    status:
    - preview
    supported_by: community

EXAMPLES:

- name: get ibmc account
  ibmc_get_account:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
```

查看插件版本信息

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行以下命令查看版本信息。

ansible-playbook -v show_ibmc_ansible_version.yml

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

PLAY [show xFusion iBMC ansible modules version]
*****

TASK [show xFusion iBMC ansible modules version]
*****
ok: [127.0.0.1] => {"changed": false, "msg": "xFusion iBMC ansible modules
version is 2.0.8"}

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

- 4.1 查询服务器基本信息
- 4.2 管理设备启动项
- 4.3 管理电源
- 4.4 管理 iBMC 用户
- 4.5 iBMC 网络配置
- 4.6 管理 NTP 服务
- 4.7 管理 SNMP Trap 服务
- 4.8 导入导出 Profile 文件
- 4.9 固件升级
- 4.10 RAID 配置
- 4.11 OS 部署
- 4.12 BIOS 管理
- 4.13 日志管理
- 4.14 通用接口

- [4.15 文件本地传输](#)
- [4.16 管理 HTTPS 服务器根证书](#)
- [4.17 查询安全服务信息（生成 json 文件）](#)
- [4.18 打开或关闭 HTTPS 文件服务器证书校验](#)

4.1 查询服务器基本信息

功能介绍

- 支持查询服务器 BMC 版本、BIOS 版本、CPLD 版本、Smart Provisioning 版本、序列号、资产标签、服务器型号、服务器健康状态、内存信息和健康状态、CPU 信息和健康状态、硬盘信息、健康状态信息。
- 默认生成 json 文件，若需生成 CSV 文件，在执行查询命令之前，需先配置 get_basic_info.yml 文件中的“csv_format”为“True”。

操作步骤

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。
cd /home/plugin/ibmc_ansible/examples
2. 执行查询服务器基本信息命令。

ansible-playbook get_basic_info.yml

如下返回信息表示命令执行成功。

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

PLAY [get bmc basic info]
*****

TASK [get bmc basic info]
*****

ok: [host0.domain.com]

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

查询信息后生成的文件（如“172.26.100.9_BasicInfo.json”）默认保存在“/home/plugin/ansible_ibmc/report/basic_info”目录中，建议导出文件后再查看。

4.2 管理设备启动项

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

支持查询和设置启动设备、启动参数的使能状态、启动模式。

4.2.1 查询启动项配置信息

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行查询启动项配置信息命令。

ansible-playbook get_boot_device.yml

说明

若需直接在返回信息中查看查询结果，可执行 **ansible-playbook -vv get_boot_device.yml** 命令进行查询。

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook get_boot_device.yml

PLAY [get boot device]
*****

TASK [get boot device]
*****
ok: [host0.domain.com]

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

查询结果可在 “/home/plugin/ansible_ibmc/report” 目录下的 “ansibleibmc.report” 文件中进行检查，如 3 所示。

3. 执行以下命令查看查询结果。

cd /home/plugin/ansible_ibmc/report

cat ansibleibmc.report

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

4.2.2 配置启动设备

参数配置

修改 “/home/plugin/ibmc_ansible/examples/set_boot_device.yml” 文件。


```
[plugin@localhost examples]$ vi set_boot_device.yml
---
- hosts: myhosts
  connection: local
  name: set boot device
  gather_facts: False

# boot_target: Current boot device, Available values: Cd, None, Pxe, Floppy, Hdd,
BiosSetup.
# boot_enabled: Whether the boot settings are effective, Available values: Disabled,
Once, Continuous.
# boot_mode: Boot mode, Available values: UEFI, Legacy.

tasks:
- name: set boot device
  ibmc_set_boot_device:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    boot_target: "Cd"
    boot_enabled: "Once"
    boot_mode: "Legacy"
```

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行配置启动设备命令。

ansible-playbook set_boot_device.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook set_boot_device.yml

PLAY [set boot device]
*****
*****

TASK [set boot device]
*****
*****

ok: [host9]

PLAY RECAP
*****
*****

host9                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.3 管理电源

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

支持查询和设置电源状态。

4.3.1 查询电源状态

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行查询电源状态命令。

```
ansible-playbook get_power_status.yml
```

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook get_power_status.yml

PLAY [get ibmc os power status]
*****

TASK [get ibmc os power status]
*****
ok: [host0.domain.com]

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

查询结果保存在 “/home/plugin/ansible_ibmc/report” 目录下的
“ansibleibmc.report” 文件中。

3. 执行以下命令查看查询结果。

```
cd /home/plugin/ansible_ibmc/report
```

```
cat ansibleibmc.report
```

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

4.3.2 配置电源状态

参数配置

修改 “/home/plugin/ibmc_ansible/examples/set_power.yml” 文件。

```
[plugin@localhost examples]$ vi set_power.yml
---
```

```
- hosts: myhosts
  connection: local
  name: power manager
  gather_facts: False
#power_cmd: Available values:"poweron" "poweroff" "forcerestart"
"gracefulshutdown" "forcepowercycle" "nmi"
  tasks:
    - name: power manager
      ibmc_set_power:
        ibmc_ip: "{{ ibmc_ip }}"
        ibmc_user: "{{ ibmc_user }}"
        ibmc_pswd: "{{ ibmc_pswd }}"
      power_cmd: "poweron"
```

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行配置电源状态命令。

ansible-playbook set_power.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook set_power.yml

PLAY [power manager]
*****
*****
*****

TASK [power manager]
*****
*****
*****

ok: [host9]

PLAY RECAP
*****
*****
*****

host9                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.4 管理 iBMC 用户

功能介绍

支持查询、创建、修改和删除 iBMC 用户。

4.4.1 查询 iBMC 用户（生成 json 文件）

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行查询 iBMC 用户命令。

ansible-playbook get_account.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook get_account.yml

PLAY [get ibmc Account]
*****

TASK [get ibmc Account]
*****

ok: [host0.domain.com]

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

查询信息后生成的 json 文件（如 “172.26.100.9_AccountInfo.json”）默认保存在 “/home/plugin/ansible_ibmc/report/account_info” 目录中，建议导出 json 文件后再查看。

4.4.2 创建 iBMC 用户

参数配置

- 修改 “/home/plugin/ibmc_ansible/examples/group_vars/myhosts” 文件中的 “account_user”（新建用户名）和 “account_pswd”（新建用户的密码）参数。

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

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

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

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

# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
```

```
sftp_pswd: "sftp_pswd"

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

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

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

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

- 修改 “/home/plugin/ibmc_ansible/examples/create_account.yml” 文件。

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

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行创建 iBMC 用户命令。

```
ansible-playbook create_account.yml
```

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook create_account.yml

PLAY [create ibmc Account]
*****
*****
*****

TASK [create ibmc Account]
*****
*****
*****

ok: [host9]
```

```
PLAY RECAP
*****
*****
*****
host9                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.4.3 修改 iBMC 用户

参数配置

- 修改 “/home/plugin/ibmc_ansible/examples/group_vars/myhosts” 文件中的 “account_user”（修改后的用户名）和 “account_pswd”（修改后的用户密码）参数。

```
[plugin@localhost examples]$ vi
/home/plugin/ibmc_ansible/examples/group_vars/myhosts
---
# Here we define global variables for our server group, but if some servers
# require custom values place these variables in /etc/ansible/hosts to override
# for each individual host

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

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

# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
sftp_pswd: "sftp_pwd"

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

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

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

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

- 修改 “/home/plugin/ibmc_ansible/examples/modify_account.yml” 文件。

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

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

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行修改 iBMC 用户命令。

ansible-playbook modify_account.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook modify_account.yml

PLAY [modify ibmc Account]
*****
*****
*****

TASK [modify ibmc Account]
*****
*****
*****

ok: [host9]

PLAY RECAP
*****
*****
```

```
*****
host9                        : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.4.4 删除 iBMC 用户

参数配置

修改 “/home/plugin/ibmc_ansible/examples/delete_account.yml” 文件中的
“delete_account”（需删除的用户名）参数。

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

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

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。
cd /home/plugin/ibmc_ansible/examples
2. 执行删除 iBMC 用户命令。

ansible-playbook delete_account.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook delete_account.yml

PLAY [delete ibmc Account]
*****
*****
*****

TASK [delete ibmc Account]
*****
*****
*****

ok: [host9]

PLAY RECAP
*****
*****
*****

host9                        : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```


4.5 iBMC 网络配置

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

- 支持查询 iBMC 网络信息。
- 支持单独修改 IP 使能模式或 IP 地址信息。

须知

- 不能同时修改 IP 使能模式和 IP 地址信息，否则会出现修改不成功的现象。
- IPv4 使能模式和 IPv6 使能模式之间不能相互跳转，否则会造成服务器无法连接的问题。

4.5.1 查询 iBMC 网络信息（生成 json 文件）

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。
cd /home/plugin/ibmc_ansible/examples
2. 执行查询 iBMC 网络信息命令。

ansible-playbook get_ibmc_ip.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook get_ibmc_ip.yml
```

```
PLAY [get ibmc ip]
```

```
*****
```

```
TASK [get ibmc ip]
```

```
*****
```

```
ok: [host0.domain.com]
```

```
PLAY RECAP
```

```
*****
```

```
*
```

```
host0.domain.com      : ok=1    changed=0    unreachable=0    failed=0
```

```
skipped=0    rescued=0    ignored=0
```

查询信息后生成的 json 文件（如 “172.26.100.9_iBMCIPInfo.json”）默认保存在 “/home/plugin/ansible_ibmc/report/ibmc_ip” 目录中，建议导出 json 文件后再查看。

4.5.2 配置 iBMC 网络信息

参数配置

修改 “/home/plugin/ibmc_ansible/examples/set_ibmc_ip.yml” 文件。

