

华为认证 Computing 系列教程

# HCIA-Computing

# 计算产品部署实验

# 指导手册

版本：3.0



华为技术有限公司

版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

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。

## 华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://e.huawei.com>

---

## 华为认证体系介绍

华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，打造的覆盖ICT（Information and Communications Technology，信息技术）全技术领域的认证体系，包含ICT基础设施认证、基础软硬件认证、云平台及云服务认证三类认证。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。

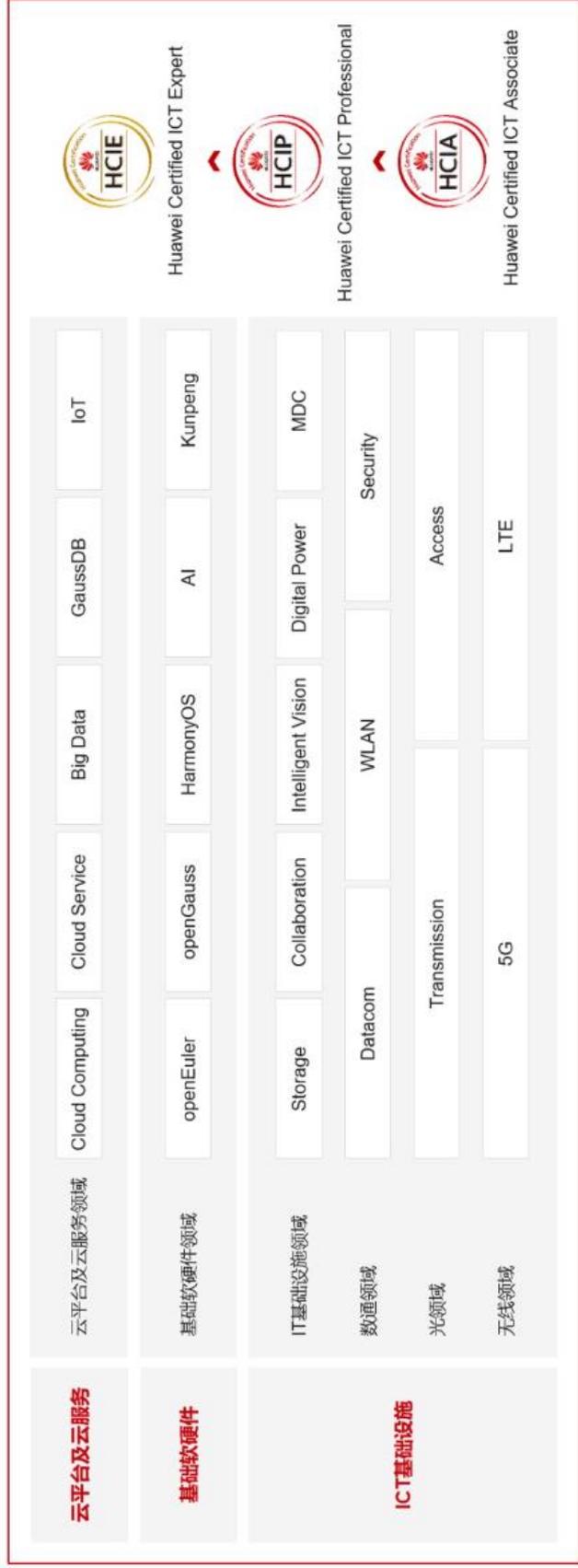
华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

华为认证HCIA-Computing V3.0定位于培养与认证具备计算产品安装与基础维护能力的计算工程师。

通过HCIA-Computing V3.0认证，您将了解计算产业的发展历史及产业前景，理解华为计算产业的愿景和目标，掌握计算系统及服务器产品相关知识，具备计算系统及业务系统搭建、基础维护能力，胜任计算产品售前工程师、计算产品交付工程师、IT工程师等岗位。

---

# 华为职业认证



## 简介

本书为 HCIA-Computing 认证培训教程，适用于准备参加 HCIA-Computing 考试的学员或者希望了解服务器基础知识、openEuler 系统安装以及华为 Atlas 智能卡的驱动配置及简单使用的读者。如果无法获取 Atlas 800（型号 3000）则可以使用 Taishan 200（型号 2280）服务器替代，本实验中的 NPU 相关操作均为选做。

## 内容描述

本实验指导书共包含四个实验，从设备初始化配置开始，逐一介绍了华为 Atlas 800（型号 3000）服务器的部署、系统部署、常见服务的部署及 Atlas 智能卡的部署。

- 实验一为服务器的基础配置，包括服务器初始化配置、RAID 配置等；
- 实验二为 Atlas 800 服务器的操作系统安装，以虚拟光驱的形式安装单台服务器的操作系统；
- 实验三为常见服务的部署，包括基于 FTP 的 rpm 源、DNS 服务器和邮件服务器；
- 实验四为选做实验，内容包括推理卡 NPU 安装、CANN 部署以及推理卡测试。

## 读者知识背景

本课程为华为认证基础课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

- 具有电脑操作的基础能力，理解硬件、系统和驱动的基本概念。
- 熟悉 Linux 基础命令。

## 实验环境说明

### 组网说明

本实验环境面向准备 HCIA-Computing 考试的 IT 工程师。每套实验环境包括 1 台 Atlas 800 服务器，每台服务器配置 2 块 SSD 作为系统盘，4 块 SATA 作为数据盘。

---

# 准备实验环境

## 检查设备

实验开始之前请每组学员检查自己的实验设备是否齐全，实验清单如下：

设备名称	数量	备注
Atlas 800 ( 型号3000 )	1台	一人一台

---

# 目录

---

<b>前 言</b> .....	<b>3</b>
简介.....	3
内容描述.....	3
读者知识背景.....	3
实验环境说明.....	3
准备实验环境.....	4
<b>1 实验环境介绍</b> .....	<b>7</b>
1.1 实验介绍.....	7
1.1.1 关于本实验.....	7
1.1.2 实验设备介绍.....	7
<b>2 Atlas 800 服务器初始化</b> .....	<b>8</b>
2.1 实验介绍.....	8
2.1.1 关于本实验.....	8
2.1.2 实验目的.....	8
2.2 实验任务配置.....	8
2.2.1 Atlas 800 服务器 iBMC 配置.....	8
2.2.2 Atlas 800 服务器 RAID 配置.....	16
2.3 思考题.....	24
<b>3 openEuler 操作系统的安装和操作</b> .....	<b>25</b>
3.1 实验介绍.....	25
3.1.1 关于本实验.....	25
3.1.2 实验目的.....	25
3.1.3 实验所需软件.....	25
3.2 实验任务配置.....	25
3.2.1 操作系统安装.....	25
3.2.2 磁盘分区与逻辑卷配置.....	35
3.3 思考题.....	37
<b>4 常见服务的部署</b> .....	<b>38</b>
4.1 实验介绍.....	38
4.1.1 关于本实验.....	38

---

4.1.2 实验目的.....	38
4.1.3 实验所需软件.....	38
4.2 部署基于 FTP 的私有 yum 源.....	39
4.3 部署邮箱服务器.....	41
4.3.1 搭建 DNS 服务器.....	41
4.3.2 源码编译安装 MySQL.....	43
4.3.3 数据库内容配置.....	46
4.3.4 安装及配置 postfix 邮局.....	47
4.3.5 配置 dovecot.....	52
4.3.6 使用邮件客户端测试.....	56
4.4 思考题.....	63
<b>5 (选做) Atlas 800(3000)推理场景 AI 软件部署.....</b>	<b>64</b>
5.1 实验介绍.....	64
5.1.1 关于本实验.....	64
5.1.2 实验目的.....	64
5.1.3 实验所需软件.....	64
5.2 实验任务配置.....	65
5.2.1 安装 NPU 驱动.....	65
5.2.2 安装 AI 软件.....	74
5.2.3 AI 算力测试.....	79
5.2.4 推理应用示例.....	81
5.3 思考题.....	84

---

# 1 实验环境介绍

## 1.1 实验介绍

### 1.1.1 关于本实验

Atlas 800 服务器是华为推出的在 AI 领域常用的智能计算服务器，本实验围绕 Atlas 800 服务器的部署进行展开，包括服务器的初始化开始，到操作系统的安装、数据库的安装、Atlas 300 智能卡的使用等几个部分。

### 1.1.2 实验设备介绍

设备名称、型号与版本的对应关系如下：

表1-1 实验设备配套关系

设备名称	设备型号	软件版本
Atlas 800	Model 3000	/
Atlas智能卡	Atlas 300I Pro Inference Card	固件版本：3.3.5
RAID卡	AVAGO MegaRAID SAS 9460-8i	SAS3508
硬盘	SATA-SSD-480GB*2 SATA-HDD-8T*4	/
实验操作桌面	个人电脑	Windows

# 2 Atlas 800 服务器初始化

---

## 2.1 实验介绍

### 2.1.1 关于本实验

本实验主要介绍 Atlas 800 服务器的初始化操作，包含 iBMC 网口配置、RAID 配置、服务器硬件状态检查等内容。

### 2.1.2 实验目的

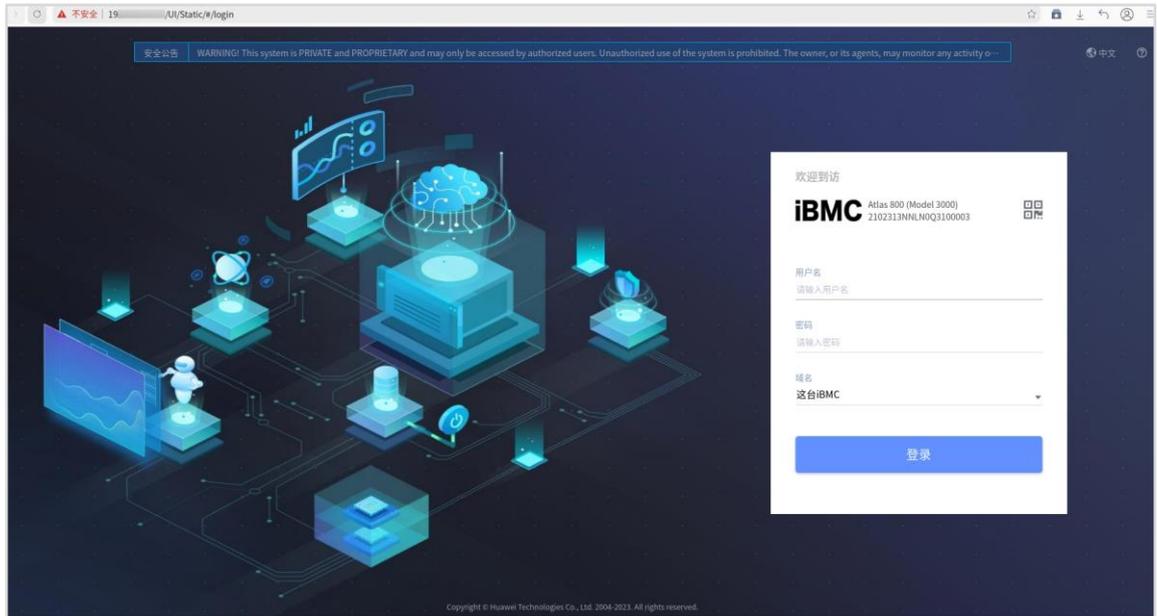
- 掌握 Atlas 800 服务器的 BIOS 配置；
- 理解 iBMC 的作用；
- 掌握在 BIOS 和 WEB 中配置 RAID 的方法；
- 理解服务器 iBMC 的功能。

## 2.2 实验任务配置

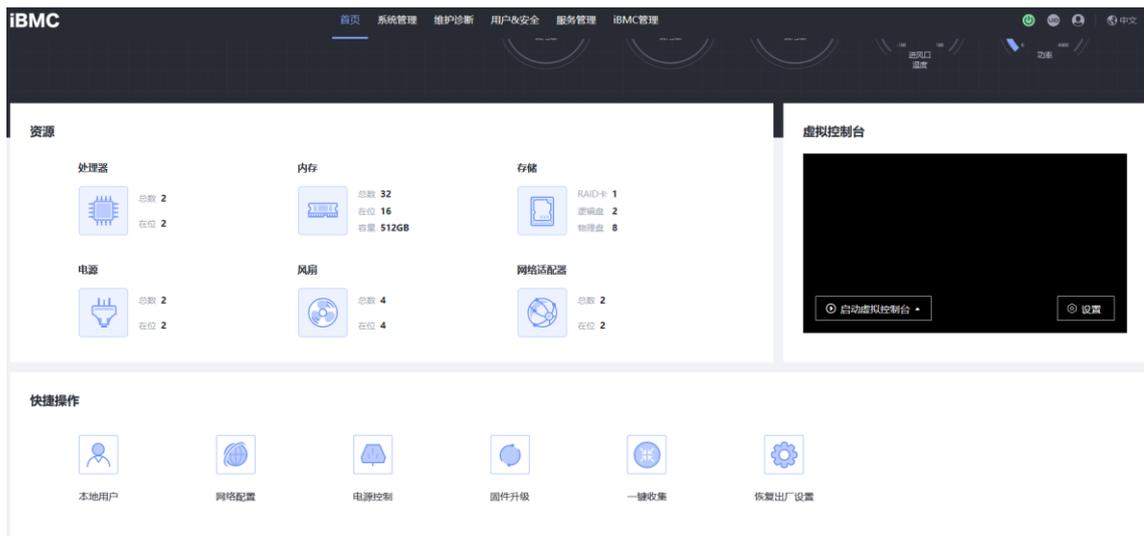
### 2.2.1 Atlas 800 服务器 iBMC 配置

说明：实际环境需要使用键盘、显示器直连服务器进行配置。本实验在已知 iBMC 端口配置的情况下使用虚拟控制台进行服务器的 iBMC 配置。

- 步骤 1 根据实际环境规划，在浏览器中输入服务器的 iBMC 端口 IP 地址，进入到服务器的 iBMC 登录界面。输入规划的用户名和密码登录服务器。
-



步骤 2 进入 iBMC 的 web 界面后，可以在首页看到服务器的基本信息，包括处理器、内存、存储等基本硬件情况。也可以对服务器进行一些快捷操作，比如网络配置、电源管理等。



步骤 3 点击“系统管理”，会进入到系统信息页签，可以看到服务器的产品信息。

**iBMC** 系统管理 | 维护诊断 | 用户及安全 | 服务管理 | iBMC管理

**系统管理**

- 系统信息
- 性能监控
- 存储管理
- 电源&功率
- 风扇&散热
- BIOS配置

**系统信息**

产品信息 | 处理器 | 内存 | 网络适配器 | 传感器 | 其他

**产品信息**

产品名称	Atlas 800 (Model 3000)(VD)	生产厂商	Huawei
产品序列号	2102313NNLN0Q3100003	部件编码	02313NNL
资产标签	--	产品位置	--

**主板信息**

iBMC固件版本	3.11.00.09(U82)	单板ID	0x00a9
BIOS版本	6.67(U75)	主板厂商	Huawei
CPLD版本	7.03(U6076)	主板型号	S104-00712
iBMC主UBoot版本	17.0.10.2 (10:13:38 Dec 27 2023)	主板序列号	2106411453FSQ1004916
iBMC备UBoot版本	17.0.10.2 (10:13:38 Dec 27 2023)	PCB 版本	.B
部件编码	06411453		

**系统软件**

步骤 4 在系统信息中，点击进入“处理器”页签，可以查看处理器的详细信息。

**系统管理**

- 系统信息
- 性能监控
- 存储管理
- 电源&功率
- 风扇&散热
- BIOS配置

**系统信息**

产品信息 | 处理器 | 内存 | 网络适配器 | 传感器 | 其他

总数 2 在位 2

名称	厂商	型号	主频	核数/线程数	一级/二级/三级缓存	状态
CPU1	HISilicon	HUAWEI Kunpeng 920 5250	2600 MHz	48 cores/48 threads	6144/24576/49152 KB	启用