```
[plugin@localhost examples]$ vi set_ibmc_ip.yml
---
- hosts: myhosts
  connection: local
  name: set ibmc ip
  gather_facts: False

# target_bmc_ip: ibmc_ip that you specify to set network information, and you can
# only choose from the group of hosts.
# ip_version: Whether IPv4/IPv6 is enabled, Available values: IPv4, IPv6,
# IPv4AndIPv6.
# ipv4_addr: IPv4 address info.
#   # address: IPv4 address.
#   # subnet_mask: Subnet mask of the IPv4 address.
#   # gateway: Gateway of the IPv4 address.
#   # address_origin: How the IPv4 address is allocated. Available values: Static,
#   DHCP.
# ipv6_addr: IPv6 address info.
#   # address: IPv6 address.
#   # prefix_length: Prefix length of the IPv6 address, must be an integer, value
#   range: 0 to 128.
#   # address_origin: How the IPv6 address is allocated. Available values: Static,
#   DHCPv6.
# ipv6_gateway: IPv6 gateway address of the iBMC network port.
# hostname: iBMC HostName. Contains a maximum of 64 characters, including only
# letters, digits, and hyphens(-).
# Cannot start or end with a hyphen.
# domain_name: Domain name. Contains a maximum of 67 characters. The format of FQDN
# is hostname.domain_name.
# For example, if hostname is "testhostname" and domain_name is "ibmc.com", then
# FQDN is "testhostname.ibmc.com".

tasks:
- name: set ibmc ip
  ibmc_set_ip:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    target_bmc_ip: "192.168.3.11"
    ip_version: "IPv4AndIPv6"
    ipv4_addr:
      - address: "192.168.2.10"
        subnet_mask: "255.255.0.0"
        gateway: "192.168.0.1"
        address_origin: "Static"
    ipv6_addr:
      - address: "fc00:192::10"
        prefix_length: 7
        address_origin: "Static"
```

```
ipv6_gateway: "fc00:192::1"
hostname: "testhostname"
domain_name: "ibmc.com"
```

说明

目前仅支持配置单台设备 iBMC 网络信息，不支持批量配置。

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行配置 iBMC 网络信息命令。

ansible-playbook set_ibmc_ip.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook set_ibmc_ip.yml

PLAY [set ibmc ip]
*****
*****
*****

TASK [set ibmc ip]
*****
*****
*****

ok: [host9]

PLAY RECAP
*****
*****
*****

host9                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.6 管理 NTP 服务

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

- 支持查询 NTP 服务信息。
- 支持配置 NTP 服务使能、首选/备选 NTP 服务器地址、服务器身份认证使能、NTP 地址模式（IPv4/IPv6/Static）和最小/最大轮询间隔值。

4.6.1 查询 NTP 服务信息

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行查询 NTP 服务信息命令。

ansible-playbook get_ntp.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook get_ntp.yml

PLAY [get ntp]
*****

TASK [get ntp]
*****

ok: [host0.domain.com]

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

查询结果保存在 “/home/plugin/ansible_ibmc/report” 目录下的
“ansibleibmc.report” 文件中。

3. 执行以下命令查看查询结果。

cd /home/plugin/ansible_ibmc/report

cat ansibleibmc.report

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

4.6.2 配置 NTP

参数配置

修改 “/home/plugin/ibmc_ansible/examples/set_ntp.yml” 文件。

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

# service_enabled: Enable or disable bmc ntp service, Available values: True, False.
# pre_ntp_server: Config preferred NtpServer, you can enter ipv4 ipv6 or domain
name, NTP Server will be blanked when set to an empty string.
```

```
# alt_ntp_server: Config alternate NtpServer, you can enter ipv4 ipv6 or domain
name, NTP Server will be blanked when set to an empty string.
# server_auth_enabled: Enable or disable Server Authentication service, Available
values: True, False.
# ntp_address_origin: Config Ntp Address Origin, Available values: IPv4, IPv6,
Static.
# min_polling_interval: Config Min Polling Interval time, must be an integer, in
3~17 and <= max_polling_interval.
# max_polling_interval: Config Max Polling Interval time, must be an integer, in
3~17 and >= min_polling_interval.

tasks:
- name: set ntp
  ibmc_set_ntp:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    service_enabled: True
    pre_ntp_server: "192.168.2.10"
    alt_ntp_server: "192.168.2.20"
    server_auth_enabled: False
    ntp_address_origin: "Static"
    min_polling_interval: 3
    max_polling_interval: 17
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行配置 NTP 命令。

```
ansible-playbook set_ntp.yml
```

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook set_ntp.yml

PLAY [set ntp]
*****
*****
*****

TASK [set ntp]
*****
*****
*****

ok: [host9]

PLAY RECAP
*****
*****
*****

host9                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.7 管理 SNMP Trap 服务

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

- 支持查询 SNMP 的 Trap 服务。
- 支持配置 Trap 功能的使能状态、TrapV3 用户名、上报模式、主机标识、团体名、告警发送级别和 Trap 服务器。

4.7.1 查询 SNMP 服务信息（生成 json 文件）

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行查询 SNMP 服务信息命令。

```
ansible-playbook get_snmp_trap.yml
```

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook get_snmp_trap.yml

PLAY [get snmp trap]
*****

TASK [get snmp trap]
*****
ok: [host0.domain.com]

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

查询信息后生成的 json 文件（如 “172.26.100.9_SNMPTrapInfo.json”）默认保存在 “/home/plugin/ansible_ibmc/report/snmp_trap” 目录中，建议导出 json 文件后再查看。

4.7.2 配置 SNMP Trap

参数配置

- 修改 “/home/plugin/ibmc_ansible/examples/set_snmp_trap.yml” 文件。

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

```
# service_enabled: Whether trap is enabled, Available values: True, False.
# trap_version: Trap version, Available values: V1, V2C, V3.
# trap_v3_user: SNMPv3 user name, valid only for trap version is V3.
# trap_mode: Trap mode, Available values: OID, EventCode, PreciseAlarm.
# trap_server_identity: Host identifier, Available values: BoardSN,
ProductAssetTag, HostName.
# alarm_severity: Severity levels of the alarm to be sent, Available values:
Critical, Major, Minor, Normal.
# trap_servers: Can set one or more trap server, When all parameters of the
trap server are empty, it indicates that the trap server is not configured.
# trap_server_enabled: Whether the trap server is enabled, Available values:
True, False.
# trap_server_address: Server address, you can enter ipv4 ipv6 or domain name.
# trap_server_port: Server port number, must be an integer, Available value
range: 1 to 65535.

tasks:
- name: set snmp trap
  ibmc_set_snmp_trap:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    community: "{{ community }}"
    service_enabled: True
    trap_version: "V3"
    trap_v3_user: "root"
    trap_mode: "OID"
    trap_server_identity: "HostName"
    alarm_severity: "Normal"
    trap_servers:
      - trap_server_enabled: True
        trap_server_address: "192.168.2.10"
        trap_server_port: 160
      - trap_server_enabled: True
        trap_server_address: "192.168.2.11"
        trap_server_port: 161
      - trap_server_enabled: False
        trap_server_address: "192.168.2.12"
        trap_server_port: 162
      - trap_server_enabled: False
        trap_server_address: "192.168.2.13"
        trap_server_port: 163
```

- 修改 “/home/plugin/ibmc_ansible/examples/group_vars/myhosts” 文件中的 “community”（团体名称）参数。

📖 说明

当 “trap_version” 配置为 “V1” 或者 “V2C” 时，可在
“/home/plugin/ibmc_ansible/examples/group_vars/myhosts” 文件中配置 “community” 参数。

```
[plugin@localhost examples]$ vi
/home/plugin/ibmc_ansible/examples/group_vars/myhosts
---
# Here we define global variables for our server group, but if some servers
```

```
# require custom values place these variables in /etc/ansible/hosts to override
# for each individual host

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

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

# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
sftp_pswd: "sftp_pswd"

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

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

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

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

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行配置 SNMP Trap 命令。

```
ansible-playbook set_snmp_trap.yml
```

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook set_snmp_trap.yml

PLAY [set snmp trap]
*****
*****
*****

TASK [set snmp trap]
*****
*****
*****
ok: [host9]

PLAY RECAP
*****
*****
*****
```



```
host9          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

4.8 导入导出 Profile 文件

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

支持 BIOS 和 iBMC 的 Profile 配置文件的导入和导出。

- 支持使用本地或远程的方式将 Profile 文件导入到服务器。
- 支持将服务器的 Profile 文件导出到本地环境或者远程路径上。

4.8.1 导入 Profile 文件

参数配置

修改 “/home/plugin/ibmc_ansible/examples/profile_import.yml” 文件。

本地导入文件时，配置的参数如下：

- **file_name**：需导入文件名称。
- **local_import**：需导入的 Ansible 环境下的本地文件路径。

远程导入文件时，配置的参数如下：

- **file_name**：需导入文件名称。
- **remote_file**：需导入的远程目录下的文件路径。
 - iBMC 临时 “/tmp” 目录。
 - 远程文件服务器目录下的文件夹，文件路径输入形式为 “文件传输协议://远程文件服务器 IP 地址/文件夹”。

支持的文件传输协议包括 sftp、https、nfs、cifs、scp。

- **file_server_user**：远程文件服务器用户名。
- **file_server_pswd**：远程文件服务器密码。

本地导入时，配置的参数如下：

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

- hosts: myhosts
  connection: local
  name: import profile
  gather_facts: False

# file_name: the file name you want to import
# local_import: local file path of the Ansible environment to be imported.
# remote_import: remote path for saving imported files. The file path can be /tmp
```

```
on the BMC; or a folder on a remote file server, the format is protocol://ip/folder
# protocols: Available values: sftp,https,nfs,cifs,scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password

tasks:
- name: import profile
  ibmc_profile_import:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    file_name: "192.168.1.1_20210318045050_profile.xml"
    local_import: "/home"
```

远程导入时，配置的参数如下：

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

- hosts: myhosts
  connection: local
  name: import profile
  gather_facts: False

# file_name: the file name you want to import
# local_import: local file path of the Ansible environment to be imported.
# remote_import: remote path for saving imported files. The file path can be /tmp
on the BMC; or a folder on a remote file server, the format is protocol://ip/folder
# protocols: Available values: sftp,https,nfs,cifs,scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password

tasks:
- name: import profile
  ibmc_profile_import:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    file_name: "192.168.1.1_20210318045050_profile.xml"
    remote_import: "sftp://192.168.1.1/data/"
    file_server_user: "{{sftp_user}}"
    file_server_pswd: "{{sftp_pswd}}"
```

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行导入 Profile 文件命令。

ansible-playbook profile_import.yml

如下返回信息表示命令执行成功。

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

```
support for it.
  utils.CryptographyDeprecationWarning,

PLAY [import profile]
*****
*****

TASK [import profile]
*****
*****

ok: [host0]

PLAY RECAP
*****
*****

host0          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

4.8.2 导出 Profile 文件

参数配置

修改 “/home/plugin/ibmc_ansible/examples/profile_export.yml” 文件。

导出文件至 Ansible 环境上时，配置的参数如下：

- **file_name**: 文件导出后的名称，可选参数。如不配置该参数，默认导出后文件名称为 “IP 地址_profile.xml”。
- **local_export**: 用于保存导出文件的 Ansible 环境下的本地路径。

导出文件至远程路径时，配置的参数如下：

- **file_name**: 文件导出后的名称，可选参数。如不配置该参数，默认导出后文件名称为 “IP 地址_profile.xml”。
- **remote_export**: 用于保存导出文件的远程路径。
 - iBMC 临时 “/tmp” 目录。
 - 远程文件服务器目录下的文件夹，文件路径输入形式为 “文件传输协议://远程文件服务器 IP 地址/文件夹”。支持的文件传输协议包括 sftp、https、nfs、cifs、scp。
- **file_server_user**: 远程文件服务器用户名。
- **file_server_pswd**: 远程文件服务器密码。

导出文件至 Ansible 环境上时，配置的参数如下：

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

- hosts: myhosts
  connection: local
  name: import profile
  gather_facts: False

# file_name: the file name you want to import
```

```
# local_import: local file path of the Ansible environment to be imported.
# remote_import: remote path for saving imported files. The file path can be /tmp
on the BMC; or a folder on a remote file server, the format is protocol://ip/folder
# protocols: Available values: sftp,https,nfs,cifs,scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password

tasks:
- name: import profile
  ibmc_profile_import:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    file_name: "192.168.1.1_20210318045050_profile.xml"
    local_export: "/home"
```

导出文件至远程路径时，配置的参数如下：

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

- hosts: myhosts
  connection: local
  name: import profile
  gather_facts: False

# file_name: the file name you want to import
# local_import: local file path of the Ansible environment to be imported.
# remote_import: remote path for saving imported files. The file path can be /tmp
on the BMC; or a folder on a remote file server, the format is protocol://ip/folder
# protocols: Available values: sftp,https,nfs,cifs,scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password

tasks:
- name: import profile
  ibmc_profile_import:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    file_name: "192.168.1.1_20210318045050_profile.xml"
    remote_export: "sftp://192.168.1.1/data/"
    file_server_user: "{{sftp_user}}"
    file_server_pswd: "{{sftp_pswd}}"
```

执行命令

1. 进入“/home/plugin/ibmc-ansible/examples”文件目录。

cd /home/plugin/ibmc-ansible/examples

2. 执行导出 Profile 文件命令。

ansible-playbook profile_import.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook profile_export.yml
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-
x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177:
```

```
CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by
the OpenSSL project, please upgrade. The next version of cryptography will drop
support for it.
  utils.CryptographyDeprecationWarning,

PLAY [export profile]
*****
*****

TASK [export profile]
*****
*****

ok: [host0]

PLAY RECAP
*****
*****

host0          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

4.9 固件升级

功能介绍

- 支持查询固件版本信息。
- 支持升级带外固件，包括 BMC、BIOS、CPLD。
- 支持 Smart Provisioning 方式的带内固件升级。

4.9.1 查询固件版本信息（生成 json 文件）

说明

- 在使用其他工具升级带内固件后，需要重新启动 Smart Provisioning，才能获取最新的固件版本信息。
- 管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。
cd /home/plugin/ibmc_ansible/examples
2. 执行查询固件版本信息命令。

ansible-playbook get_firmware_info_by_sp.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook get_firmware_info_by_sp.yml

PLAY [get firmware info by sp]
*****
*****
```

```
TASK [get firmware info by sp]
*****
*****
ok: [host9]

PLAY RECAP
*****
*****
host9          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

查询信息后生成的 json 文件（如“172.26.100.9_fwInfo.json”）默认保存在“/home/plugin/ansible_ibmc/report/inband_fw_info”目录中，建议导出 json 文件后再查看。

4.9.2 升级固件

4.9.2.1 带外固件升级

参数配置

修改“/home/plugin/ibmc_ansible/examples/update_outband_fw.yml”文件。

使用本地的升级文件时，配置的参数如下：

local_file: 本地升级文件的路径，输入形式为“目录/文件名”，如/home/cpldimage.hpm。

使用远程目录的升级文件时，配置的参数如下：

- **remote_file:** 远程目录升级文件的路径。
 - 使用 iBMC 临时目录时，必须为“/tmp”目录，而且要指定一个文件名，如/tmp/cpldimage.hpm。
 - 使用远程文件服务器目录时，文件路径输入形式为“文件传输协议://远程文件服务器 IP 地址/目录/文件名”，如 sftp://192.168.1.1/data/cpldimage.hpm。
支持的文件传输协议包括 sftp、https、nfs、cifs、scp。
- **file_server_user:** 远程文件服务器用户名。
- **file_server_pswd:** 远程文件服务器密码。

说明

- 交换模块 CX320 和 CX621 仅支持 SFTP 协议，管理模块 MM921 仅支持 SFTP 和 NFS 协议。
- 升级前请确认固件包中的 hpm 文件已上传至文件服务器相应路径下。固件包的获取方法如下：
- 登录[技术支持](#)。

1. 选择对应的服务器型号进入目录。
2. 选择“软件”页签。
3. 选择补丁版本路径。
4. 下载所需的固件包。

使用本地的升级文件时，配置如下：

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

  tasks:
    - name: update outband fw
      ibmc_outband_fw_update:
        ibmc_ip: "{{ ibmc_ip }}"
        ibmc_user: "{{ ibmc_user }}"
        ibmc_pswd: "{{ ibmc_pswd }}"
        local_file: "/home/cpldimage.hpm"
```

使用远程目录的升级文件时，配置如下：

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

  # local_file: the local firmware file needs to be upgraded, the format is
  # directory/filename, for example: /home/cpldimage.hpm.
  # remote_file: remote firmware file. The firmware file can be /tmp on the BMC, the
  # format is directory/filename, for example: /tmp/cpldimage.hpm;
  # or a firmware file on a remote file server, the format is
  # protocol://ip/folder/filename, for example: sftp://172.26.200.11/data/cpldimage.hpm.
  # protocols: Available values: sftp,https,nfs,cifs,scp
  # file_server_user: remote file server user name
  # file_server_pswd: remote file server password

  tasks:
    - name: update outband fw
      ibmc_outband_fw_update:
        ibmc_ip: "{{ ibmc_ip }}"
        ibmc_user: "{{ ibmc_user }}"
        ibmc_pswd: "{{ ibmc_pswd }}"
        remote_file: "sftp://192.168.1.1/data/cpldimage.hpm"
        file_server_user: "{{sftp_user}}"
        file_server_pswd: "{{sftp_pswd}}"
```

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行升级命令。

```
ansible-playbook update_outband_fw.yml
```

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook update_outband_fw.yml

PLAY [update outband fw]
*****
*****
*****

TASK [update outband fw]
*****
*****
*****
ok: [host9]

PLAY RECAP
*****
*****
*****
host9                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.9.2.2 带内固件升级

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

支持 Smart Provisioning 方式的带内固件升级。具体支持的服务器以及具体支持的固件升级信息请参见《[Smart Provisioning 用户指南](#)》。

参数配置

修改“/home/plugin/ibmc_ansible/examples/update_inband_fw.yml”文件中的“image_url”参数。

image_url: 升级文件的路径，输入形式为“文件传输协议://用户名:密码@服务器 IP 地址/目录/文件名”。文件传输协议包括 SFTP、HTTPS、NFS、CIFS、SCP。

说明

升级前请确认固件升级包和数字签名文件已上传至文件服务器相应路径下。固件升级包和数字签名文件请访问 [FusionServer iDriver](#) 获取。

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



```
tasks:
- name: update inband fw
  ibmc_inband_fw_update:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    image_url:
      - "sftp://172.26.200.11/data/NIC-IOM-X722-10GE_SFP-GE_Electrical-FW-3.33_0x80000f09.zip"
    file_server_user: "{{ sftp_user }}"
    file_server_pswd: "{{ sftp_pswd }}"
```

文件服务器的用户名和密码需在“/home/plugin/ibmc_ansible/examples/group_vars”目录下的“myhosts”文件中修改。

```
[plugin@localhost examples]$ vi
/home/plugin/ibmc_ansible/examples/group_vars/myhosts
---
# Here we define global variables for our server group, but if some servers
# require custom values place these variables in /etc/ansible/hosts to override
# for each individual host