**详细信息**

名称	CPU1	厂商	HISilicon
型号	HUAWEI Kunpeng 920 5250	处理器ID	10-D0-1F-48-00-00-00-00
主频	2600 MHz	核数/线程数	48 cores/48 threads
一级/二级/三级缓存	6144/24576/49152 KB	状态	启用
序列号	A9850EE42020AB1F	其他参数	64-bit Capable  Multi-Core  Execute Protection  Enhanced Virtualization  Power/Performance Control

▼ CPU2 HISilicon HUAWEI Kunpeng 920 5250 2600 MHz 48 cores/48 threads 6144/24576/49152 KB 启用

步骤 5 在系统信息中，点击进入“内存”页签，可以查看内存的详细信息。

**系统信息**

产品信息 | 处理器 | 内存 | 网络适配器 | 传感器 | 其他

总数 32 在位 16

名称	厂商	容量	配置速度	最大速度	类型	位置
DIMM00	JINGSHUI	32768 MB	2933 MT/s	3200 MT/s	DDR4	mainboard

**详细信息**

名称	DIMM00	位置	mainboard
厂商	JINGSHUI	容量	32768 MB
配置速度	2933 MT/s	最大速度	3200 MT/s
类型	DDR4	最小电压	1200 mV
类型详细信息	Synchronous  Registered (Buffered)	位宽	72 bit
部件编码	JSR413137SXXM-XSCC	Rank数	1 rank
序列号	12107711		

▼ DIMM01 JINGSHUI 32768 MB 2933 MT/s 3200 MT/s DDR4 mainboard

步骤 6 在系统信息中，点击进入“网络适配器”页签，可以查看服务器网卡的详细信息。

**系统信息**

产品信息 | 处理器 | 内存 | **网络适配器** | 传感器 | 其他

以太网卡

- NIC 1 (TM210)
- NIC 2 (TM280)

**网卡设备**

网卡名称	NIC 1 (TM210)	型号	4*GE
单板ID	0x0068	厂商	Huawei
芯片厂商	Huawei	芯片型号	CPU Integration
PCB版本	.A	资源归属	CPU1
总线信息	0000:7c:00.0		

**端口属性**

端口	状态	网口类型	介质类型
Port1	连接	物理网口	电口

**网络属性**

固件版本	--	总线信息	0000:7d:00.0
MAC地址	B0:4F:A6:56:CE:FD	永久MAC地址	B0:4F:A6:56:CE:FD

步骤 7 点击左侧导航栏的“存储管理”，进入存储管理页签，可以看到服务器的 RAID 卡、磁盘等信息。

**系统管理**

- 系统信息
- 性能监控
- 存储管理**
- 电源&功率
- 风扇&散热
- BIOS配置

**存储管理**

添加 编辑

Ctrl Card 3 (9460-8i)

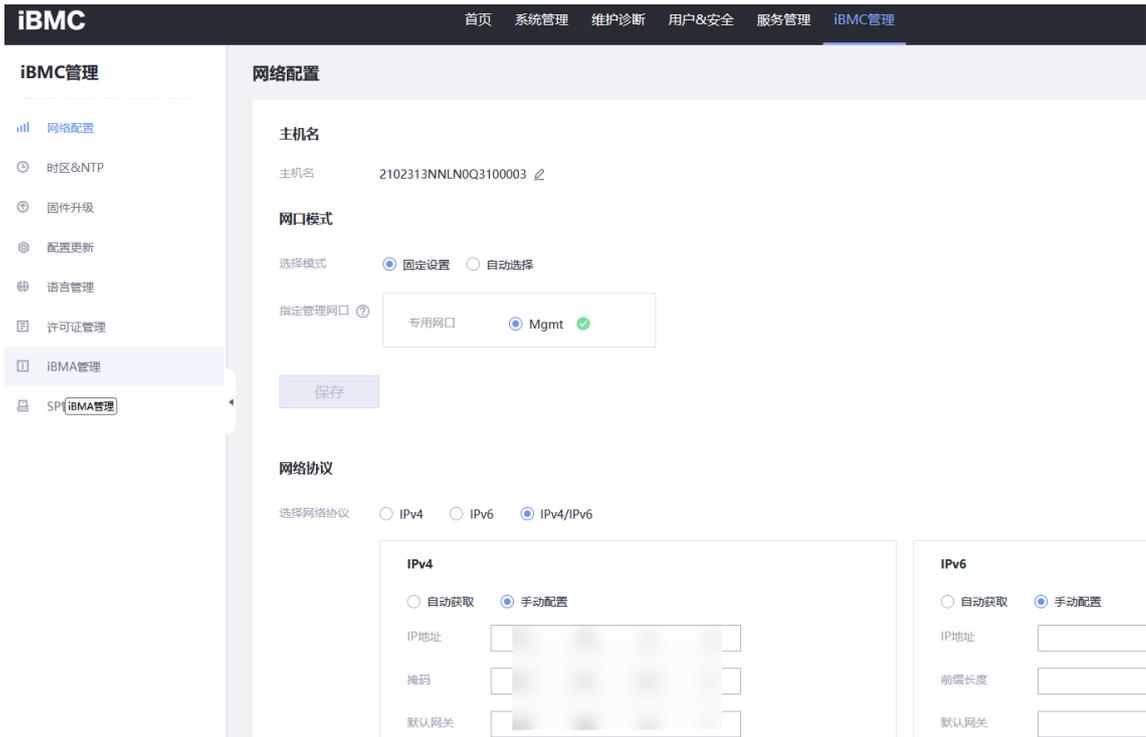
此页面的RAID控制器、逻辑驱动器、物理驱动器的信息依赖RAID卡的带外管理功能，并且在系统引导完成后或全自动BMA 2.0才能显示。

**控制器信息**

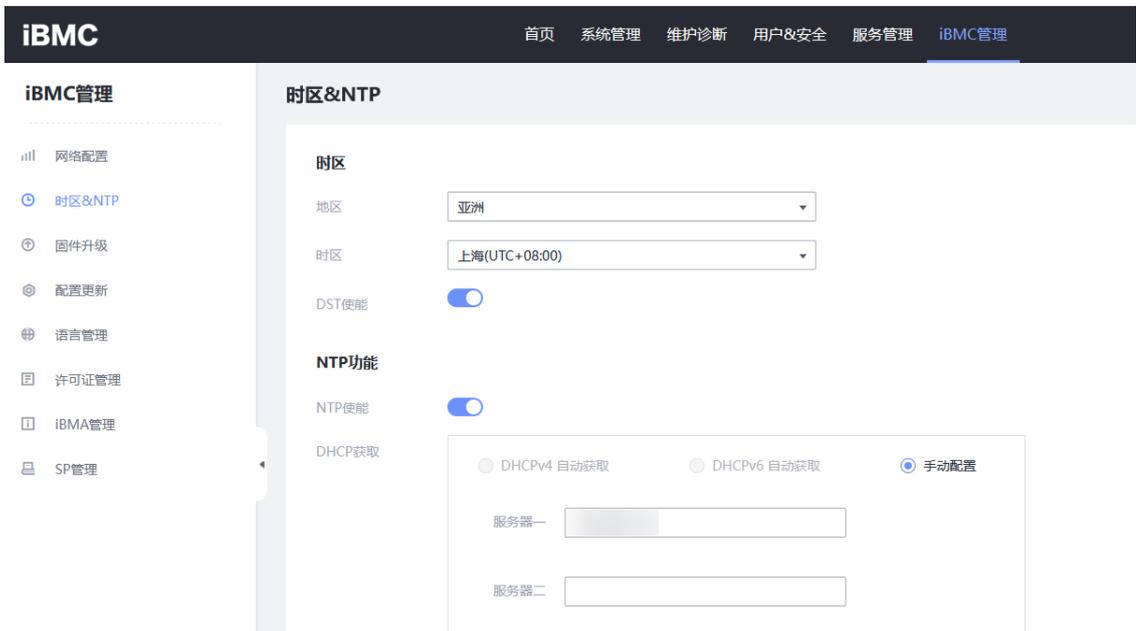
设置

名称	AVAGO MegaRAID SAS 9460-8i	类型	SAS3508
固件版本	5.230.00-3767	支持带外管理	是
健康状态	正常	支持的RAID级别	RAID(0/1/5/6/10/50/60)
工作模式	RAID	内存大小	2048 MB
设备接口	SAS 12 GB	SAS地址	500062b21536e640
支持的条带大小范围	64 KB - 1 MB	JBOD模式	已禁用
Cache Pinned状态	否	物理盘故障记忆	已启用
配置版本	5.2300.00-0005	回滚	已启用

步骤 8 点击上方的“iBMC 管理”，会进入到 iBMC 的网络页签，可以看到当前的 iBMC 网络配置信息。实际部署时可以选择在 web 界面更改 iBMC 的 IP 地址，本实验不做修改。

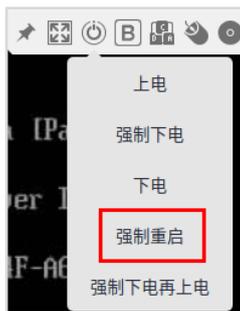


步骤 9 在左侧导航栏中点击“时区&NTP”，进入后可以根据实际情况修改服务器 iBMC 时区和 NTP 配置。本实验不做修改。



步骤 10 回到“首页”，点击“启动虚拟控制台”，选择“HTML5 集成远程控制台（共享）”。



步骤 11 在控制台页中点击电源图标“

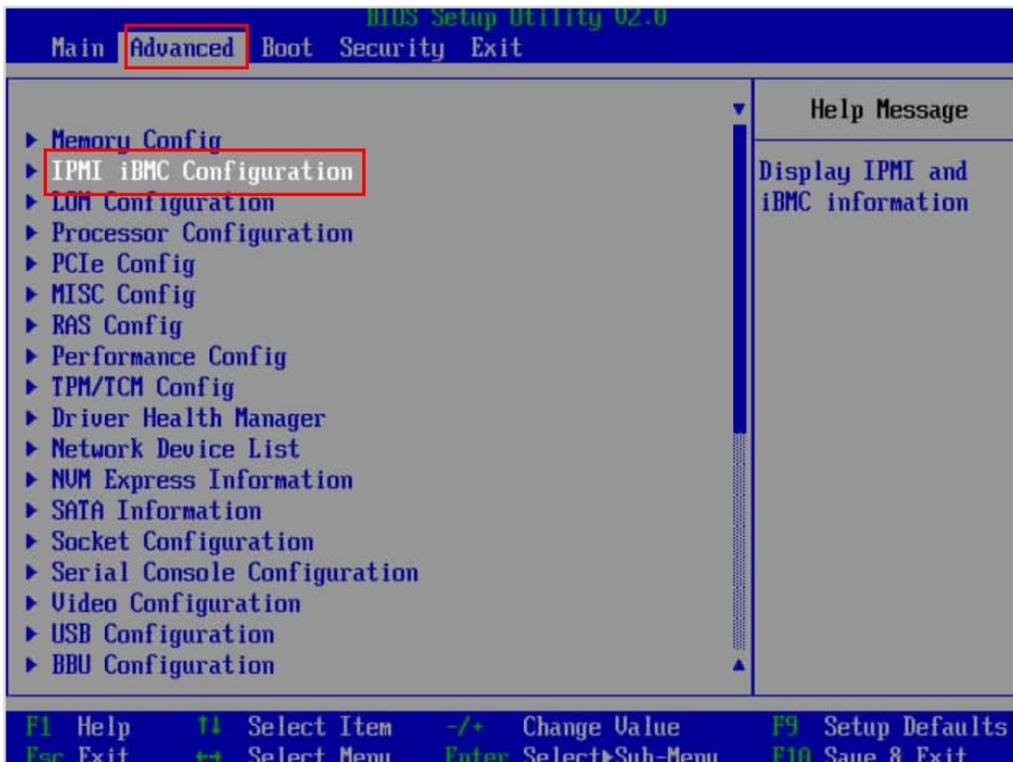
The screenshot shows a power control menu with the following options: 上电, 强制下电, 下电, 强制重启 (highlighted with a red box), and 强制下电再上电.

步骤 12 等待服务器出现如下页面时，按“Delete”键。

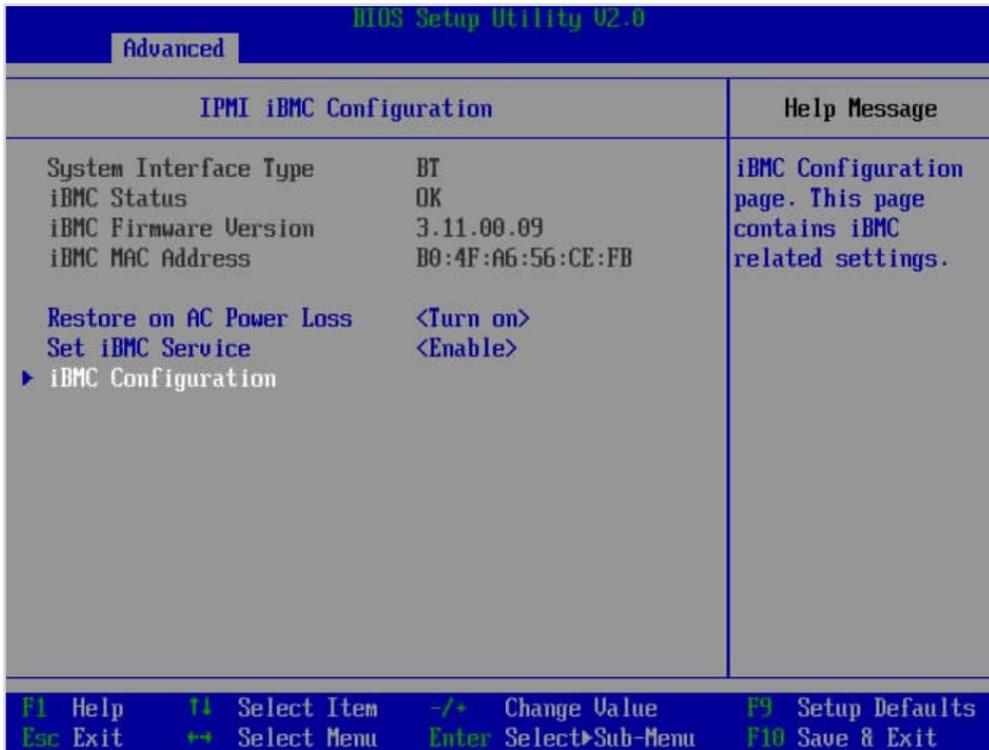
```
Version : 6.67
Processor Type : HUAWEI Kunpeng 920 5250
Total Memory : 524288 MB
BMC IP : 192.168.10.14
CPU ID : 481FD010

Press Del go to Setup Utility (F4 on Remote Keyboard)
Press F12 go to PXE
Press F2 go to Boot Option
Press F6 go to SP Boot
9 seconds left, Press F4 or DEL to enter Setup, F2 for boot options
```