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

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

# input the sftp user and password when we need to use the sftp service
sftp_user: "sftp_user"
sftp_pswd: "sftp_pwd"

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

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

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

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

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。
cd /home/plugin/ibmc_ansible/examples
2. 执行升级命令。

ansible-playbook update_inband_fw.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook update_inband_fw.yml

PLAY [update inband fw]
*****

TASK [update inband fw]
*****
ok: [host0.domain.com]

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

4.10 RAID 配置

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

- 仅支持配置带外管理的 RAID 卡。
- 支持多 RAID 卡的场景。
- 支持 RAID 卡（LSI SAS3108、Avago SAS3408iMR、Avago SAS3004iMR、Avago SAS3508）的查询、配置、修改和删除。

4.10.1 查询 RAID 配置（生成 json 文件）

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行查询 RAID 配置命令。

```
ansible-playbook get_raid.yml
```

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook get_raid.yml

PLAY [get raid]
*****

TASK [get raid]
*****
ok: [host0.domain.com]

PLAY RECAP
```

```
*****
*
host0.domain.com      : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

查询信息后生成的 json 文件（如“172.26.100.9_RAIDInfo.json”）默认保存在“/home/plugin/ansible_ibmc/report/raid”目录中，建议导出 json 文件后再查看。

4.10.2 删除 RAID 组

参数配置

修改“/home/plugin/ibmc_ansible/examples/delete_raid.yml”文件。

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

# storage_id: ID of the storage resource
# 1.Delete one RAID storage, Format: RAIDStorage+Controller_ID
# 2.Delete multiple RAID storage, Separated by commas, Format:
RAIDStorage+Controller_ID1,RAIDStorage+Controller_ID2,...
# 3.Delete all RAID storage, Format: all
# volume_id: Volume resource ID
# 1.Delete one volume, Format: LogicalDrive+Volume_ID
# 2.Delete multiple volume, Separated by commas, Format:
LogicalDrive+Volume_ID1,LogicalDrive+Volume_ID2,...
# 3.Delete all volume, Format: all

tasks:
- name: delete raid
  ibmc_delete_raid:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    storage_id: "RAIDStorage0,RAIDStorage1"
    volume_id: "LogicalDrive0,LogicalDrive1"
```

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行删除 RAID 组命令。

ansible-playbook delete_raid.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook delete_raid.yml

PLAY [delete raid]
*****
*****
```

```
*****
TASK [delete raid]
*****
*****
*****
ok: [host9]

PLAY RECAP
*****
*****
*****
host9          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

4.10.3 创建 RAID 组

参数配置

修改 “/home/plugin/ibmc_ansible/examples/create_raid.yml” 文件。

须知

不同的 RAID 卡支持可配置的参数不同，具体请参考《[V2&V3 服务器 RAID 控制卡 用户指南](#)》、《[V5 服务器 RAID 控制卡 用户指南](#)》。

```
[plugin@localhost examples]$ vi create_raid.yml
---
- hosts: myhosts
  connection: local
  name: create raid
  gather_facts: False

# storage_id: ID of the storage resource. Format: RAIDStorage+Controller_ID
# capacity_mbyte: Volume capacity, must be an integer, the size unit is MB. It is
an optional parameter
# stripe_size: Stripe size of a volume, must be an integer. It is an optional
parameter. Available values: 65536, 131072, 262144, 524288, 1048576
# cachecade_flag: Whether it is a CacheCade volume. It is an optional parameter,
Available values: True, False
# drives: Member disk list number. It is a mandatory parameter. Format: "1,2,..,N"
# volume_raid_level: RAID level of the volume. It is a mandatory parameter.
Available values: RAID0, RAID1, RAID5, RAID6, RAID10, RAID50, RAID60
# volume_name: Volume name. It is an optional parameter. A string of up to 15 bytes.
Value range: ASCII code corresponding to 0x20 to 0x7E
# df_read_policy: Default read policy of the volume. It is an optional parameter.
Available values: NoReadAhead, ReadAhead
# df_write_policy: Default write policy of the volume. It is an optional parameter.
Available values: WriteThrough, WriteBackWithBBU, WriteBack
# df_cache_policy: Default cache policy of the volume. It is an optional parameter.
Available values: CachedIO, DirectIO
# span_num: Number of spans of the volume, must be an integer. It is an optional
```

```
parameter
# 1.Set this parameter to 1 when creating a RAID0, RAID1, RAID5, or RAID6 array.
# 2.Set this parameter to a value from 2 to 8 when creating a RAID10, RAID50, or
RAID60 array.
# access_policy: Volume access policy. It is an optional parameter. Available
values: ReadWrite, ReadOnly, Blocked
# disk_cache_policy: Cache policy for member disks. It is an optional parameter.
Available values: Unchanged, Enabled, Disabled
# init_mode: Volume initialization mode. It is an optional parameter. Available
values: UnInit, QuickInit, FullInit

tasks:
- name: create raid
  ibmc_create_raid:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    volumes:
      - storage_id: "RAIDStorage0"
        capacity_mbyte: 1000
        stripe_size: 65536
        cachecade_flag: False
        drives: "0,1"
        volume_raid_level: "RAID0"
        volume_name: "volume_name"
        df_read_policy: "NoReadAhead"
        df_write_policy: "WriteThrough"
        df_cache_policy: "CachedIO"
        span_num: 1
        access_policy: "ReadWrite"
        disk_cache_policy: "Unchanged"
        init_mode: "UnInit"
```

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。
cd /home/plugin/ibmc_ansible/examples
2. 执行创建 RAID 组命令。

ansible-playbook create_raid.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook create_raid.yml

PLAY [create raid]
*****
*****
*****

TASK [create raid]
*****
*****
*****

ok: [host9]
```

```
PLAY RECAP
*****
*****
*****
host9          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

4.10.4 修改 RAID 配置

参数配置

修改 “/home/plugin/ibmc_ansible/examples/modify_raid.yml” 文件。

须知

不同的 RAID 卡支持可修改的参数不同，具体请参考《[V2&V3 服务器 RAID 控制卡 用户指南](#)》或《[V5 服务器 RAID 控制卡 用户指南](#)》。

```
[plugin@localhost examples]$ vi modify_raid.yml
---
- hosts: myhosts
  connection: local
  name: modify raid
  gather_facts: False

# storage_id: ID of the storage resource. Format: RAIDStorage+Controller_ID
# volume_id: Volume resource ID. Format: LogicalDrive+Volume_ID
# volume_name: Volume name. It is an optional parameter. A string of up to 15 bytes.
Value range: ASCII code corresponding to 0x20 to 0x7E
# df_read_policy: Default read policy of the volume. It is an optional parameter.
Available values: NoReadAhead, ReadAhead
# df_write_policy: Default write policy of the volume. It is an optional parameter.
Available values: WriteThrough, WriteBackWithBBU, WriteBack
# df_cache_policy: Default cache policy of the volume. It is an optional parameter.
Available values: CachedIO, DirectIO
# boot_enable: Whether it is the boot device. Available values: True.
# bgi_enable: Whether background initialization is enabled. Available values: True,
False.
# access_policy: Volume access policy. It is an optional parameter. Available
values: ReadWrite, ReadOnly, Blocked
# ssd_cache_enable: Whether the CacheCade volume is used as the cache. Available
values: True, False.
# disk_cache_policy: Cache policy for member disks. It is an optional parameter.
Available values: Unchanged, Enabled, Disabled

tasks:
- name: modify raid
  ibmc_modify_raid:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
```

```
ibmc_pswd: "{{ ibmc_pswd }}"
volumes:
  - storage_id: "RAIDStorage0"
    volume_id: "LogicalDrive0"
    volume_name: "volume_name"
    df_read_policy: "NoReadAhead"
    df_write_policy: "WriteThrough"
    df_cache_policy: "CachedIO"
    boot_enable: True
    bgi_enable: False
    access_policy: "ReadWrite"
    ssd_cache_enable: False
    disk_cache_policy: "Unchanged"
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行修改 RAID 配置命令。

ansible-playbook modify_raid.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook modify_raid.yml

PLAY [modify raid]
*****
*****
*****

TASK [modify raid]
*****
*****
*****

ok: [host9]

PLAY RECAP
*****
*****
*****

host9                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.11 OS 部署

📖 说明

- 管理模块 MM921、交换模块 CX320/CX621 不支持 OS 部署功能。
- 待部署 OS 的服务器需提前完成 RAID 配置。

4.11.1 Smart Provisioning 方式

Smart Provisioning 方式具体支持的服务器请参见《[Smart Provisioning 用户指南](#)》。

说明

此方式包括 “deploy_centos7u3_by_sp.yml”、“deploy_esxi65_by_sp.yml” 和 “deploy_win2012r2_by_sp.yml” 命令，以下以 “deploy_centos7u3_by_sp.yml” 为例进行说明。

参数配置

修改 “/home/plugin/ibmc_ansible/examples/deploy_centos7u3_by_sp.yml” 文件。

关于各参数的具体信息请参见[表 4-1](#)。

```
[plugin@localhost examples]$ vi deploy_centos7u3_by_sp.yml
---
- hosts: myhosts
  connection: local
  name: ibmc deploy centos7u3 by sp
  gather_facts: False
#os_img: The os image path ; Format:
protocol://Username:Password@IPaddress/Folder/image_file; Available protocol:
nfs,cifs,https
#OSType:Os type; Available values:RHEL6U9, RHEL6U10, RHEL7U3 ,RHEL7U4, RHEL7U5,
RHEL7U6, CentOS6U9, CentOS6U10, CentOS7U3, CentOS7U4, CentOS7U5, CentOS7U6, ESXi6.0,
ESXi6.5, ESXi6.7
# SLES11SP4, SLES12SP2, SLES12SP3, Ubuntu16.04, Ubuntu16.04.1, Ubuntu16.04.2,
Win2016, Win2016 Standard Desktop, Win2016 Standard Core, Win2016 Datacenter
Desktop
# Win2016 Datacenter Core, Win2012_R2, Win2012_R2 Standard Desktop, Win2012_R2
Standard Core, Win2012_R2 Datacenter Desktop, Win2012_R2 Datacenter Core
# EulerOSV2SP3
#InstallMode: OS Installation mode; Available values:Recommended, Customized
#Language: Available values: Please refer to the installation guide of the OS.
#TimeZone: Available values: Please refer to the installation guide of the OS.
#Keyboard: Available values: Please refer to the installation guide of the OS.
#BootType: Bios boot mode,This parameter is optional; Available values: UEFIBoot,
LegacyBoot, SecureBoot
#CDKey: key of the OS Installation, This parameter is optional
#RootPwd: Root user password, this parameter is mandatory;
# Windows: a sting of at least 6 characters.SUSE: a sting of at least 6 characters.
# Centos/Redhat/ EulerOS: a sting of at least 6 characters excluding #,$, and space.
# Ubuntu: a sting of at least 8 characters excluding #, $, and space.
# Vmware: a string of 7 to 40 characters. For the ESXi 6.7, the password must
consist of letters, digits, and special characters.
# (NOTE: Smart Provisioning supports special characters #, $ and spaces from V119.)
#HostName: Host Name,This parameter is optional Installation
#CheckFirmware: Whether to verify firmware.This parameter is optional; Available
values:True, False
#Partition: Partition information. This parameter is optional. Please refer to the
installation guide of the OS
#Software: Software list. This parameter is mandatory.Format:{ "FileName": "iBMA" }
#Autopart: Whether auto-partitioning is supported. Linux/VMware: true Window: false
```



```
#MediaType:Type of the media where the OS can be deployed.This parameter is
optional; Available values:SANBoot, Disk, USB
#AutoPosition: Whether the installation drive is automatically selected ; Available
values:True
#NetCfg: Network config
tasks:
- name: ibmc deploy centos7u3 by sp
  ibmc_deploy_os_by_sp:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    os_img: "nfs://172.26.200.11/data/centos7u3.iso"
    os_config:
      InstallMode: "Recommended"
      OSType: "CentOS7U3"
      BootType: "UEFIBoot"
      CDKey: ""
      RootPwd: "{{ os_pswd }}"
      HostName: "test"
      Language: "en_US.UTF-8"
      TimeZone: "America/New_York"
      Keyboard: "us"
      CheckFirmware: False
      Partition: []
      Autopart: True
      AutoPosition: True
      Software: []
      NetCfg:
        - Device:
            Name: "eth10086"
            MAC: "***:***:***:***:***:***"
            IPv4Addresses:
              - Address: "192.168.2.44"
                SubnetMask: "255.255.0.0"
                Gateway: "192.168.2.1"
                AddressOrigin: "Static"
            IPv6Addresses:
              - Address: ""
                PrefixLength: ""
                Gateway: ""
                AddressOrigin: "Static"
            NameServers:
              - DNS: "192.168.2.1"
              - DNS: "192.168.2.2"
```

表4-1 参数说明

参数名称	参数说明	取值
os_img	OS 镜像路径（必填）	输入形式：文件传输协议://用户名:密码@服务器 IP 地址/目录/文件名 文件传输协议包括 NFS、CIFS、HTTPS

参数名称	参数说明	取值
InstallMode	安装模式（必填）	<ul style="list-style-type: none"> Recommended Customized
OSType	安装 OS 类型（必填）	<p>可取值包括：</p> <p>RHEL6U9、RHEL6U10、 RHEL7U3、RHEL7U4、 RHEL7U5、RHEL7U6、 CentOS6U9、CentOS6U10、 CentOS7U3、CentOS7U4、 CentOS7U5、CentOS7U6、 ESXi6.0、ESXi6.5、ESXi6.7、 SLES11SP4、SLES12SP2、 SLES12SP3、Ubuntu16.04、 Ubuntu16.04.1、Ubuntu16.04.2、 Win2016、Win2016 Standard Desktop、Win2016 Standard Core、 Win2016 Datacenter Desktop、 Win2016 Datacenter Core、 Win2012_R2、Win2012_R2 Standard Desktop、Win2012_R2 Standard Core、Win2012_R2 Datacenter Desktop、Win2012_R2 Datacenter Core、EulerOSV2SP3</p>
BootType	BIOS 启动模式（选填）	<ul style="list-style-type: none"> UEFIBoot LegacyBoot SecureBoot
CDKey	OS 系统的安装密钥（选填）	<ul style="list-style-type: none"> Windows/Vmware：选填，取值为 25 位安装密钥，每 5 位之间用 ‘-’ 连接，选值为[a-zA-Z0-9] Linux：为空
RootPwd	管理员初始密码（必填）	<p>在 myhosts 文件中设置该参数，设置规则如下：</p> <ul style="list-style-type: none"> Windows：最少 6 位 SUSE：最少 6 位 CentOS/Red Hat/EulerOS：最少 6 位，不含#\$空格 Ubuntu：最少 8 位，不含#\$空格 Vmware：最少 7 位，ESXi6.7 版本至少需要字母、数字和特殊字符等三种不同类型字符，最多 40 位 <p>说明</p>

参数名称	参数说明	取值
		V119 及以上版本的 Smart Provisioning 支持输入#\$空格特殊字符。
HostName	主机名称（选填）	取值规则：取值为[a-z,A-Z0-9-]，总长度不超过 15 个字符 <ul style="list-style-type: none"> Linux：非必填，必须配置网络才可生效 Windows：非必填 Vmware：非必填，必须配置网络才可生效
Language	系统语言（必填）	取值规则：字符串，请参考各个操作系统的安装指南 <ul style="list-style-type: none"> Linux：必填 Windows：必填 Vmware：为空
TimeZone	系统时区（必填）	取值规则：字符串，请参考各个操作系统的安装指南 <ul style="list-style-type: none"> Linux：必填 Windows：必填 Vmware：保持为空
Keyboard	系统键盘类型（必填）	取值规则：字符串，请参考各个操作系统的安装指南 <ul style="list-style-type: none"> Linux：必填 Windows：必填 Vmware：为空
CheckFirmware	是否校验固件（必填）	<ul style="list-style-type: none"> true false
Partition	分区信息（选填） 格式为： Partition: - Size: "64" FileSystem: "NTFS" Name: "C"	<ul style="list-style-type: none"> Windows：Name 的取值为 C-Z 的字符；FileSystem 的取值为 NTFS；Size 的取值大于 32，如果填 max，则将整盘作为数据盘 Linux：Name 的取值不包含<> :&和空格，如/、/home、swap 等；FileSystem 的取值为 ext4、ext3、ext2 和 xfs；Size 的取值大于 0，其中根分区需要大于 10，swap 分区需要大于 1，如果填 max，则分配剩余的空间 Vmware：不支持

参数名称	参数说明	取值
Autopart	是否支持自动分区（必填）	<ul style="list-style-type: none"> Linux/Vmware: 选值为 true Window: 选值为 false, 不支持自动分区
MediaType	支持系统部署的介质类型（不选该字段表示部署在本地硬盘上）（选填）	<ul style="list-style-type: none"> SANBoot: 选择在 SANBoot 盘部署操作系统（仅支持 VMware 6.5.1 和 VMware 6.7） Disk: 选择在 Disk 设备上部署操作系统 USB: 选择在 USB 设备上部署操作系统（仅支持 VMware 6.5）
AutoPosition	是否支持自动选择安装盘（必填）	true（当前仅支持自动选择安装盘）
Software	需要安装的软件列表（必填） 格式为： Software: - FileName: "iBMA"	iBMA
NetCfg	网络配置（选填）	<p>[]或配置以下参数： 说明 []表示不指定设备，可批量部署。</p> <ul style="list-style-type: none"> Device: 设备网络信息 <ul style="list-style-type: none"> Name: 被部署服务器网卡网口名称 MAC: 设备 MAC 地址 IPv4Addresses: 网口的 IPv4 地址信息 <ul style="list-style-type: none"> Address: IPv4 地址 SubnetMask: 子网掩码 AddressOrigin: IPv4 地址获取模式，可设置为“Static”或“DHCP” Gateway: IPv4 网关地址 IPv6Addresses: 网口的 IPv6 地址信息 <ul style="list-style-type: none"> Address: IPv6 地址 PrefixLength: IPv6 地址的前缀长度 AddressOrigin: IPv6 地址获取模式，可设置为“Static”或

参数名称	参数说明	取值
		<p>“DHCP”</p> <ul style="list-style-type: none">Gateway: IPv6 网关地址 <p>说明</p> <p>Ubuntu 和 VMware 系统不支持配置该选项。</p> <ul style="list-style-type: none">NameServers: DNS 服务器的 IP 地址，取值可以为 IPv4 或 IPv6 的网络地址

OSType、Language、TimeZone 和 Keyboard 的取值示例如下：

OSType	Language	TimeZone	Keyboard
RHEL/CentOS/Euler OS/Ubuntu	en_US.UTF-8	America/New_York	us
SLES	en_US	America/New_York	english-us
Windows	en-US	Eastern Standard Time	0x00000409