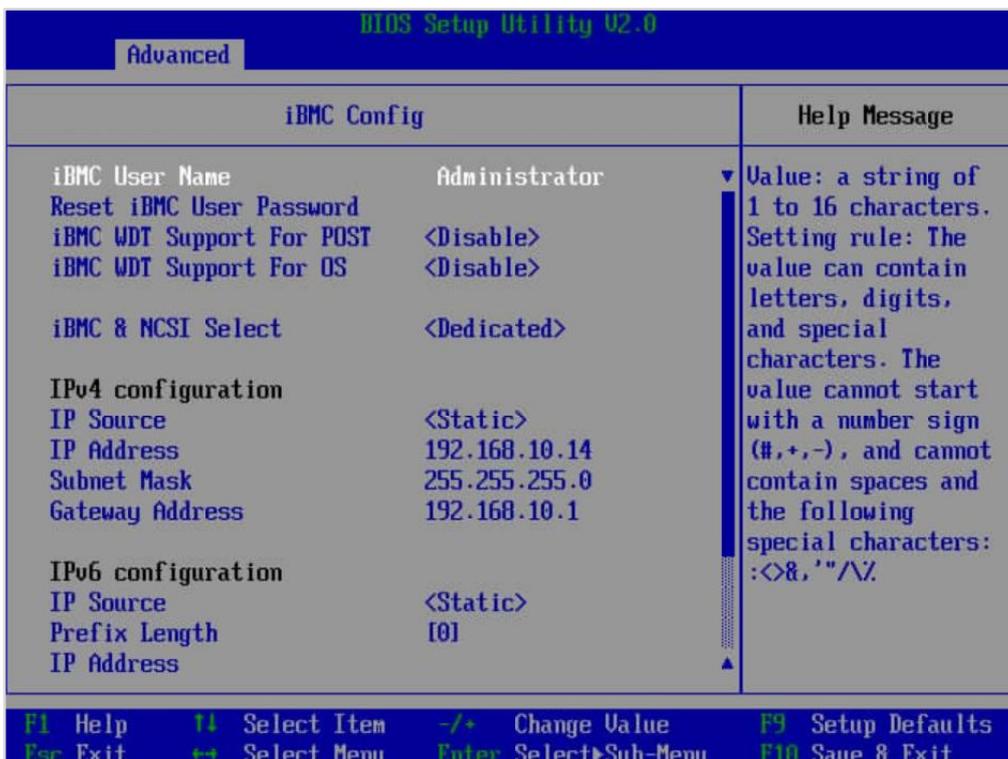
- 步骤 13 在弹出的窗口中输入密码，按回车键进入 BIOS，若有提示修改密码，请忽略，再次按回车进入。（默认是 Admin@9000）
- 步骤 14 按向右键进入“Advanced”页签，按向下键将光标移动到“IPMI iBMC Configuration”项，然后按回车，进入 iBMC 配置页。



步骤 15 按向下键，将光标移动到“iBMC Configuration”项，按回车键进入配置。

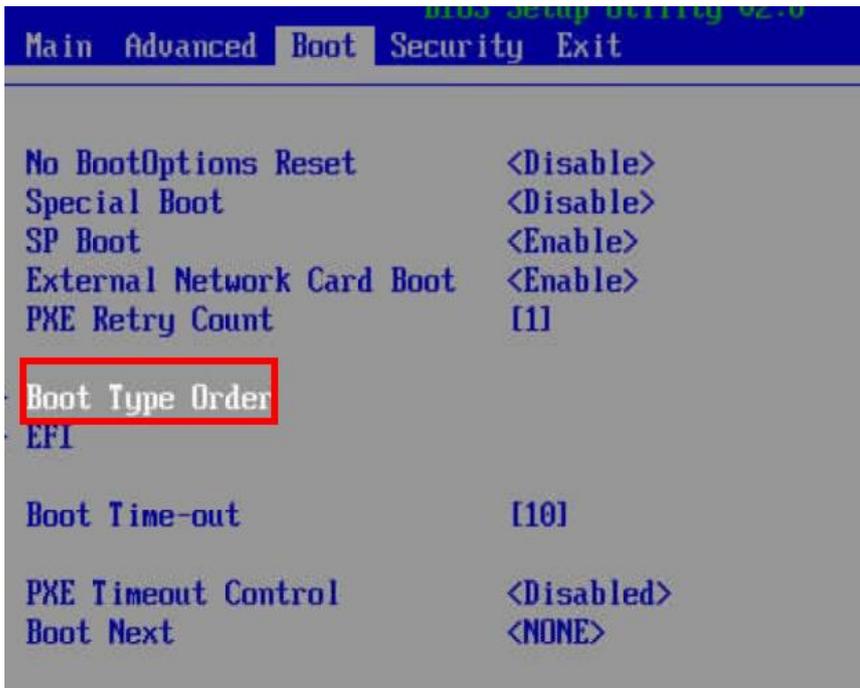


步骤 16 在此页中可以对 iBMC 的登录信息进行配置，包括登录的用户名、密码、IP 地址等信息。本次实验不做修改，实际环境中根据规划进行配置。

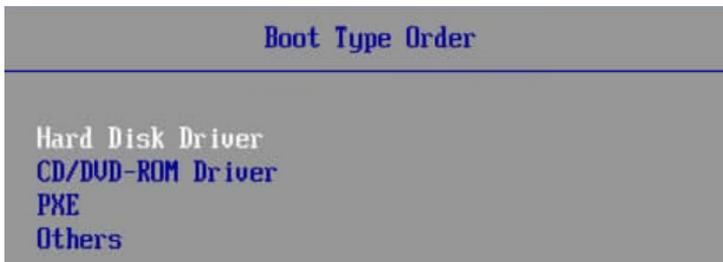


步骤 17 按两次 Esc 键返回到“Advanced”页签。

步骤 18 按左键进入到“Boot”页签，按下键将光标移动到“Boot Type Order”。

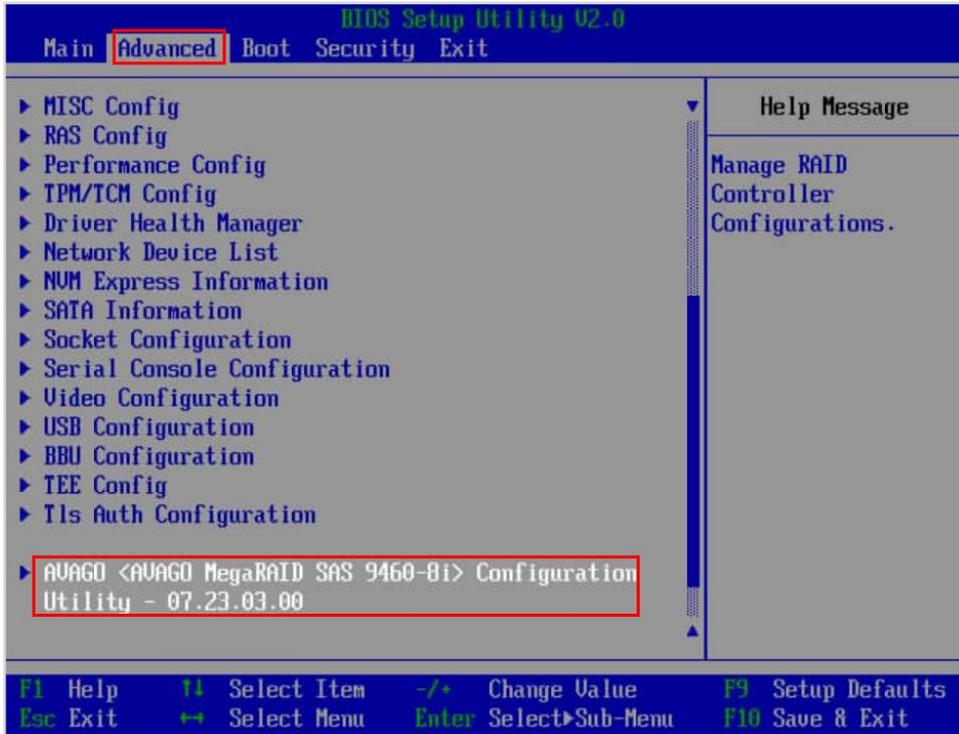


步骤 19 按回车键，可以选择启动顺序。在批量安装服务器操作系统时建议使用 PXE 方式安装。

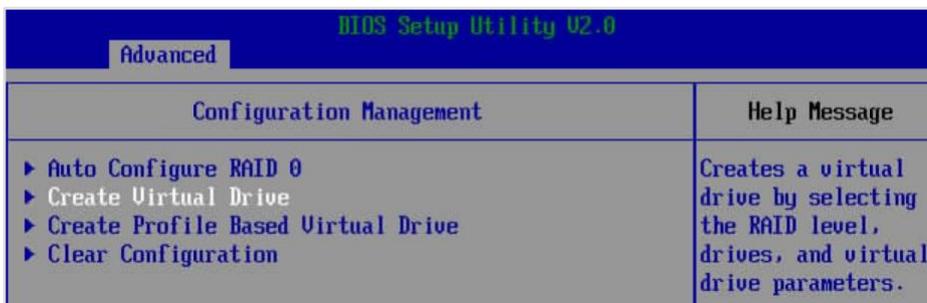
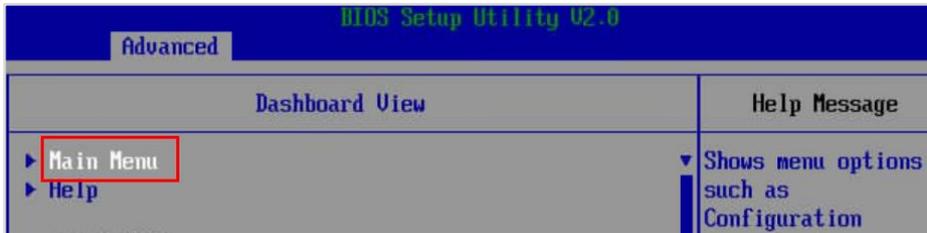


## 2.2.2 Atlas 800 服务器 RAID 配置

步骤 1 返回 BIOS 配置的“Advanced”页签，按向下键，将光标移动到最后的 AVAGO RAID 卡配置项，按回车键进入。



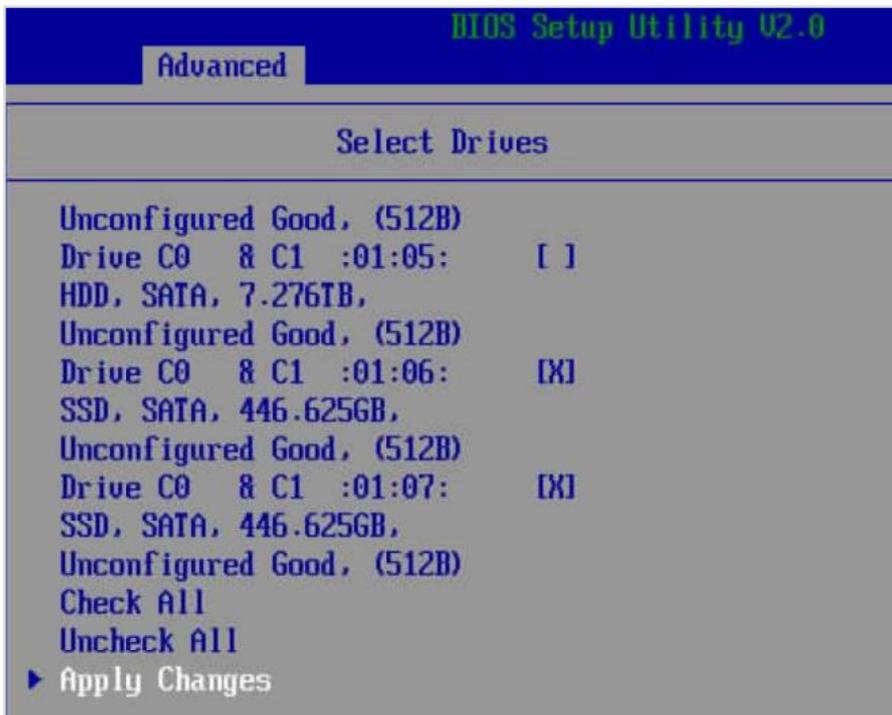
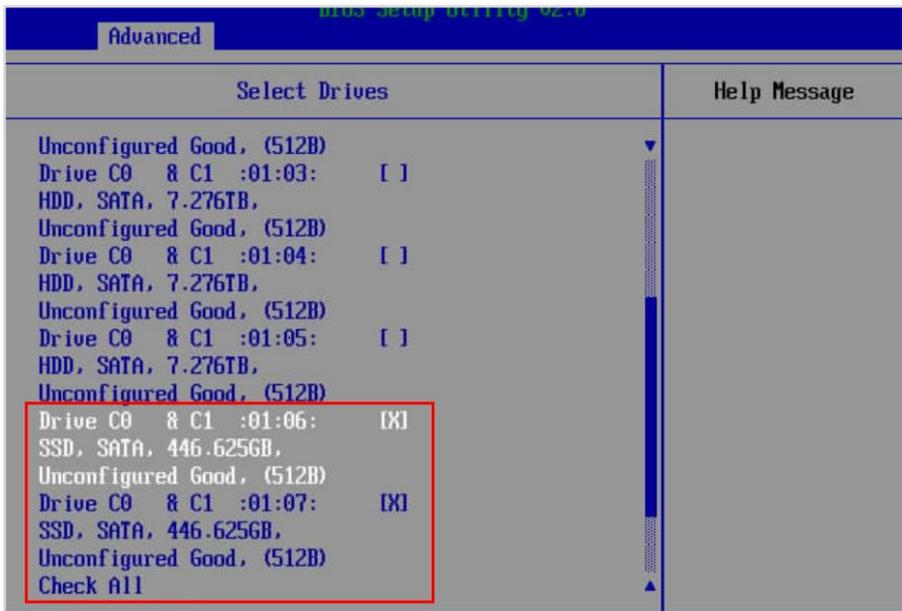
步骤 2 依次进入“Main Menu”、“Configuration Management”、“Create Virtual Drive”，进入 RAID 创建页面。

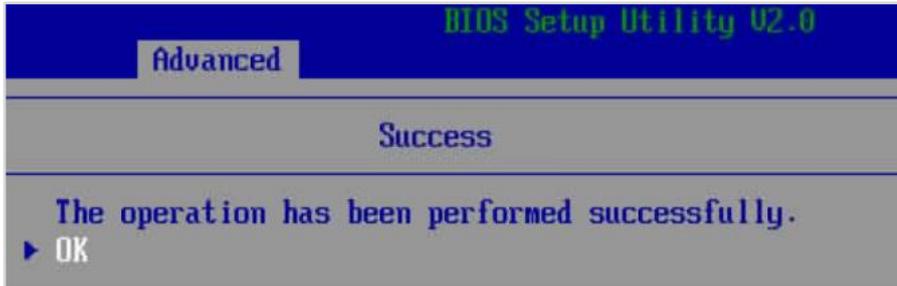


步骤 3 修改 “Select RAID Level” 为 RAID1。

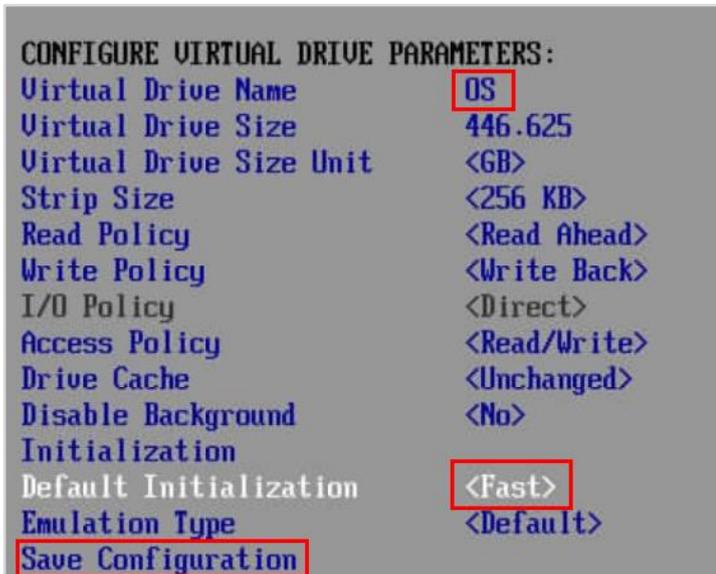


步骤 4 在 “Select Drives” 中选择两块 SSD 磁盘，按空格选中硬盘，然后将光标移动到最下方的 “Apply Changes”，按回车键，确认选择，然后选择 OK，按回车键。