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行部署命令。

ansible-playbook deploy_centos7u3_by_sp.yml

如下返回信息表示命令执行成功。

```
[plugin@localhost examples]$ ansible-playbook deploy_centos7u3_by_sp.yml

PLAY [ibmc deploy centos7u3 by sp]
*****
*****

TASK [ibmc deploy centos7u3 by sp]
*****
*****
ok: [host1.domain.com]

PLAY RECAP
*****
*****
*****
host1.domain.com      : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.12 BIOS 管理

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

支持查询和设置 BIOS 信息，以及恢复 BIOS 默认配置。

4.12.1 查询 BIOS 信息（生成 json 文件）

参数配置（可选）

修改 “/home/plugin/ibmc_ansible/examples/get_bios.yml” 文件的 “bios_attribute” 参数。

bios_attribute: 查询指定的 BIOS 参数项信息，可选参数。若不配置该参数，则查询所有 BIOS 参数项信息。

查询后生成 “IP 地址_BIOSInfo.json” 文件，保存在 /home/plugin/ansible_ibmc/report/bios/ 文件夹中。

说明

- **IP 地址_BIOSInfo.json:** “IP 地址” 为配置的服务器 IP 地址。
- **/home/plugin/ansible_ibmc/report/bios/:** “plugin” 为实际使用的用户名。

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

---
- hosts: myhosts
  connection: local
  name: get bios
  gather_facts: False

# bios_attribute: User-specified BIOS attributes to be queried

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

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。
cd /home/plugin/ibmc_ansible/examples
2. 执行查询 BIOS 信息命令。

ansible-playbook get_bios.yml

如下返回信息表示命令执行成功。

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

PLAY [get bios]
*****
*****

TASK [get bios]
*****
*****

ok: [host0]

PLAY RECAP
*****
*****

host0          : ok=1   changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

4.12.2 设置 BIOS 信息

参数配置

修改“/home/plugin/ibmc_ansible/examples/set_bios.yml”文件中的“Immediately”和“bios_attribute”参数。

- **Immediately**: 设置生效时间, 取值为“True”和“False”, 默认为“False”。“True”表示立即重启使设置生效, 服务器会自动重启; “False”表示不会立即重启服务器, 需要用户自行重启使设置生效。

须知

服务器自动重启会影响业务, 请谨慎操作。

- **bios_attribute**: 需要设置的 BIOS 参数项, 必选项。

说明

下面以设置“QuickBoot”和“QuietBoot”为例。关于可设置的 BIOS 参数项可参考《[服务器 Purley 平台 BIOS 参数参考](#)》。

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

```
name: set ibmc bios
gather_facts: False

# Immediately: Whether to restart the system immediately for the configuration to
take effect: True, False
# bios_attribute: BIOS attributes set by the user

tasks:
- name: set ibmc bios
  ibmc_set_bios :
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    Immediately: False
    bios_attribute:
      QuickBoot: Disabled
      QuietBoot: Enabled
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行设置 BIOS 信息命令。

ansible-playbook set_bios.yml

如下返回信息表示命令执行成功。

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

PLAY [set ibmc bios]
*****

TASK [set ibmc bios]
*****

ok: [host0]

PLAY RECAP
*****

host0          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```


4.12.3 恢复 BIOS 默认配置

参数配置

修改 “/home/plugin/ibmc_ansible/examples/reset_bios.yml” 文件的 “Immediately” 参数。

Immediately: 设置生效时间, 取值为 “True” 和 “False”, 默认为 “False”。

- True: 表示立即重启生效, 服务器会自动重启。
- False: 表示不会立即重启服务器, 需要用户自行重启使设置生效。

须知

服务器自动重启会影响业务, 请谨慎操作。

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

---
- hosts: myhosts
  connection: local
  name: reset ibmc bios
  gather_facts: False

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

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

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行重置 BIOS 信息命令。

ansible-playbook reset_bios.yml

如下返回信息表示命令执行成功。

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

```
PLAY [reset ibmc bios]
*****
*****

TASK [reset ibmc bios]
*****
*****
ok: [host0]

PLAY RECAP
*****
*****
host0                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.13 日志管理

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

支持一键收集 iBMC 日志、收集 SEL 日志、清空 SEL 日志。

4.13.1 一键收集 iBMC 日志

参数配置

修改 “/home/plugin/ibmc_ansible/examples/collect_ibmc_logs.yml” 文件。

- **save_mode**: 表示在远程文件服务器存储或者本地存储。远程文件服务器存储取值为 “sftp”、“https”、“nfs”、“cifs” 或 “scp”；本地存储取值为 “local”。
- **file_server_ip**: 远程文件服务器 IP 地址，当 “save_mode” 设置为 “sftp”、“https”、“nfs”、“cifs” 或 “scp” 时，需要设置。
- **file_server_user**: 远程文件服务器用户名，当 “save_mode” 设置为 “sftp”、“https”、“cifs” 或 “scp” 时，需要设置。
- **file_server_passwd**: 远程文件服务器密码，当 “save_mode” 设置为 “sftp”、“https”、“cifs” 或 “scp” 时，需要设置。
- **file_name**: 日志文件保存的路径与文件名。只设置文件名时，日志文件默认保存在 /home/plugin/ansible_ibmc/report/collect_IBMC_log/ 路径下（“plugin” 为实际使用的用户名）。

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

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

tasks:
- name: collect logs
  ibmc_collect_logs :
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    save_mode: "sftp"
    file_server_ip: "sftp_server_ip"
    file_server_user: "{{ sftp_user }}"
    file_server_pswd: "{{ sftp_pswd }}"
    file_name: "/usr/dump.tar.gz"
```

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行一键收集 iBMC 日志命令。

```
ansible-playbook collect_ibmc_logs.yml
```

如下返回信息表示命令执行成功。

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

PLAY [collect logs]
*****
*****
*****

TASK [collect logs]
*****
*****
*****
ok: [host0]

PLAY RECAP
*****
*****
*****
host0                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.13.2 收集 SEL 日志

参数配置

修改 “/home/plugin/ibmc_ansible/examples/collect_sel_logs.yml” 文件。

- **save_mode**: 表示在远程文件服务器存储或者本地存储。远程文件服务器存储取值为 “sftp”、“https”、“nfs”、“cifs” 或 “scp”；本地存储取值为 “local”。
- **file_server_ip**: 远程文件服务器 IP 地址，当 “save_mode” 设置为 “sftp”、“https”、“nfs”、“cifs” 或 “scp” 时，需要设置。
- **file_server_user**: 远程文件服务器用户名，当 “save_mode” 设置为 “sftp”、“https”、“cifs” 或 “scp” 时，需要设置。
- **file_server_pswd**: 远程文件服务器密码，当 “save_mode” 设置为 “sftp”、“https”、“cifs” 或 “scp” 时，需要设置。
- **file_name**: 日志文件保存的路径与文件名。只设置文件名时，日志文件默认保存在 /home/plugin/ansible_ibmc/report/collect_SEL_log/ 路径下（“plugin” 为实际使用的用户名）。

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

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

tasks:
- name: collect sel logs
  ibmc_collect_sel_logs :
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    save_mode: "local"
    file_server_ip: "sftp_server_ip"
    file_server_user: "{{ sftp_user }}"
    file_server_pswd: "{{ sftp_pswd }}"
    file_name: "/home/plugin/SEL_log.tar.gz"
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。
cd /home/plugin/ibmc_ansible/examples
2. 执行收集 SEL 日志命令。
ansible-playbook collect_sel_logs.yml

如下返回信息表示命令执行成功。

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

PLAY [collect sel logs]
*****
*****

TASK [collect sel logs]
*****
*****
ok: [host0]

PLAY RECAP
*****
*****
host0                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.13.3 清空 SEL 日志

操作步骤

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

2. 执行清空 SEL 日志命令。

```
ansible-playbook clear_sel_logs.yml
```

如下返回信息表示命令执行成功。

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

PLAY [clear sel logs]
*****
*****

TASK [clear sel logs]
*****
*****
ok: [host0]
```

```
PLAY RECAP
*****
*****
*****
host0          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

4.14 通用接口

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

提供一个通用公共接口，通过配置它的 URL 地址和请求体，可实现 Redfish 的全部功能。

关于 Redfish 的更多内容请参见《[服务器 iBMC Redfish 接口说明](#)》。

参数配置

修改 “/home/plugin/ibmc_ansible/examples/common_api.yml” 文件。

- url: 请求 URL 地址。路径需要以/redfish 开头，如 /redfish/v1/Chassis/1/Oem/xFusion/Actions/Chassis.ControlIndicatorLED。
- request_method: 请求方式。可取值为 “patch”、“post”、“get” 或 “delete”。
- request_body: 请求消息体，请求体内容为 json 格式，如 '{"IndicatorLED":"Blinking","Duration":50}'。当请求方式设置为 “get” 或 “delete” 时，不可配置该参数。

```
[plugin@localhost examples]$ vi common_api.yml
---
- hosts: myhosts
  connection: local
  name: common api
  gather_facts: False

# url: request resource
# request_method: request method: GET, POST, PATCH, DELETE
# request_body: request body content

tasks:
- name: common api
  ibmc_common_api:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    url: "/redfish/v1/Chassis/1/Oem/xFusion/Actions/Chassis.ControlIndicatorLED"
    request_method: "POST"
    request_body: '{"IndicatorLED":"Blinking", "Duration":50}'
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行配置命令。

ansible-playbook common_api.yml

如下返回信息表示命令执行成功。

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

PLAY [common api]
*****
*****

TASK [common api]
*****
*****
ok: [host0]

PLAY RECAP
*****
*****
host0 : ok=1 changed=0 unreachable=0 failed=0
skipped=0 rescued=0 ignored=0
```

4.15 文件本地传输

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

支持上传本地文件、下载文件至本地。

4.15.1 上传本地文件

参数配置

修改 “/home/plugin/ibmc_ansible/examples/upload_file.yml” 文件的 “imgfile” 参数。

imgfile: 待上传本地文件的所在路径以及文件名，输入形式为 “文件路径/文件名”，如 /home/plugin/SOO.keytab。上传成功后的文件保存在 iBMC 的/tmp/web 目录下。

说明

服务器支持的文件类型请参见《[服务器 iBMC Redfish 接口说明](#)》中的“接口介绍 > UpdateService 资源的操作 > 文件上传”章节。

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

---
- hosts: myhosts
  connection: local
  name: file upload
  gather_facts: False

# imgfile: User-specified file to be uploaded, The format is file_path/file_name.
After the upload is successful, the file is placed in the /tmp/web on iBMC.
# The file types allowed by the V3 board are as follows:
{"hpm", "cer", "pem", "cert", "crt", "pfx", "p12", "xml", "keys", "pub"}
# The file types allowed by the V5 board are as follows:
{"hpm", "zip", "asc", "cer", "pem", "cert", "crt", "pfx", "p12", "xml", "keys", "pub", "keytab"}

# The maximum allowable hpm file of V3 single-board is 46M; the maximum allowable hpm, zip, asc file of v5 single-board is 90M.
# The maximum allowable size of cer, pem, cert, crt, xml, p12, and keytab files is 1M.
# The maximum allowable size of pfx and keys files is 2M, and the maximum allowable size of pub files is 2KB.

tasks:
- name: file upload
  ibmc_upload_file :
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    imgfile: "/home/plugin/SOO.keytab"
```

执行命令

1. 进入“/home/plugin/ibmc_ansible/examples”文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行上传本地文件命令。

ansible-playbook upload_file.yml

如下返回信息表示命令执行成功。

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

PLAY [file upload]
*****
*****
```



```
TASK [file upload]
*****
*****
ok: [host0]

PLAY RECAP
*****
*****
host0          : ok=1    changed=0    unreachable=0    failed=0
skipped=0      rescued=0    ignored=0
```

4.15.2 下载文件至本地

参数配置

修改 “/home/plugin/ibmc_ansible/examples/download_file.yml” 文件中的 “file_name” 和 “local_path” 参数。

- **file_name**: 需要下载的 iBMC 上的文件，只能是/tmp/web 目录或者/tmp/web 子目录下的文件。
例如：如果文件路径为/tmp/web/111.txt，此处的配置应为 **111.txt**；如果文件路径为/tmp/web/**/111.txt，此处的配置应为 **/**/111.txt**。
- **local_path**: 文件的本地保存路径，如果不配置，默认下载到 /home/plugin/ansible_ibmc/report/download/目录下。

说明

- **plugin**: 表示实际使用的用户名。
- 下载后文件的名称会自动修改，会在原来的名称前面加上 IP 和时间。

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

---
- hosts: myhosts
  connection: local
  name: file download
  gather_facts: False

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

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

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行下载文件命令。

ansible-playbook download_file.yml

如下返回信息表示命令执行成功。

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

PLAY [file download]
*****
*****

TASK [file download]
*****
*****
ok: [host0]

PLAY RECAP
*****
*****
host0                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.16 管理 HTTPS 服务器根证书

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

支持导入或删除远程 HTTPS 服务器根证书、导入远程 HTTPS 服务器根证书的吊销列表。

说明

仅 iBMC 3.01.12.20 及以后版本支持该功能。

4.16.1 导入远程 HTTPS 服务器根证书

参数配置

修改 “/home/plugin/ibmc_ansible/examples/https_ca_import.yml” 文件。

- **certpath**: 远程 HTTPS 服务器根证书的路径。支持执行机本地路径、iBMC 的 tmp 目录、远程文件服务器路径（例如 sftp://user:password@ip/path；目前支持的传输

协议包括 https、scp、sftp、cifs、nfs)；文件的扩展名支持 “.crt”、“.cer”、“.pem”。

- **certID**: 远程 HTTPS 传输服务器认证的根证书 ID。整型类型值，取值范围为 5~8（5~8 对应的证书用于 HTTPS 协议的远程文件传输）。

须知

指定证书 ID 导入时，如果此 ID 之前已导入一个证书，则会用新证书覆盖旧证书，为避免冲突覆盖，设置 certID 前，请先参考 [4.17 查询安全服务信息（生成 json 文件）](#) 查询已指定的 ID。

- **usage**: 证书用途，取值为 “FileTransfer”。
- **import_location**: 需要导入的证书所在位置。取值范围包括：tmp（指 iBMC 的 /tmp 目录）、local（指本地执行机）、sftp/https/nfs/cifs/scp（指远程文件服务器）

须知

“CertID” 和 “usage” 参数需至少设置其中一个。

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

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

# certpath: certificate to be imported (including the path and file name).
# When the certificate is imported from a remote file server, the format is
protocol://file_server_ip/folder/file_name
# The file name extension must be in (".crt", ".cer", ".pem").
# certID: ID of the root certificate used to authenticate the remote HTTPS server.
# - Available values: [5, 6, 7, 8].
# usage: certificate usage
# - Available values: "FileTransfer".
# import_location: location of the certificate.
# If the certificate is stored in the tmp directory of the BMC, the value is tmp.
# If the certificate is stored in a local directory, the value is local.
# If the certificate is stored on a remote file server, the value is the file
server protocol.
# - Available values: tmp, local, sftp, https, nfs, cifs, scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password

tasks:
- name: import https ca
  ibmc_https_ca_import:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
```

```
ibmc_pswd: "{{ ibmc_pswd }}"
certpath: "/tmp/xFusionCA3.crt"
certID: 5
# usage: "FileTransfer"
import_location: "tmp"
# file_server_user: "{{sftp_user}}"
# file_server_pswd: "{{sftp_pswd}}"
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行导入远程 HTTPS 服务器根证书命令。

ansible-playbook https_ca_import.yml

如下返回信息表示命令执行成功。

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

PLAY [import https ca]
*****
*****

TASK [import https ca]
*****
*****
ok: [host1]

PLAY RECAP
*****
*****
host1 : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.16.2 删除远程 HTTPS 服务器根证书

参数配置

修改 “/home/plugin/ibmc_ansible/examples/delete_https_ca.yml” 文件。

certID: 远程 HTTPS 传输服务器认证的根证书 ID。整型类型值，取值范围为 5~8（5~8 对应的证书用于 HTTPS 协议的远程文件传输）。

说明

设置 certID 前，可先参考 [4.17 查询安全服务信息（生成 json 文件）](#) 查询已导入的远程 HTTPS 服务器根证书 ID。

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

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

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

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行删除远程 HTTPS 服务器根证书命令。

ansible-playbook delete_https_ca.yml

如下返回信息表示命令执行成功。

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

PLAY [delete https ca]
*****
*****

TASK [delete https ca]
*****
*****
ok: [host1]

PLAY RECAP
*****
*****
host1                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.16.3 导入远程 HTTPS 服务器根证书的吊销列表

参数配置

修改 “/home/plugin/ibmc_ansible/examples/https_crl_import.yml” 文件。

- **certpath**: 远程 HTTPS 服务器根证书的吊销列表所在路径。支持执行机本地路径、iBMC 的 tmp 目录、远程文件服务器路径（例如 `sftp://user:password@ip/path`；目前支持的传输协议包括 `https`、`scp`、`sftp`、`cifs`、`nfs`）；文件的扩展名必须是 “.crl”。
- **certID**: 签发吊销列表的根证书对象 ID。取值必须为查询安全服务信息时，返回的 `RootCertificate` 对象中某个数组成员的 `certID`。查询安全服务信息的操作请参考 [4.17 查询安全服务信息（生成 json 文件）](#)。
- **usage**: 证书用途，取值为 “FileTransfer”。
- **import_location**: 需要被导入的远程 HTTPS 服务器根证书的吊销列表所在位置。取值范围包括：`tmp`（指 iBMC 的/tmp 目录）、`local`（指本地执行机）、`sftp/https/nfs/cifs/scp`（指远程文件服务器）

须知

“CertID” 和 “usage” 参数需至少设置其中一个。

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

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

# certpath: the crl file to be imported (including the path and file name).
# When the certificate is imported from a remote file server, the format is
protocol://file_server_ip/folder/file_name
# The file name extension must be .crl.
# certID: ID of the root certificate used to authenticate the remote HTTPS server.
# - Available values: [5, 6, 7, 8].
# usage: certificate usage
# - Available values: "FileTransfer".
# import_location: location of the crl.
# If the crl file is stored in the tmp directory of the BMC, the value is tmp.
# If the crl file is stored in a local directory, the value is local.
# If the crl file is stored on a remote file server, the value is the file
server protocol.
# - Available values: tmp, sftp, https, nfs, cifs, scp
# file_server_user: remote file server user name
# file_server_pswd: remote file server password

tasks:
- name: import https crl
  ibmc_https_crl_import:
```

```
ibmc_ip: "{{ ibmc_ip }}"
ibmc_user: "{{ ibmc_user }}"
ibmc_pswd: "{{ ibmc_pswd }}"
certpath: "/tmp/xFusionCA3.crl"
certID: 5
# usage: "FileTransfer"
import_location: "tmp"
# file_server_user: "{{sftp_user}}"
# file_server_pswd: "{{sftp_pswd}}"
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行导入远程 HTTPS 服务器根证书的吊销列表命令。

ansible-playbook https_crl_import.yml

如下返回信息表示命令执行成功。

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

PLAY [import https_crl]
*****

TASK [import https_crl]
*****
ok: [host1]

PLAY RECAP
*****
host1                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

4.17 查询安全服务信息（生成 json 文件）

说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

查询服务器当前支持的安全服务信息。

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行查询安全服务信息命令。

ansible-playbook get_security_service_information.yml

如下返回信息表示命令执行成功。

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

PLAY [get security service information]
*****

TASK [get security service information]
*****

ok: [host1]