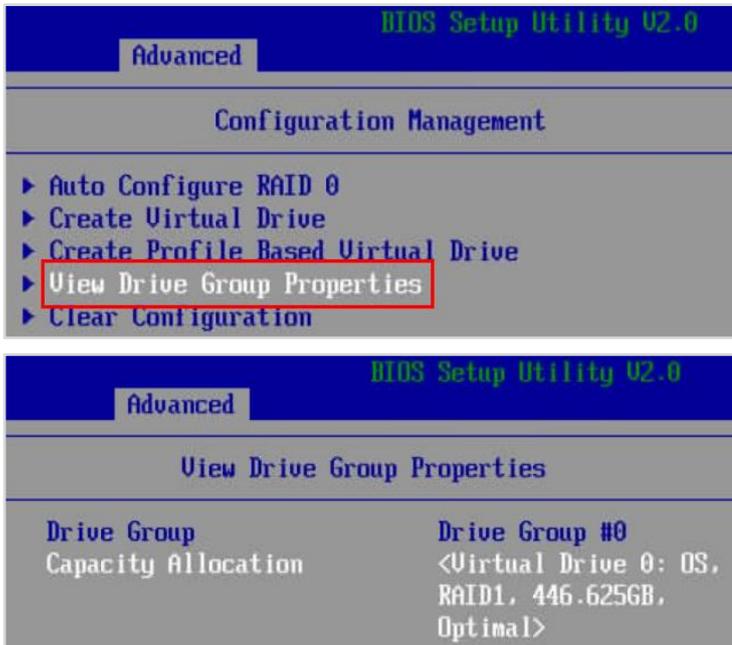




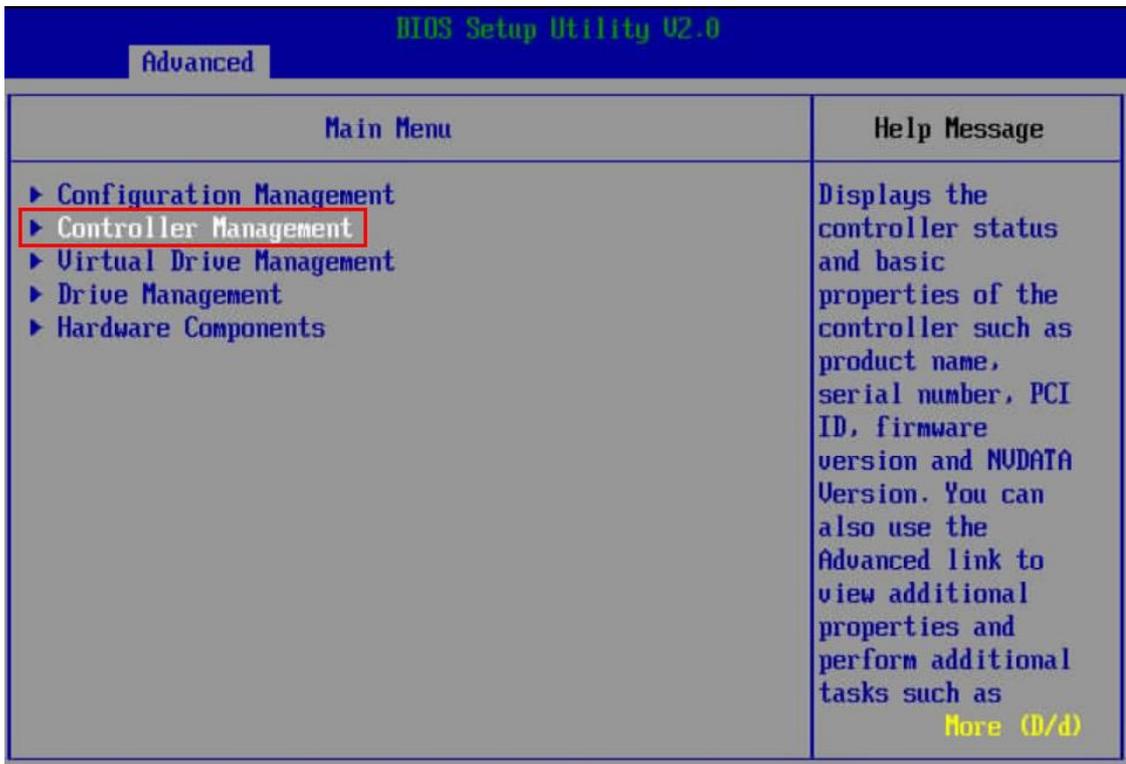
- 步骤 5 设置“Virtual Drive Name”为“OS”，设置“Default Initialization”为“Fast”。然后选择“Save Configuration”，按回车键；按空格选中“Confirm”，选择“Yes”，然后回车；成功页中继续选择“OK”，按回车键。



步骤 6 按 Esc 键返回上一级，选择“View Drive Group Properties”，按回车键进入，可以看到有一个已经配置好的 RAID1。



步骤 7 返回到“Main Menu”，将光标移动到“Controller Management”。



步骤 8 按回车键，进入“Controller Management”，将光标移动到“Select Boot Device”。按回车键，选择新建的 RAID1。

BIOS Setup Utility V2.0

Advanced

Controller Management	Help Message	
<b>BASIC PROPERTIES:</b> Product Name AVAGO MegaRAID SAS Serial Number SKD1416560 Controller Status <Optimal> Personality Mode <RAID> <b>Select Boot Device &lt;None&gt;</b> PCI ID 0x100010x001610x100010x 9461 PCI 0x000010x00410x0010x0 Segment:Bus:Device:Function PCI Slot Number 12 Package Version 51.23.0-4723 Firmware Version 5.230.00-3767 NUDATA Version 5.2300.00-0005 Supported Device Interfaces <SAS,SATA> Drive Count [8]		Selects the primary bootable device. This setting is applicable only for legacy boot mode.

None

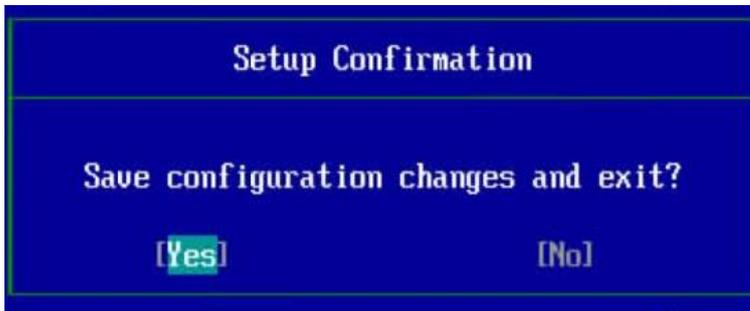
**Virtual Drive 0: OS, RAID1, 446.625GB, Optimal**

BIOS Setup Utility V2.0

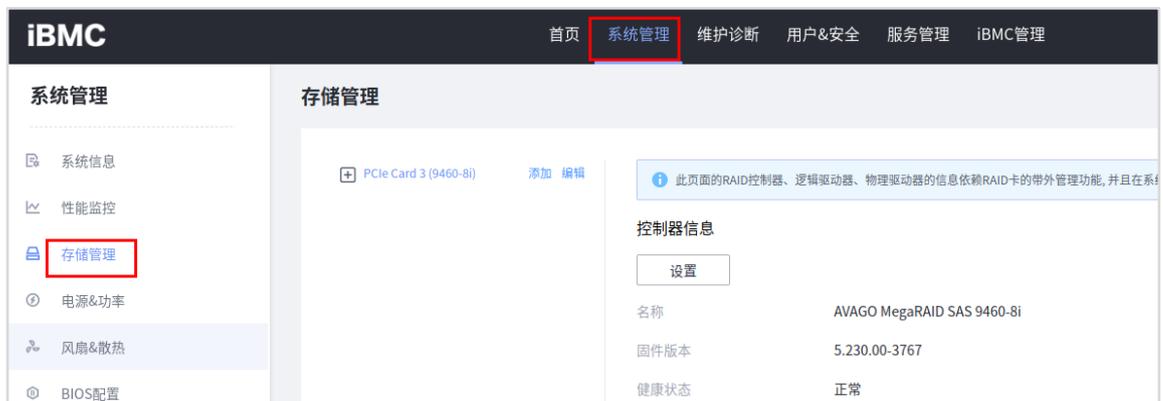
Advanced

Controller Management	Help Message	
<b>BASIC PROPERTIES:</b> Product Name AVAGO MegaRAID SAS Serial Number SKD1416560 Controller Status <Optimal> Personality Mode <RAID> <b>Select Boot Device &lt;Virtual Drive 0: OS, RAID1, 446.625GB, Optimal&gt;</b> PCI ID 0x100010x001610x100010x 9461 PCI 0x000010x00410x0010x0 Segment:Bus:Device:Function PCI Slot Number 12 Package Version 51.23.0-4723 Firmware Version 5.230.00-3767 NUDATA Version 5.2300.00-0005		Selects the primary bootable device. This setting is applicable only for legacy boot mode.

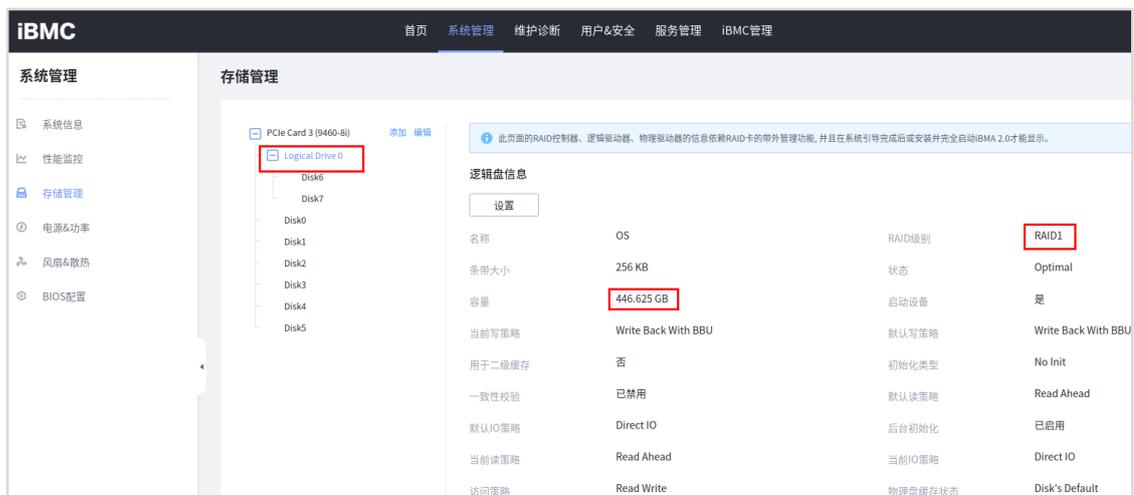
步骤 9 连续按“Ecs”键，返回到“Advanced”页签，再按“F10”，在弹出的保存配置提示窗中选择“Yes”，然后再按“Ecs”键退出 BIOS 配置页。



步骤 10 回到 iBMC 的 web 界面，点击“系统管理”-->“存储管理”，进入 iBMC 的 RAID 卡配置界面。



步骤 11 依次展开“PCIe Card”、“Logical Drive”，可以看到我们在 BIOS 中创建的 RAID1 及其用到的两块磁盘。需要注意的是，在 BIOS 中配置的 RAID 信息这里不一定立刻会显示，需要系统引导完成后或安装并完全启动 iBMA 2.0 才能显示，若无 Logical Drive 信息，请等待 5 分钟后刷新页面。



步骤 12 RAID 配置也可以通过 iBMC 的 web 界面进行配置。点击“PCIe Card”后面的“添加”。



**步骤 13** 在弹出的“创建逻辑盘”窗口中配置新 RAID 的信息。名称设置为“Data”，初始化类型设置为“Quick Init”，RAID 级别设置为 5，选中服务器中安装的 4 块 8T 容量的 SATA HDD 磁盘，然后点击“保存”，若有弹窗，点击“确定”。（磁盘编号可以从存储管理处依次点击每个磁盘查看磁盘信息，8T 容量的实际可用容量大约为 7.2T）



**步骤 14** 刷新页面后，展开“PCIe Card”，可以看到两块逻辑驱动器。至此，RAID 的两种配置操作已经完成。



## 2.3 思考题

RAID 配置项中的条带大小是什么意思？应当如何设置？

答案：条带大小是 RAID 中数据组织划分的颗粒度，根据磁盘个数和 RAID 级别，来划分颗粒度大小。

# 3 openEuler 操作系统的安装和操作

---

## 3.1 实验介绍

### 3.1.1 关于本实验

本实验介绍单台服务器的系统安装部署操作，以 openEuler 为例进行演示。通过本实验，帮助读者掌握机房中单台服务器的系统安装方法。

### 3.1.2 实验目的

- 掌握 Atlas 800（型号 3000）服务器使用虚拟光驱安装操作系统的方法；
- 掌握 openEuler 操作系统的安装配置。

### 3.1.3 实验所需软件

表3-1 实验所需软件

软件	下载地址
openEuler-22.03-LTS -aarch64-dvd.iso	<a href="https://mirrors.tuna.tsinghua.edu.cn/openeuler/openEuler-22.03-LTS/ISO/aarch64/openEuler-22.03-LTS-aarch64-dvd.iso">https://mirrors.tuna.tsinghua.edu.cn/openeuler/openEuler-22.03-LTS/ISO/aarch64/openEuler-22.03-LTS-aarch64-dvd.iso</a>

## 3.2 实验任务配置

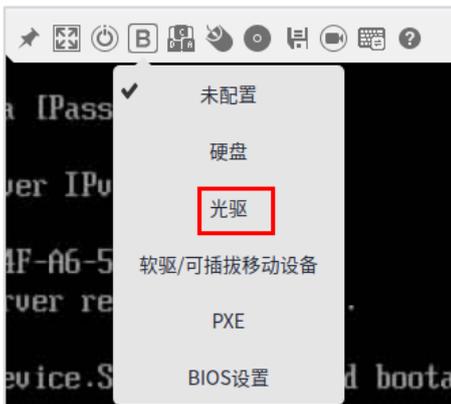
### 3.2.1 操作系统安装

步骤 1 登录服务器的 iBMC，在首页中点击“启动虚拟控制台”，选择“HTML5 集成远程控制台（共享）”。

---



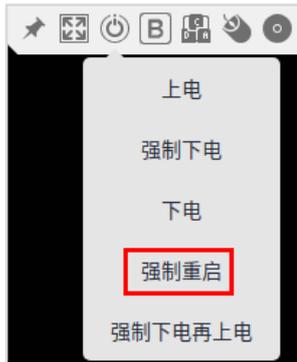
步骤 2 在控制台上方点击 Boot 启动项图标 “B”，选择“光驱”。



步骤 3 点击光盘图标 “CD/DVD”，在弹出的窗口中点击“镜像文件”后面的省略号，选择本地已下载的 openEuler-22.03-LTS-aarch64-dvd.iso，然后点击“连接”。等待“连接”变成“断开”时，表示连接成功。

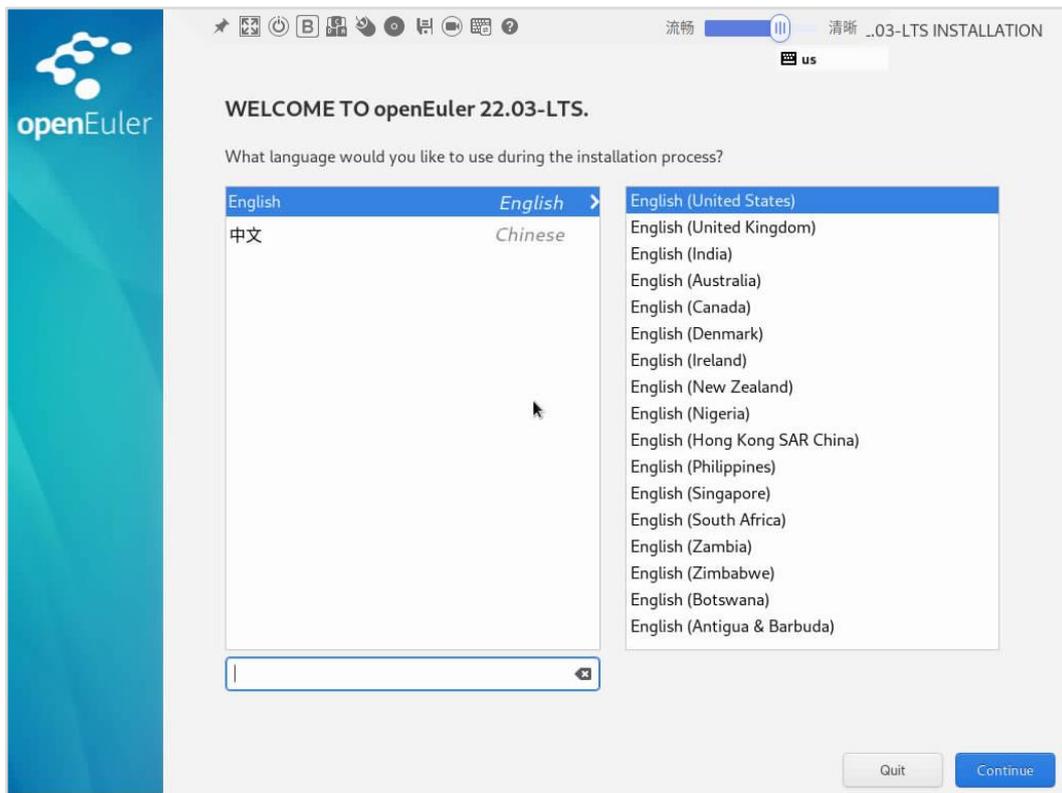


步骤 4 点击电源图标 “Power”，选择“强制重启”，在弹窗中点击“确定”，将服务器重启。

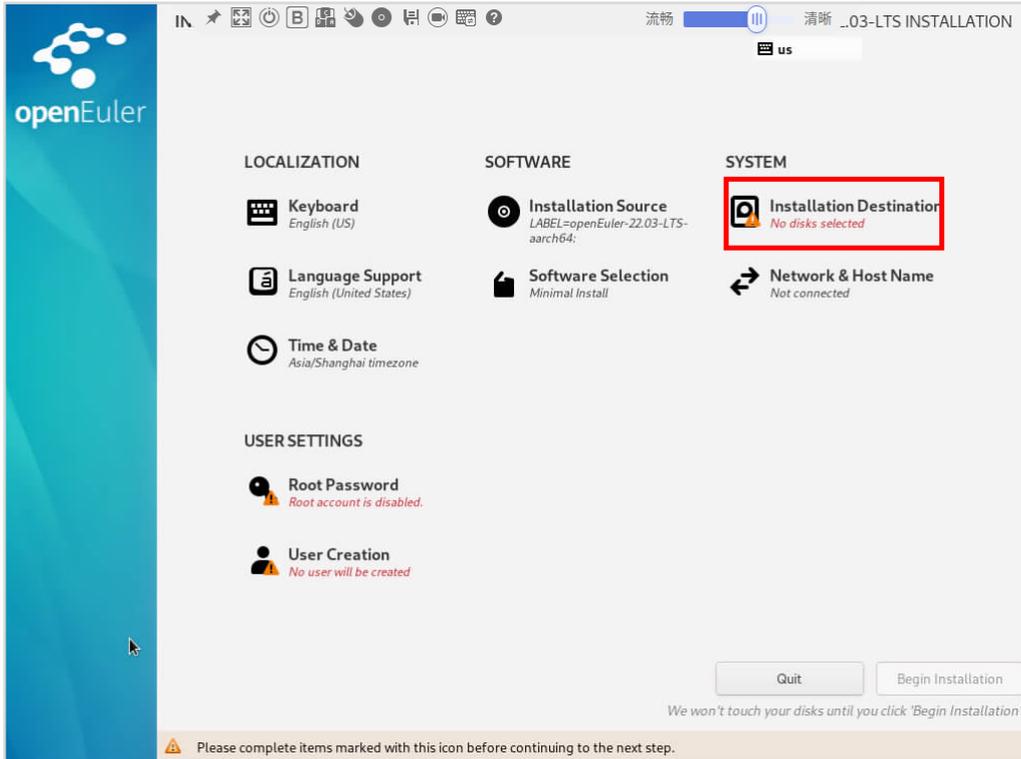


步骤 5 等待服务器重启进入系统安装界面，按向上键，将光标移动到安装系统“Install openEuler 22.03-LTS”选项，然后按回车，启动系统安装。

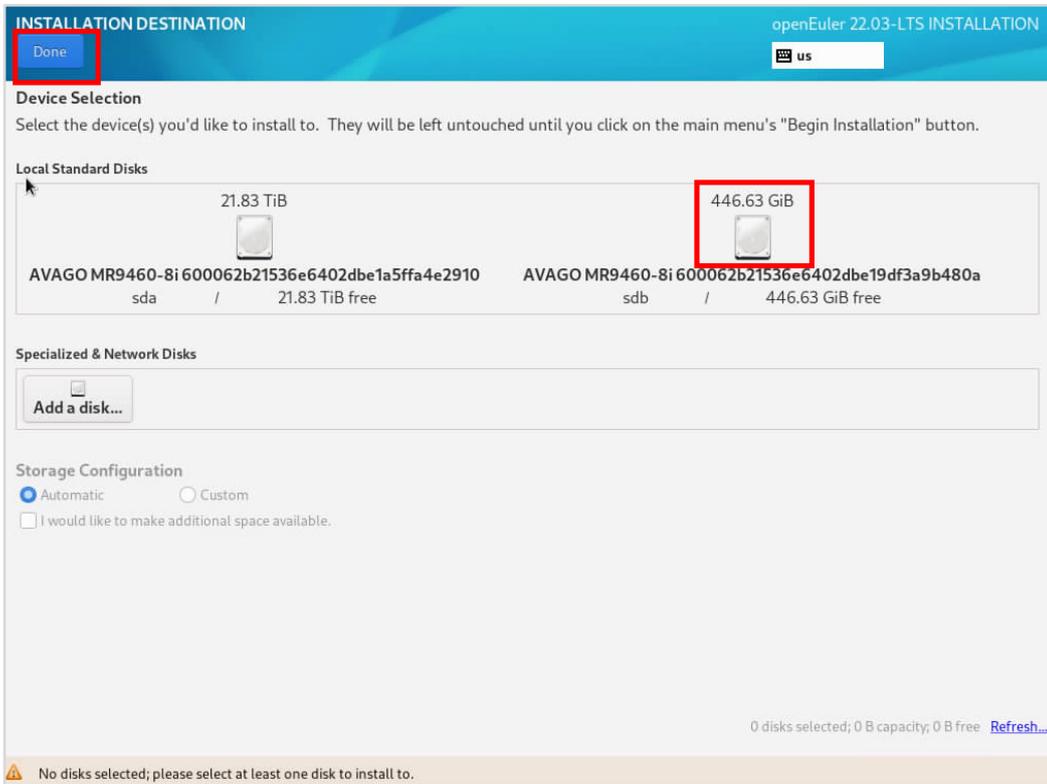
步骤 6 等待跳出安装界面，在安装语言选择界面中点击右下角的“Continue”。



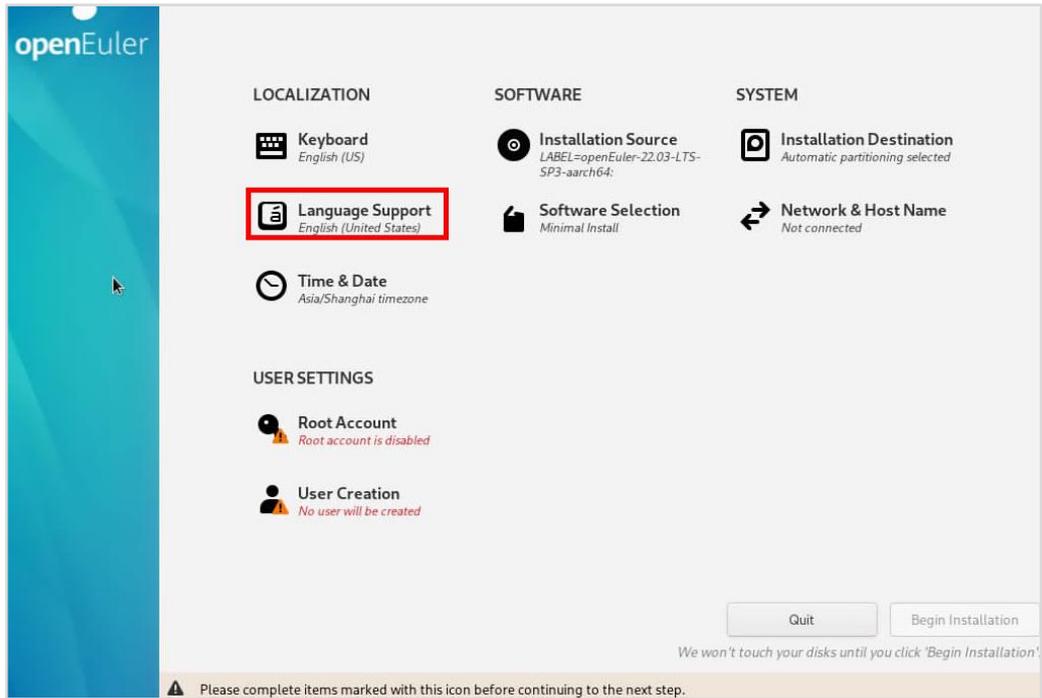
步骤 7 点击“Installation Destination”选项，进入安装位置选择页。



步骤 8 勾选容量为 446GiB 的磁盘，将系统安装到两块 SSD 组成的 RAID1 中。其他参数保持默认，点击左上角的“Done”。



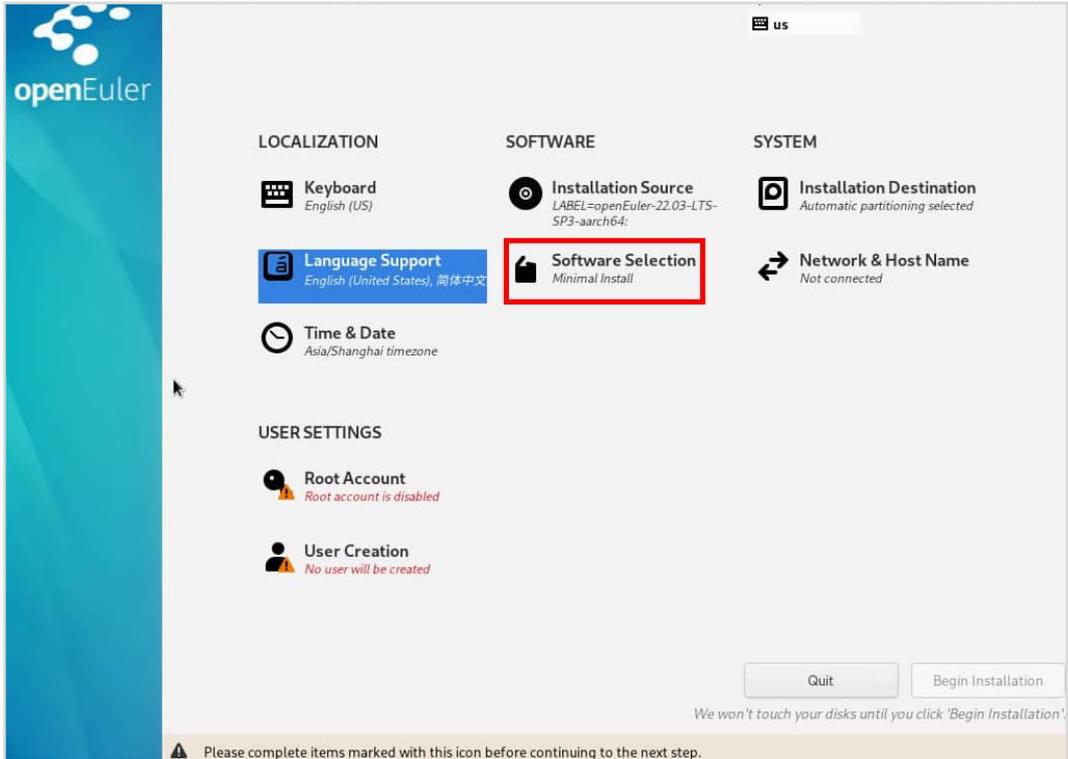
步骤 9 点击“Language Support”选项，进入语言选择页。



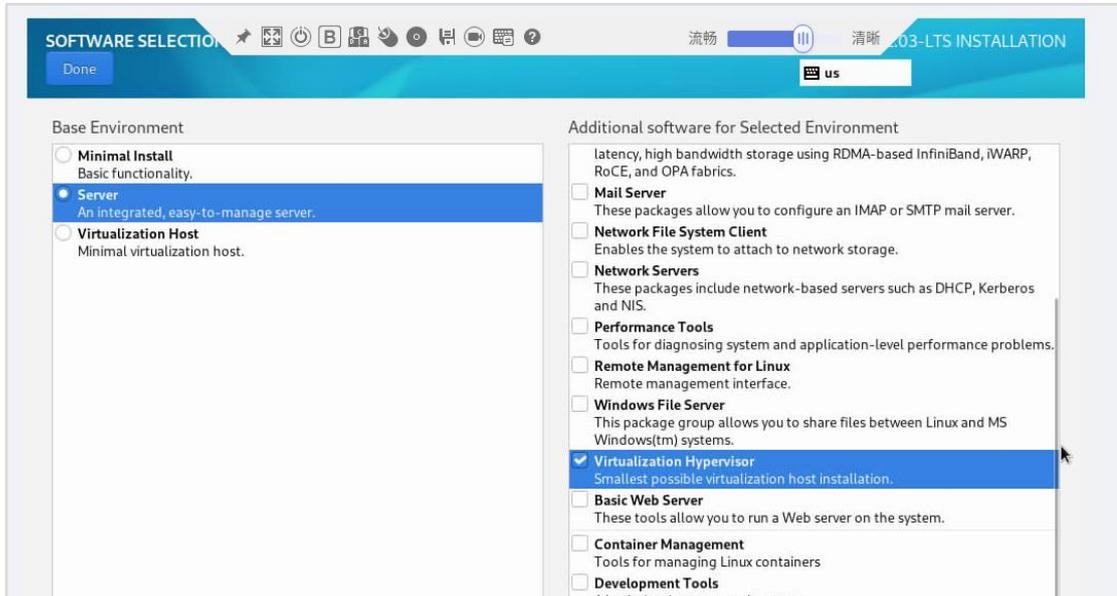
步骤 10 勾选中文中的“简体中文”，然后点击左上角的“Done”。



步骤 11 点击“Software Selection”，进入软件安装选择页。



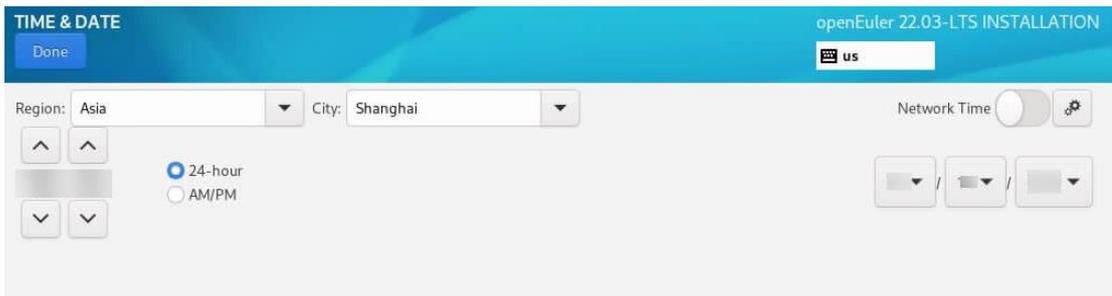
步骤 12 选择“Server”，然后在右边选择 Virtualization Hypervisor，再点击左上角的“Done”。