PLAY RECAP
*****
*****
host1 : ok=1   changed=0    unreachable=0    failed=0
skipped=0   rescued=0   ignored=0
```

查询后生成 “IP 地址_SecurityServiceInfo.json” 文件，保存在
/home/plugin/ansible_ibmc/report/security_service/目录下，可导出进行查看。

4.18 打开或关闭 HTTPS 文件服务器证书校验

📖 说明

管理模块 MM921、交换模块 CX320/CX621 不支持此功能。

功能介绍

打开或关闭 HTTPS 文件服务器证书校验功能。

📖 说明

仅 iBMC 3.01.12.20 及以后版本支持该功能。

参数配置

修改 “/home/plugin/ibmc_ansible/examples/set_https_cert_verification.yml” 文件。

verify_cmd: 远程 HTTPS 文件服务器证书校验开关。取值范围: True (打开 HTTPS 文件服务器证书校验)、False (关闭 HTTPS 文件服务器证书校验)。

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

---
- hosts: myhosts
  connection: local
  name: set https cert verification
  gather_facts: False

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

tasks:
- name: set https cert verification
  ibmc_set_https_cert_verification:
    ibmc_ip: "{{ ibmc_ip }}"
    ibmc_user: "{{ ibmc_user }}"
    ibmc_pswd: "{{ ibmc_pswd }}"
    verify_cmd: True
```

执行命令

1. 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

cd /home/plugin/ibmc_ansible/examples

2. 执行打开或关闭 HTTPS 文件服务器证书校验命令。

ansible-playbook set_https_cert_verification.yml

如下返回信息表示命令执行成功。

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

PLAY [set https cert verification]
*****

TASK [set https cert verification]
*****

ok: [host1]

PLAY RECAP
*****
*****

host1                : ok=1    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

A FAQ

A.1 如何加密文件以及如何查看、编辑和执行已加密的文件

在使用加密命令（**ansible-vault**）之前，可通过执行 **ansible-vault -h** 查看该命令的相关帮助信息。

```
[root@localhost ~]# ansible-vault -h
Usage: ansible-vault [create|decrypt|edit|encrypt|encrypt_string|rekey|view]
[options] [vaultfile.yml]

encryption/decryption utility for Ansible data files

Options:
  --ask-vault-pass          ask for vault password
  -h, --help                show this help message and exit
  --new-vault-id=NEW_VAULT_ID
                           the new vault identity to use for rekey
  --new-vault-password-file=NEW_VAULT_PASSWORD_FILE
                           new vault password file for rekey
  --vault-id=VAULT_IDS     the vault identity to use
  --vault-password-file=VAULT_PASSWORD_FILES
                           vault password file
  -v, --verbose            verbose mode (-vvv for more, -vvvv to enable
                           connection debugging)
  --version                show program's version number, config file location,
                           configured module search path, module location,
                           executable location and exit

See 'ansible-vault <command> --help' for more information on a specific
command.
```

A.1.1 如何加密文件

步骤 1 进入“/home/plugin/ibmc_ansible/examples”文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

步骤 2 执行以下命令加密文件。

```
ansible-vault encrypt 文件名称
```

如加密 `get_basic_info.yml` 文件：

执行 **ansible-vault encrypt get_basic_info.yml** 命令，提示输入密码。

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

步骤 3 输入需设置的密码，按“Enter”，提示再次输入密码。

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

步骤 4 再次输入设置的密码，按“Enter”，显示文件加密成功。

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

----结束

A.1.2 如何查看已加密文件

步骤 1 进入“`/home/plugin/ibmc_ansible/examples`”文件目录。

cd /home/plugin/ibmc_ansible/examples

步骤 2 执行以下命令查看加密文件。

ansible-vault view 文件名称

如查看已加密的 `get_basic_info.yml` 文件：

执行 **ansible-vault view get_basic_info.yml** 命令，提示输入文件的密码。

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

步骤 3 输入文件的密码，查看文件内容。

```
[plugin@localhost examples]$ ansible-vault view get_basic_info.yml
Vault password:
---
- hosts: myhosts
  connection: local
  name: get bmc basic info
  gather_facts: False

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

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

```
ibmc_pswd: "{{ ibmc_pswd }}"  
csv_format: True
```

----结束

A.1.3 如何编辑已加密文件

步骤 1 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

步骤 2 执行以下命令编辑加密文件。

```
ansible-vault edit 文件名称
```

如编辑已加密的 `get_basic_info.yml` 文件：

执行 `ansible-vault edit get_basic_info.yml` 命令，提示输入文件的密码。

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

步骤 3 输入文件的密码，编辑文件内容。

----结束

A.1.4 文件加密后，如何执行配置命令

步骤 1 进入 “/home/plugin/ibmc_ansible/examples” 文件目录。

```
cd /home/plugin/ibmc_ansible/examples
```

步骤 2 执行配置命令。

```
ansible-playbook -vv 文件名称 --ask-vault-pass
```

如查询服务器基本信息：

执行 `ansible-playbook -vv get_basic_info.yml --ask-vault-pass` 命令，提示输入文件的密码，此处以 Python3 环境为例。

```
[plugin@localhost examples]$ ansible-playbook -vv get_basic_info.yml --ask-vault-pass  
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177:  
CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by the  
OpenSSL project, please upgrade. The next version of cryptography will drop support  
for it.  
utils.CryptographyDeprecationWarning,  
ansible-playbook 2.9.9  
config file = None  
configured module search path = ['/root/.ansible/plugins/modules',  
'/usr/share/ansible/plugins/modules']  
ansible python module location = /usr/local/python3/lib/python3.7/site-packages/ansible-2.9.9-py3.7.egg/ansible  
executable location = /usr/local/python3/bin/ansible-playbook
```

```
python version = 3.7.5 (default, Nov 16 2020, 23:36:26) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]
No config file found; using defaults
Vault password:
```

步骤3 输入文件的密码，查询服务器基本信息。

```
[plugin@localhost examples]$ ansible-playbook -vv get_basic_info.yml --ask-vault-pass
/usr/local/python3/lib/python3.7/site-packages/cryptography-3.1.1-py3.7-linux-x86_64.egg/cryptography/hazmat/bindings/openssl/binding.py:177:
CryptographyDeprecationWarning: OpenSSL version 1.0.2 is no longer supported by the OpenSSL project, please upgrade. The next version of cryptography will drop support for it.
utils.CryptographyDeprecationWarning,
ansible-playbook 2.9.9
config file = None
configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
ansible python module location = /usr/local/python3/lib/python3.7/site-packages/ansible-2.9.9-py3.7.egg/ansible
executable location = /usr/local/python3/bin/ansible-playbook
python version = 3.7.5 (default, Nov 16 2020, 23:36:26) [GCC 4.8.5 20150623 (Red Hat 4.8.5-44)]
No config file found; using defaults
Vault password:

PLAYBOOK: get_basic_info.yml
*****
1 plays in get_basic_info.yml

PLAY [get bmc basic info]
*****
**
META: ran handlers

TASK [get bmc basic info]
*****
**
task path: /home/ibmc_ansible/examples/get_basic_info.yml:10
ok: [host0] => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python"}, "changed": false, "msg": "Get basic info successful! For more detail information, please refer the report log: /home/root/ansible_ibmc/report/basic_info/192.168.2.10_BasicInfo.json"}
META: ran handlers
META: ran handlers

PLAY RECAP
*****
*****
host0 : ok=1 changed=0 unreachable=0 failed=0 skipped=0
rescued=0 ignored=0
```

----结束

A.2 导入本地 HTTPS 服务器根证书/吊销列表提示 invalid upload file

问题现象

导入本地 HTTPS 服务器根证书或吊销列表失败，并提示 “invalid upload file”，如下所示。

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

PLAY [import https ca]
*****
*****
*****

TASK [import https ca]
*****
*****
*****
fatal: [host0]: FAILED! => {"ansible_facts": {"discovered_interpreter_python":
"/usr/bin/python"}, "changed": false, "msg": "Import remote https server root ca
failed! The detailed information is as follows: Send request to upload the file
failed! The error code is: 400, The error info is: {'error': {'code':
'Base.1.0.GeneralError', 'Message': 'A general error has occurred. See ExtendedInfo
for more information.', '@Message.ExtendedInfo': [{'@odata.type':
'#MessageRegistry.1.0.0.MessageRegistry', 'MessageId':
'iBMC.0.1.0.FirmwareUploadError', 'RelatedProperties': [], 'Message': 'An error
occurred during the firmware upload process. Details: invalid upload file.',
'MessageArgs': [], 'Severity': 'Warning', 'Resolution': 'Locate the cause based on
error information, rectify the fault, and submit the request again.'}]}} "}}

PLAY RECAP
*****
*****
*****
host0                : ok=0    changed=0    unreachable=0    failed=1    skipped=0
rescued=0    ignored=0
```

解决方法

步骤 1 确认证书/吊销列表文件是否正确。

- 是：执行步骤 2。
- 否：使用正确的证书/吊销列表文件，再次执行导入命令。

步骤 2 将服务器的 iBMC 升级至最新版本，再次执行导入命令。

----结束

说明

若问题未能解决，请联系技术支持。

B 获取技术支持

如果在设备维护或故障处理过程中，遇到难以确定或难以解决的问题，通过文档的指导仍然不能解决，请通过如下方式获取技术支持：

- 联系客户服务中心。
 - 客户服务电话：400-009-8999
 - 客户服务邮箱：support@xfusion.com
- 联系技术支持人员。

C 通讯矩阵

源设备	源 IP	源端口	目的设备	目的 IP	目的端口 (侦听)	协议	端口说明	侦听端口是否更改	认证方式	加密方式
Ansible 所属设备	Ansible 虚拟网口的 IP	随机端口	iBMC 所属设备	iBMC 虚拟网口 veth 的 IP	22	SSH	SSH 标准协议端口，有文件传输服务的时候开启。Ansible 作为客户端访问 iBMC 设备。	不涉及	用户名/密码	SSH