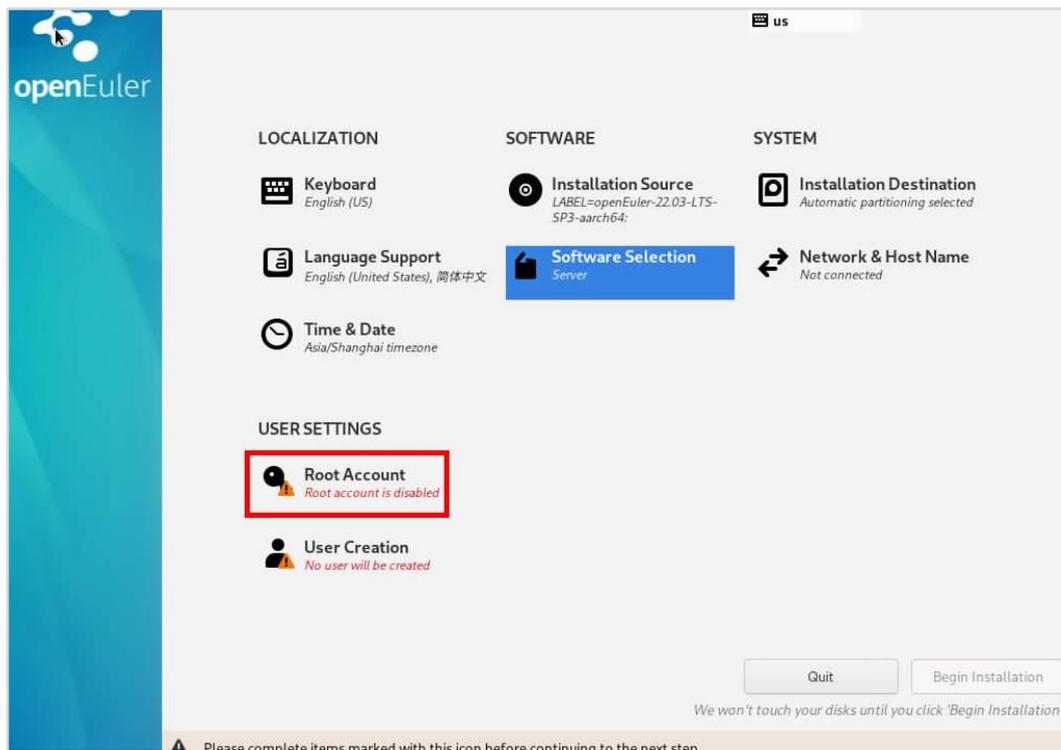
步骤 13 点击“Time & Date”，进入时间配置项。



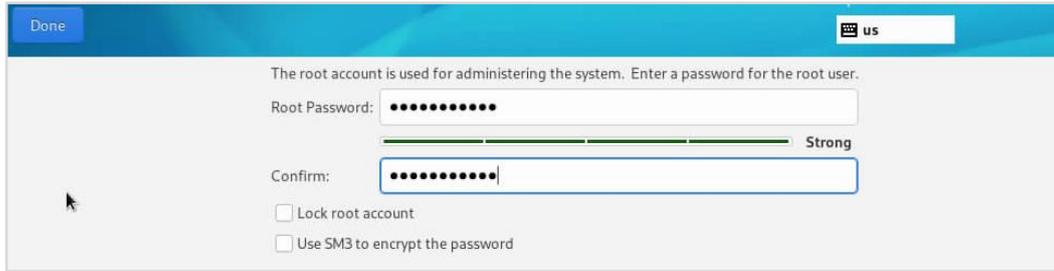
步骤 14 根据当前时间修改操作系统时间，在生产环境部署时，也可以开启“Network Time”开关，配置 NTP 时间。



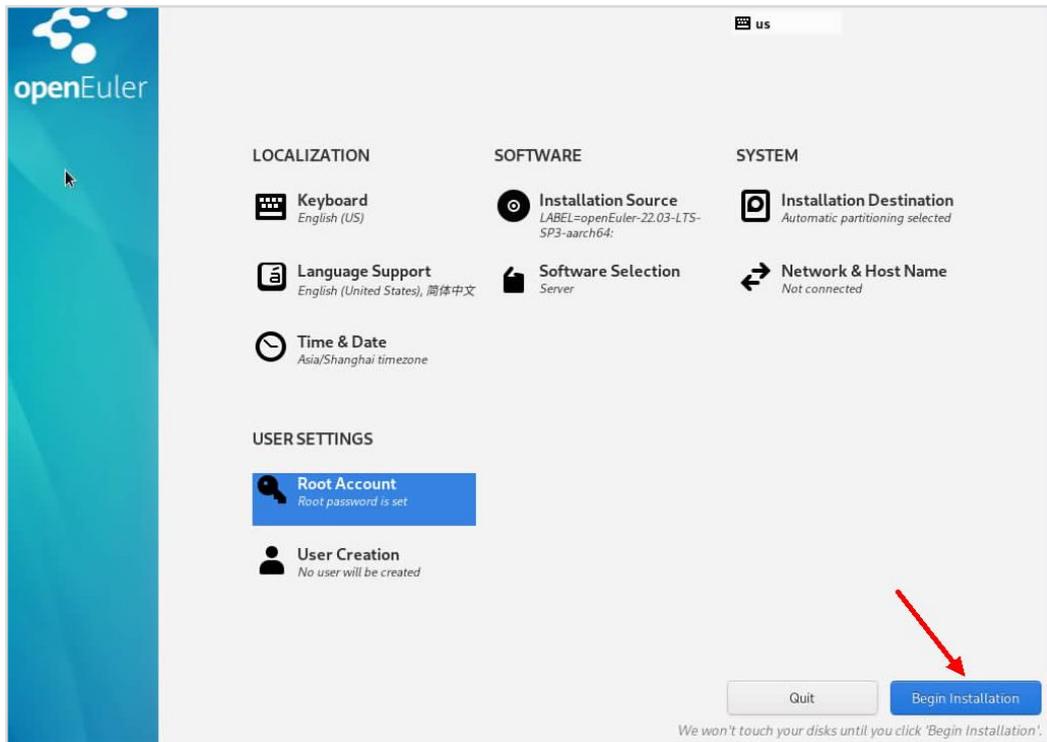
步骤 15 点击“Root Account”，进入 root 用户设置页。



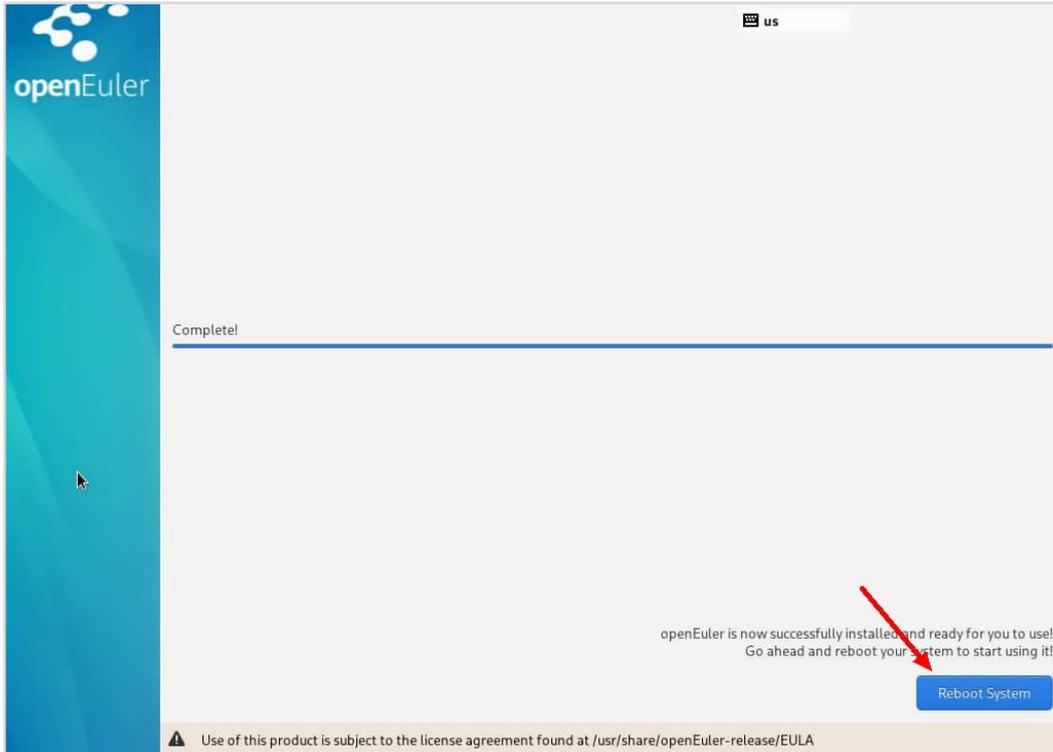
步骤 16 在两个文本框中输入自行定义的 root 密码，请设置高复杂度密码，否则不通过。设置完成后点击左上角的“Done”。



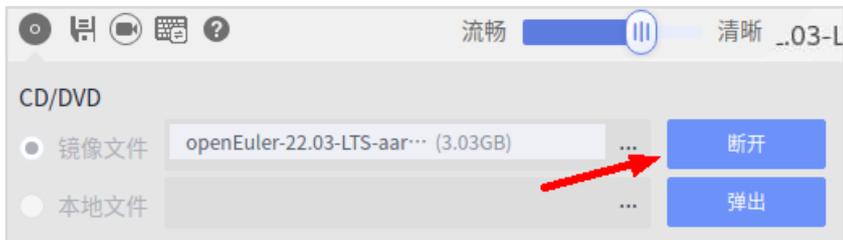
步骤 17 点击右下角的“Begin Installation”，开始安装操作系统。



步骤 18 等待安装完成后，点击右下角的“Reboot System”将服务器重启。

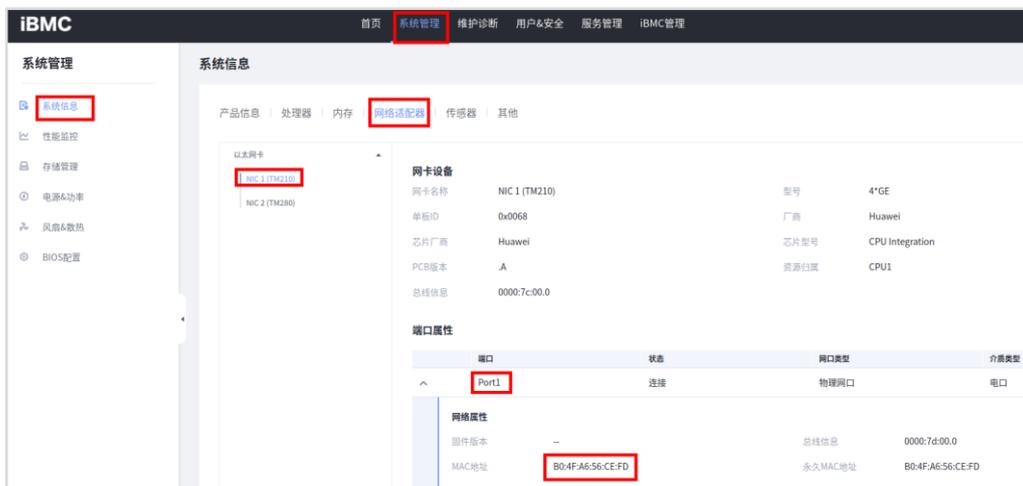


步骤 19 在虚拟光驱处断开系统 ISO 文件的连接。



步骤 20 重启完成后，在登录界面输入登录用户名 root 和设置的 root 密码，登录系统。

步骤 21 回到服务器的 iBMC 管理界面，点击“系统管理”-->“系统信息”-->“网络适配器”，确认当前已连接的卡和网口，并记住已连接网口的 MAC 地址。



步骤 22 输入如下命令检查系统网络信息。

```
[root@localhost ~]# nmcli general status      #检查网络状态
[root@localhost ~]# nmcli connection show    #显示所有连接
```

```
[root@localhost ~]# nmcli general status
STATE      CONNECTIVITY  WIFI-HW  WIFI      WWAN-HW  WWAN
connecting none          enabled  enabled   enabled   enabled
[root@localhost ~]# nmcli connection show
NAME      UUID                                  TYPE      DEVICE
enp125s0f0 9d6477f1-04c9-4959-8249-10984afa707e ethernet --
enp125s0f1 a14694dd-50a6-45b6-b260-2f96dd618134 ethernet --
enp125s0f2 363b219c-b141-4ea1-b8f8-dbb16090511e ethernet --
enp125s0f3 016980d9-43ba-43b0-924e-13d1fe1028f0 ethernet --
enp189s0f0 ee7bf81c-0714-4243-a2f5-e563abbb3b52 ethernet --
enp189s0f1 cc71f3bf-7ef2-4645-93a8-3081af3e114f ethernet --
enp189s0f2 5093664f-7595-4454-be10-a8363bbe1144 ethernet --
enp189s0f3 052d7355-7865-4a03-8ef2-8752bf3da20c ethernet --
```

步骤 23 以上一步中查到的 enp125s0f0 网卡为例，输入如下命令查看此网卡的 MAC 地址信息。

```
[root@localhost ~]# ifconfig enp125s0f0
```

```
[root@localhost ~]# ifconfig enp125s0f0
enp125s0f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether b0:4f:a6:56:ce:fd txqueuelen 1000 (Ethernet)
    RX packets 217 bytes 17151 (16.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 62 bytes 9652 (9.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

步骤 24 参照上面步骤，确认已连接网口的名称，输入如下命令编辑网卡配置文件。（以 enp125s0f0 网卡为例）

```
[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-enp125s0f0
#按 i 进入插入模式，修改如下内容项
BOOTPROTO=none
IPADDR=192.168xxxxx      #新增此项，填写规划的 IP 地址
NETMASK=255.255.255.0   #新增此项，填写规划的子网掩码
GATEWAY=192.168xxxx     #新增此项，填写规划的子网网关
#修改完毕后按 Esc 退出插入模式，输入“:wq”保存并退出。
```

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6_INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=enp125s0f0
UUID=9d6477f1-04c9-4959-8249-10984afa707e
DEVICE=enp125s0f0
ONBOOT=yes
IPADDR=192.168.1.1
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
```

步骤 25 输入如下命令重启步骤 24 中配置的网卡，并查看网卡配置是否生效。

```
[root@localhost ~]# nmcli device disconnect enp125s0f0
[root@localhost ~]# nmcli connection up id enp125s0f0
[root@localhost ~]# ifconfig enp125s0f0
```

```
root@localhost ~]# nmcli device disconnect enp125s0f0
Device 'enp125s0f0' successfully disconnected.
root@localhost ~]# nmcli connection up id enp125s0f0
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkMa
nager/ActiveConnection/14)
root@localhost ~]# ifconfig enp125s0f0
enp125s0f0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 15.1.1.1 netmask 255.255.255.0 broadcast 15.1.1.255
    inet6 fe80::b24f:a6ff:fe56:cefd prefixlen 64 scopeid 0x20<link>
    ether b0:4f:a6:56:ce:fd txqueuelen 1000 (Ethernet)
    RX packets 863 bytes 77641 (75.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 219 bytes 31826 (31.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

步骤 26 ping 网关地址，确认网络是否可达。

```
root@localhost ~]# ping 15.1.1.1
PING 15.1.1.1 (15.1.1.1) 56(84) bytes of data.
64 bytes from 15.1.1.1: icmp_seq=1 ttl=255 time=11.4 ms
64 bytes from 15.1.1.1: icmp_seq=2 ttl=255 time=1.96 ms
^C
--- 15.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.962/6.705/11.449/4.743 ms
```

步骤 27 从实验操作桌面 ping 系统配置的 IP 地址，确认网络的连通性。

```
admin@ADMIN1:~/Desktop$ ping 15.1.1.1
PING 15.1.1.1 (15.1.1.1) 56(84) bytes of data.
64 bytes from 15.1.1.1: icmp_seq=1 ttl=64 time=0.828 ms
64 bytes from 15.1.1.1: icmp_seq=2 ttl=64 time=0.158 ms
64 bytes from 15.1.1.1: icmp_seq=3 ttl=64 time=0.110 ms
^C
--- 15.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 47ms
rtt min/avg/max/mdev = 0.110/0.365/0.828/0.328 ms
```

步骤 28 确认网络正常后，关闭浏览器，本实验至此结束。

### 3.2.2 磁盘分区与逻辑卷配置

步骤 1 在实验桌面使用 ssh 工具登录 openEuler 操作系统，输入如下命令查看磁盘信息。

```
[root@localhost ~]# fdisk -l #可以看到有一个/dev/sdb，容量为 21.83TiB 的磁盘，这是在服务器中配
置的 RAID5 时创建的逻辑卷。磁盘编号也可能不是 sdb，以实际情况为准。
```

步骤 2 对磁盘进行分区，创建一个 1T 大小的分区。

```
root@localhost ~]# parted /dev/sdb #对/dev/sdb 进行操作
GNU Parted 3.4
Using /dev/sdb
```

```

Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel          #设置分区表类型为 gpt
New disk label type? gpt
(parted) mkpart          #创建分区
Partition name? []? lvm          #设置分区名称为 lvm
File system type? [ext2]? ext4    #设置文件系统类型为 ext4 (并不是格式化为 ext4 文件系统)
Start? 0          #设置分区起始位置
End? 1T          #设置分区结束位置
Warning: The resulting partition is not properly aligned for best performance: 34s % 1536s != 0s
Ignore/Cancel? ignore          #输入 Ignore, 忽略警告
(parted) print          #打印当前分区信息
Model: AVAGO MR9460-8i (scsi)
Disk /dev/sdb: 24.0TB
Sector size (logical/physical): 512B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      17.4kB 1000GB  1000GB  ext4         lvm

(parted) q          #输入 q 退出分区对话框
Information: You may need to update /etc/fstab.
    
```

步骤 3 输入如下信息查看分区情况，并创建 PV。

```

[root@localhost ~]# fdisk -l /dev/sdb
Disk /dev/sdb: 21.83 TiB, 24003062267904 bytes, 46880980992 sectors
Disk model: MR9460-8i
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 262144 bytes / 786432 bytes
Disklabel type: gpt
Disk identifier: 7D4B153A-46C9-425F-80E7-19CC754AEA0D

Device      Start          End      Sectors   Size Type
/dev/sdb1   34 1953125000 1953124967 931.3G Linux filesystem

[root@localhost ~]# pvcreate /dev/sdb1
Device read short 78336 bytes remaining
Device read short 78336 bytes remaining
Physical volume "/dev/sdb1" successfully created.
    
```

步骤 4 输入如下信息创建 VG 和逻辑卷。

```

[root@localhost ~]# vgcreate yum /dev/sdb1          #创建名称为 yum 的 VG, 使用/dev/sdb1 的空间
Device read short 78336 bytes remaining
Device read short 78336 bytes remaining
    
```

```
Volume group "yum" successfully created  
[root@localhost ~]# lvcreate -n yum -L 500G yum  
Logical volume "yum" created.
```

步骤 5 输入如下命令格式化逻辑卷。

```
[root@localhost ~]# mkfs.ext4 /dev/yum/yum
```

### 3.3 思考题

本实验展示的是单台服务器的系统暗安装，若有多台服务器，如何批量安装操作系统？

答案：可以部署 PXE 服务器，通过 PXE 引导安装操作系统。

# 4 常见服务的部署

## 4.1 实验介绍

### 4.1.1 关于本实验

本实验介绍 openEuler 操作系统下集中常见的服务的部署，包括 yum 仓库、邮箱服务器等。

### 4.1.2 实验目的

- 掌握本地 yum 源和 ftp 服务器 yum 源的搭建方法；
- 掌握 postfix+dovecot+MySQL 软件框架的邮件服务器部署。

### 4.1.3 实验所需软件

表4-1 实验所需软件

软件名称	作用	下载地址
openEuler-22.03-LTS - aarch64-dvd.iso	安装操作系统	<a href="https://mirrors.tuna.tsinghua.edu.cn/openeuler/openEuler-22.03-LTS/ISO/aarch64/openEuler-22.03-LTS-aarch64-dvd.iso">https://mirrors.tuna.tsinghua.edu.cn/openeuler/openEuler-22.03-LTS/ISO/aarch64/openEuler-22.03-LTS-aarch64-dvd.iso</a>
openEuler-22.03-LTS-everything-aarch64-dvd.iso	制作 yum 源	<a href="https://mirrors.tuna.tsinghua.edu.cn/openeuler/openEuler-22.03-LTS/ISO/aarch64/openEuler-22.03-LTS-everything-aarch64-dvd.iso">https://mirrors.tuna.tsinghua.edu.cn/openeuler/openEuler-22.03-LTS/ISO/aarch64/openEuler-22.03-LTS-everything-aarch64-dvd.iso</a>
mysql-boost-5.7.38.tar.gz	邮件服务器后端数据库	<a href="https://mirrors.huaweicloud.com/mysql/Downloads/MySQL-5.7/mysql-boost-5.7.38.tar.gz">https://mirrors.huaweicloud.com/mysql/Downloads/MySQL-5.7/mysql-boost-5.7.38.tar.gz</a>
Thunderbird Setup 115.10.1.exe	免费开源的邮件客户端	<a href="https://download-installer.cdn.mozilla.net/pub/thunderbird/releases/115.10.1/win64/zh-CN/Thunderbird%20Setup%20115.10.1.exe">https://download-installer.cdn.mozilla.net/pub/thunderbird/releases/115.10.1/win64/zh-CN/Thunderbird%20Setup%20115.10.1.exe</a>

## 4.2 部署基于 FTP 的私有 yum 源

步骤 1 使用工具将 openEuler-22.03-LTS-aarch64-dvd.iso 上传到 openEuler 操作系统的/root 目录中。

步骤 2 使用 ssh 工具登录 openEuler 操作系统，输入如下命令挂载 ISO 文件。

```
[root@localhost ~]# mkdir /mnt/cdrom
[root@localhost ~]# mount -o loop /root/openEuler-22.03-LTS-aarch64-dvd.iso /mnt/cdrom/
mount: /mnt/cdrom: WARNING: source write-protected, mounted read-only.
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	4.0M	0	4.0M	0%	/dev
tmpfs	252G	0	252G	0%	/dev/shm
tmpfs	101G	12M	101G	1%	/run
tmpfs	4.0M	0	4.0M	0%	/sys/fs/cgroup
/dev/mapper/openeuler-root	69G	5.3G	60G	9%	/
tmpfs	252G	0	252G	0%	/tmp
/dev/sda2	974M	147M	761M	17%	/boot
/dev/sda1	599M	6.5M	593M	2%	/boot/efi
/dev/mapper/openeuler-home	365G	28K	346G	1%	/home
/dev/loop0	3.2G	3.2G	0	100%	/mnt/cdrom

步骤 3 编辑服务器 yum 配置文件，新增本地 yum 配置。

```
[root@localhost ~]# cd /etc/yum.repos.d/
[root@localhost yum.repos.d]# mv openEuler.repo openEuler.repo.bak
[root@localhost yum.repos.d]# vim cdrom.repo
#按 i 进入插入模式，输入如下内容：
[local]
name=local
baseurl=file:///mnt/cdrom
enabled=1
gpgcheck=0
#按 Esc 退出插入模式，输入“:wq”保存并退出。
[root@localhost yum.repos.d]# yum clean all
[root@localhost yum.repos.d]# yum list all
```

步骤 4 安装 FTP 服务，启动 FTP，并设置 FTP 开机自启动。

```
[root@localhost yum.repos.d]# cd
[root@localhost ~]# yum -y install vsftpd
[root@localhost ~]# systemctl start vsftpd.service
[root@localhost ~]# systemctl enable vsftpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service →
/usr/lib/systemd/system/vsftpd.service.
```

步骤 5 修改 vsftp 配置文件，允许匿名用户访问，并关闭写权限。

```
[root@localhost ~]# vim /etc/vsftpd/vsftpd.conf
#将 anonymous_enable 值设置为 YES, 将 write_enable 设置为 NO
[root@localhost ~]# systemctl restart vsftpd
```

步骤 6 关闭防火墙和 selinux。

```
[root@localhost ~]# systemctl stop firewalld.service
[root@localhost ~]# setenforce 0
[root@localhost ~]# vim /etc/selinux/config
#SELINUX=enforcing"修改为"SELINUX=disabled"。
```

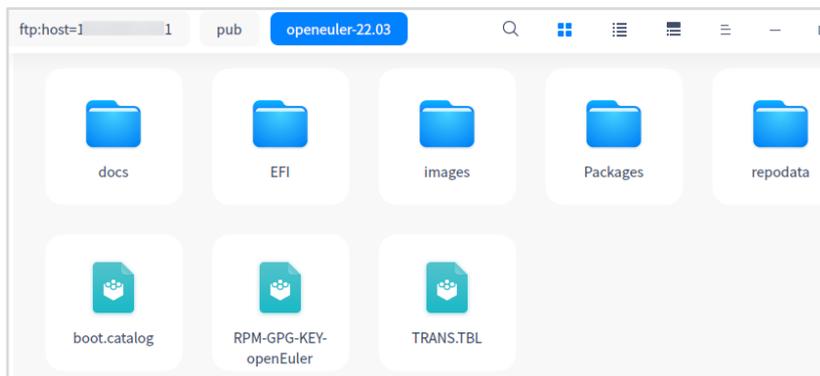
步骤 7 将 3.2.2 实验中创建的逻辑卷挂载到/var/ftp/pub/openeuler-22.03 目录，并设置自动挂载。

```
[root@localhost ~]# mkdir /var/ftp/pub/openeuler-22.03/
[root@localhost ~]# mount /dev/mapper/yum-yum /var/ftp/pub/openeuler-22.03/
[root@localhost ~]# vim /etc/fstab
#在文件最后新增如下一行信息：
/dev/mapper/yum-yum /var/ftp/pub/openeuler-22.03/ ext4 defaults 0 0
```

步骤 8 上传 openEuler-22.03-LTS-everything-aarch64-dvd.iso 到 openEuler 操作系统的/root 目录下。

步骤 9 将 openEuler-22.03-LTS-everything-aarch64-dvd.iso 文件挂载到/mnt/cdrom 目录，并将其中的内容复制到/var/ftp/pub/openeuler-22.03 目录。

```
[root@localhost ~]# umount /mnt/cdrom
[root@localhost ~]# mount -o loop openEuler-22.03-LTS-everything-aarch64-dvd.iso /mnt/cdrom/
[root@localhost ~]# cp -rf /mnt/cdrom/* /var/ftp/pub/openeuler-22.03/
#等待命令执行完毕，此时可以使用实验桌面打开文件管理器，在地址栏输入
ftp://192.168.xxx.xxx/pub/openeuler-22.03/查看内容，此处 IP 地址更换为 openEuler 操作系统的 IP 地址。
若有弹窗，请选择匿名连接
```



步骤 10 输入如下命令，删除原有的 yum 配置文件，新增指向到 ftp 服务器的 yum 配置文件，并确认 yum 源状态正常。

```
[root@localhost ~]# cd /etc/yum.repos.d/
[root@localhost yum.repos.d]# mv cdrom.repo ftp.repo
```

```
[root@localhost yum.repos.d]# vim ftp.repo
#清空内容，输入如下内容：
[cdrom]
name=cdrom
baseurl=ftp://192.168.xxx.xxx/pub/openeuler-22.03/
enabled=1
gpgcheck=0
#在文件中输入上述内容，其中 IP 地址请更换为操作系统 IP。
[root@localhost yum.repos.d]# yum clean all
6 files removed
[root@localhost yum.repos.d]# yum list all
#若有如下图所示能显示出安装包，源是 ftp，则表示 yum 源配置正常
```

```
xz-help.noarch          5.2.5-3.oe2203sp3      ftp
xz-lzma-compat.aarch64 5.2.5-3.oe2203sp3      ftp
yelp.aarch64           2:3.38.3-1.oe2203sp3   ftp
yelp-xsl.noarch        3.38.3-1.oe2203sp3     ftp
yp-tools.aarch64       4.2.3-6.oe2203sp3      ftp
yp-tools-help.noarch   4.2.3-6.oe2203sp3      ftp
ypbind.aarch64         3:2.7.2-2.oe2203sp3    ftp
ypbind-help.noarch     3:2.7.2-2.oe2203sp3    ftp
ypserv.aarch64         4.2-2.oe2203sp3        ftp
```

## 4.3 部署邮箱服务器

### 4.3.1 搭建 DNS 服务器

步骤 1 使用 ssh 工具登录 openEuler 系统，设置主机名，并更新。（主机名以 mail.huawei.com 为例）

```
[root@localhost ~]# hostnamectl set-hostname mail.huawei.com
[root@localhost ~]# echo "192.168.xxx.xxx mail.huawei.com" >> /etc/hosts #IP 地址更换为操作系统 IP
[root@localhost ~]# bash
Welcome to 5.10.0-182.0.0.95.oe2203sp3.aarch64
.....
[root@mail ~]# #可以看到主机名已经更新为 mail
```

步骤 2 安装 bind。

```
[root@mail ~]# dnf -y install bind
#安装完成后默认在/etc 目录下生成以下文件
./named
./named.conf # 主配置文件
./named.rfc1912.zones # 区域配置文件
./named.root.key
```

步骤 3 编辑 named.conf 主配置文件。

```
[root@mail ~]# vim /etc/named.conf
#修改如下内容，并保存退出
```

```
listen-on port 53 { 192.168.xxx.xxx; };      #修改为操作系统 IP 地址
allow-query    { any; };                    #将 localhost 改为 any
[root@mail ~]# named-checkconf             #配置检查, 若无反馈则表示正确
```

```
options {
listen-on port 53 { 192.168.10.10; };
listen-on-v6 port 53 { ::1; };
directory     "/var/named";
dump-file     "/var/named/data/cache_dump.db";
statistics-file "/var/named/data/named_stats.txt";
memstatistics-file "/var/named/data/named_mem_stats.txt";
secroots-file  "/var/named/data/named.secroots";
recursing-file "/var/named/data/named.recursing";
allow-query   { any; };
}
```

步骤 4 编辑 DNS 区域配置文件, 添加正向区域配置、反向区域配置。

```
[root@mail ~]# vim /etc/named.rfc1912.zones
#在最后增加如下内容, 并保存退出。(域名以 huawei.com 为例)
#正向区域配置
zone "huawei.com" IN {
    type master;
    file "huawei.com.zone";
    allow-update { none; };
};
#反向区域配置
zone "10.168.192.in-addr.arpa" IN {
    type master;
    file "huawei.com.local";
    allow-update { none; };
};
```

#此处的 10 为本实验手册编辑时创建的 192.168.10.X 网段的第三个字段 10, 请读着根据实际情况填写

步骤 5 配置正向解析与反向解析, 并启动 named 服务。

```
[root@mail ~]# cd /var/named/
[root@mail named]# ls
data dynamic named.ca named.empty named.localhost named.loopback slaves
[root@mail named]# cp -p named.localhost huawei.com.zone      #正向区域解析文件
[root@mail named]# cp -p named.localhost huawei.com.local     #反向区域解析文件
[root@mail named]# vim huawei.com.zone
#修改其中默认的 A 记录地址为操作系统地址 192.168.xxx.xxx
#添加三条记录, 并保存退出
www  IN A    192.168.XXX.XXX      #IP 地址改为操作系统 IP
mail IN A    192.168.XXX.XXX      #IP 地址改为操作系统 IP
mx   MX 5    192.168.XXX.XXX      #IP 地址改为操作系统 IP
[root@mail named]# vim huawei.com.local
#添加一条反向解析
XXX IN PTR  www.huawei.com.    #XXX 换成操作系统 IP 后一个字段, 比如 192.168.10.100, 此处就写 100, 注意后面还有一个点
```

```
[root@mail named]# cd
[root@mail ~]# systemctl restart named
[root@mail ~]# systemctl enable named
```

```
$TTL 1D
@      IN SOA      e   rname.invalid. (
                                0      ; serial
                                1D     ; refresh
                                1H     ; retry
                                1W     ; expire
                                3H )   ; minimum

NS     e
A      192.168.1.1
AAAA   ::1
www IN A 192.168.1.1
mail IN A 192.168.1.1
mx  MX 5 192.168.1.1
```

步骤 6 关闭防火墙。

```
[root@mail ~]# systemctl stop firewalld
[root@mail ~]# systemctl disable firewalld
```

步骤 7 将本机的 DNS 指向 DNS 服务器。

```
[root@mail ~]# vim /etc/resolv.conf
#新增一行
nameserver 192.168.xxx.xxx #IP 地址为操作系统 IP
```

步骤 8 测试 DNS 服务器能否正常解析。

```
[root@mail ~]# nslookup mail.huawei.com
Server:          192.168.XXX.XXX #此处及后面的 IP 返回为 openEuler 操作系统 IP
Address:         192.168.XXX.XXX#53

Name:   mail.huawei.com
Address: 192.168.XXX.XXX
```

## 4.3.2 源码编译安装 MySQL

步骤 1 将 mysql-boost-5.7.38.tar.gz 压缩包上传到操作系统的/root 目录下。

步骤 2 执行如下命令解压 MySQL 源码包。

```
[root@mail ~]# tar -zxvf mysql-boost-5.7.38.tar.gz
```

步骤 3 执行如下命令编译安装 MySQL。

```
[root@mail ~]# dnf -y install gcc gcc-c++ automake zlib zlib-devel bzip2 bzip2-devel bzip2-libs readline
readline-devel bison ncurses ncurses-devel libaio-devel openssl openssl-devel gmp gmp-devel mpfr
mpfr-devel libmpc libmpc-devel bison* ncurses* bzip2 wget libtirpc libtirpc-devel ncurses-devel rpcgen
cmake #安装依赖
[root@mail ~]# cd mysql-5.7.38/
[root@mail mysql-5.7.38]# cmake . -DCMAKE_INSTALL_PREFIX=/usr/local/mysql \
```

```

-DMYSQL_DATADIR=data/data \
-DSYSCONFDIR=/etc \
-DWITH_INNOBASE_STORAGE_ENGINE=1 \
-DWITH_PARTITION_STORAGE_ENGINE=1 \
-DWITH_FEDERATED_STORAGE_ENGINE=1 \
-DWITH_BLACKHOLE_STORAGE_ENGINE=1 \
-DWITH_MYISAM_STORAGE_ENGINE=1 \
-DENABLED_LOCAL_INFILE=1 \
-DENABLE_DTRACE=0 \
-DDEFAULT_CHARSET=utf8mb4 \
-DDEFAULT_COLLATION=utf8mb4_general_ci \
-DWITH_EMBEDDED_SERVER=1 \
-DDOWNLOAD_BOOST=1 \
-DWITH_BOOST=/root/mysql-5.7.38/boost/boost_1_59_0
[root@mail mysql-5.7.38]# make -j$(nproc)
[root@mail mysql-5.7.38]# make install
    
```

#### 步骤 4 创建 MySQL 用户和组。

```
[root@mail mysql-5.7.38]# useradd mysql
```

#### 步骤 5 创建 MySQL 相关目录，并配置权限。

```

[root@computing mysql-5.7.38]# cd /usr/local/mysql/
[root@mail mysql]# mkdir -p data/{log,data,run}
[root@mail mysql]# chown -R mysql:mysql data
[root@mail mysql]# chown -R mysql:mysql /usr/local/mysql/
    
```

#### 步骤 6 初始化 MySQL。（请记住最后生成的一串字符串，这个是 MySQL 数据库 root 用户的初始密码）

```
[root@mail mysql]# bin/mysqld --initialize --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data/data --user=mysql
```

```

.168498Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp.
.528475Z 0 [Warning] InnoDB: New log files created, LSN=45790
.589307Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
.595753Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this
340-b04fa656cefd.
.596266Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened
.909426Z 0 [Warning] A deprecated TLS version TLSv1 is enabled. Please use TLSv1.2 or higher.
.909440Z 0 [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
.909925Z 0 [Warning] CA certificate ca.pem is self signed.
.961104Z 1 [Note] A temporary password is generated for root@localhost: qv,b?Ipj+0fU
    
```

#### 步骤 7 创建“mysql.log”和“mysql.pid”文件，赋予“mysql”用户及用户组权限。

```

[root@mail mysql]# cd /usr/local/mysql/data/log/
[root@mail log]# touch mysql.log
[root@mail log]# touch ../run/mysql.pid
[root@mail log]# chown -R mysql:mysql /usr/local/mysql/
    
```

#### 步骤 8 修改 my.cnf 中的文件路径。

```
[root@mail log]# vim /etc/my.cnf
```

```
#写入如下内容:
[client]
port=3306
socket=/usr/local/mysql/data/data/mysql.sock

[mysqld]
datadir=/usr/local/mysql/data/data
socket=/usr/local/mysql/data/data/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld_safe]
log-error=/usr/local/mysql/data/log/mysql.log
pid-file=/usr/local/mysql/data/run/mysql.pid

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
```

### 步骤 9 启动 MySQL 服务。

```
[root@mail log]# mkdir /etc/my.cnf.d
[root@mail log]# cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql
[root@mail log]# chmod +x /etc/init.d/mysql
[root@mail log]# chkconfig mysql on
[root@mail log]# systemctl start mysql
```

### 步骤 10 配置环境变量。

```
[root@mail log]# vim /etc/profile
#在文件的最后新增一行如下内容:
export PATH=/usr/local/mysql/bin:$PATH
[root@mail log]# source /etc/profile
```

### 步骤 11 建立套接字软连接，并登录 MySQL。

```
[root@mail log]# cd
[root@mail ~]# ln -s /data/data/mysql.sock /tmp/mysql.sock
[root@mail ~]# mysql -uroot -p
Enter password:          #此处输入步骤 6 中初始化 MySQL 时生成的密码，输入密码不会显示
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.38
```

```

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>                                #正常登录
    
```

**步骤 12** 修改数据库 root 用户密码，并设置权限。（其中，mypassword 需要自行定义修改，后文 MySQL 数据库密码以 mypassword 为例，遇到时请修改成此处定义的密码）

```

mysql> SET PASSWORD = PASSWORD('mypassword');
mysql> UPDATE mysql.user SET authentication_string =PASSWORD('mypassword') WHERE User='mysql';
mysql> GRANT ALL PRIVILEGES ON *.* TO mysql@localhost IDENTIFIED BY 'mypassword' WITH GRANT
OPTION;
mysql> GRANT ALL PRIVILEGES ON *.* TO mysql@"%" IDENTIFIED BY 'mypassword' WITH GRANT
OPTION;
mysql> GRANT ALL PRIVILEGES ON *.* TO root@localhost IDENTIFIED BY 'mypassword' WITH GRANT
OPTION;
mysql> GRANT ALL PRIVILEGES ON *.* TO root@"%" IDENTIFIED BY 'mypassword' WITH GRANT
OPTION;
mysql> flush privileges;
mysql> quit
Bye
    
```

**步骤 13** 验证密码。

```

[root@mail ~]# mysql -uroot -pmypassword    #请将 mypassword 修改为上一步定义的密码
#能正常登录 MySQL 即表示配置正确。
    
```

### 4.3.3 数据库内容配置

**步骤 1** 创建一个 mail 库，用于存放邮件服务相关数据，并创建邮件数据库管理员 mail\_admin。

```

mysql> CREATE DATABASE mail;
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON mail.* TO 'mail_admin'@"%" IDENTIFIED BY
'mypassword';          #此处的 mypassword 可以根据规划进行修改
mysql> FLUSH PRIVILEGES;
    
```

**步骤 2** 创建 mail 库中的虚拟域名表、用户信息表和邮件转发表。

```

mysql> use mail;
mysql> CREATE TABLE `domains` (`id` int(11) NOT NULL auto_increment,`name` varchar(50) NOT
NULL,PRIMARY KEY (`id`)) ;
mysql> CREATE TABLE `users` (`id` int(11) NOT NULL auto_increment,`domain_id` int(11) NOT
NULL,`password` varchar(106) NOT NULL,`email` varchar(100) NOT NULL,PRIMARY KEY (`id`),UNIQUE
KEY `email` (`email`),FOREIGN KEY (domain_id) REFERENCES domains(id) ON DELETE CASCADE);
    
```

```
mysql> CREATE TABLE `aliases` (`id` INT NOT NULL AUTO_INCREMENT,`domain_id` INT NOT NULL,`source` varchar(100) NOT NULL,`destination` varchar(100) NOT NULL,PRIMARY KEY (`id`),FOREIGN KEY (domain_id) REFERENCES domains(id) ON DELETE CASCADE);
```

步骤 3 在虚拟域名表中插入域名数据。

```
mysql> INSERT INTO `mail`.`domains` (`id`,`name`) VALUES ('1','huawei.com'),('2','mail.huawei.com');
```

步骤 4 在用户信息表中插入用户数据。

```
mysql> INSERT INTO `mail`.`users`
-> (`id`,`domain_id`,`password`,`email`)
-> VALUES
-> ('1','1', ENCRYPT('123456'),'test1@huawei.com'),
-> ('2','1', ENCRYPT('123456'),'test2@huawei.com');
```

步骤 5 设置别名。

```
mysql> INSERT INTO `mail`.`aliases` (`id`,`domain_id`,`source`,`destination`) VALUES ('1','1','alias@huawei.com','test1@huawei.com'),('2','1','alias@huawei.com','test2@huawei.com');
```

步骤 6 检查数据是否正常插入，若每个查询语句都能正常查询到两条记录，则表示正常。

```
mysql> SELECT * FROM mail.domains;
mysql> SELECT * FROM mail.users;
mysql> SELECT * FROM mail.aliases;
mysql> quit
Bye
```

## 4.3.4 安装及配置 postfix 邮局

步骤 1 安装 postfix、dovecot。

```
[root@mail ~]# dnf -y install postfix dovecot dovecot-mysql
[root@mail ~]# postconf -d | grep mail_version
mail_version = 3.3.1
milter_macro_v = $mail_name $mail_version
```

步骤 2 修改 postfix 主配置文件。

```
[root@mail ~]# cp /etc/postfix/main.cf /etc/postfix/main.cf.org #备份原文件
[root@mail ~]# vim /etc/postfix/main.cf
#可以通过查找到对应内容行进行修改，若没有此行，则新增
myhostname = mail.huawei.com #94 行左右，修改
mydomain = huawei.com #102 行左右，修改
mydestination = $myhostname, localhost.$mydomain, localhost #修改
mynetworks = 192.168.xx.0/24 #283 行左右，修改，这里换成规划的子网网段
inet_interfaces = all #135 行左右，修改
message_size_limit = 30720000 #新增
```

```

relayhost = #332 行左右, 修改
alias_maps = hash:/etc/aliases #405 行左右, 修改

smtpd_tls_key_file = /etc/pki/dovecot/private/dovecot.pem #715 行左右, 修改
smtpd_tls_cert_file = /etc/pki/dovecot/certs/dovecot.pem #709 行左右, 修改
smtpd_use_tls=yes #新增
smtpd_tls_auth_only = yes #新增
smtpd_sasl_type = dovecot #新增
smtpd_sasl_path = private/auth #新增
smtpd_sasl_auth_enable = yes #新增
smtpd_recipient_restrictions =permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination
#新增

virtual_transport = dovecot #新增
virtual_mailbox_domains = mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf #新增
virtual_mailbox_maps = mysql:/etc/postfix/mysql-virtual-mailbox-maps.cf #新增
virtual_alias_maps = mysql:/etc/postfix/mysql-virtual-alias-maps.cf,mysql:/etc/postfix/mysql-virtual-
email2email.cf #新增

```

### 步骤 3 创建虚拟域名配置文件。

```

[root@mail ~]# vim /etc/postfix/mysql-virtual-mailbox-domains.cf
user = mail_admin
password = mypassword #创建邮件数据库管理员 mail_admin 时配置的密码
hosts = 127.0.0.1
dbname = mail
query = SELECT 1 FROM domains WHERE name='%s'

```

### 步骤 4 创建虚拟邮箱配置文件。

```

[root@mail ~]# vim /etc/postfix/mysql-virtual-mailbox-maps.cf
user = mail_admin
password = mypassword #创建邮件数据库管理员 mail_admin 时配置的密码
hosts = 127.0.0.1
dbname = mail
query = SELECT 1 FROM users WHERE email='%s'

```

### 步骤 5 创建电子邮件与文件映射。

```

[root@mail ~]# vim /etc/postfix/mysql-virtual-alias-maps.cf
user = mail_admin
password = mypassword #创建邮件数据库管理员 mail_admin 时配置的密码
hosts = 127.0.0.1
dbname = mail
query = SELECT destination FROM aliases WHERE source='%s'
[root@mail ~]# vim /etc/postfix/mysql-virtual-email2email.cf
user = mail_admin

```

```
password = mypassword          #创建邮件数据库管理员 mail_admin 时配置的密码
hosts = 127.0.0.1
dbname = mail
query = SELECT email FROM users WHERE email='%s'
```

步骤 6 启动并测试 postfix。

```
[root@mail ~]# systemctl restart postfix.service
[root@mail ~]# systemctl enable postfix.service
```

步骤 7 测试数据库连接配置。

```
[root@mail ~]# postmap -q huawei.com mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf
1
[root@mail ~]# postmap -q test1@huawei.com mysql:/etc/postfix/mysql-virtual-mailbox-maps.cf
1
[root@mail ~]# postmap -q alias@huawei.com mysql:/etc/postfix/mysql-virtual-alias-maps.cf
test1@huawei.com,test2@huawei.com
```

步骤 8 更改 master.cf 配置文件。

```
[root@mail ~]# vim /etc/postfix/master.cf          #可以清空原文，然后粘贴如下内容
#
# Postfix master process configuration file.  For details on the format
# of the file, see the master(5) manual page (command: "man 5 master").
#
# Do not forget to execute "postfix reload" after editing this file.
#
# =====
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# =====

smtp      inet  n       -       n       -       -       smtpd
#smtp     inet  n       -       n       -       1       postscreen
#smtpd    pass  -       -       n       -       -       smtpd
#dnsblog  unix  -       -       n       -       0       dnsblog
#tlsproxy unix  -       -       n       -       0       tlsproxy
submission inet n       -       n       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_reject_unlisted_recipient=no
  -o smtpd_client_restrictions=$mua_client_restrictions
  -o smtpd_helo_restrictions=$mua_helo_restrictions
  -o smtpd_sender_restrictions=$mua_sender_restrictions
  -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
smtps     inet  n       -       n       -       -       smtpd
```

```

-o syslog_name=postfix/smtps
-o smtpd_tls_wrappermode=yes
-o smtpd_sasl_auth_enable=yes
-o smtpd_reject_unlisted_recipient=no
-o smtpd_client_restrictions=$mua_client_restrictions
-o smtpd_helo_restrictions=$mua_helo_restrictions
-o smtpd_sender_restrictions=$mua_sender_restrictions
-o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
-o milter_macro_daemon_name=ORIGINATING
#628      inet  n      -      n      -      -      qmcpd
pickup   unix  n      -      n      60     1      pickup
cleanup  unix  n      -      n      -      0      cleanup
qmgr     unix  n      -      n      300    1      qmgr
#qmgr    unix  n      -      n      300    1      oqmgr
tlsmgr   unix  -      -      n      1000?  1      tlsmgr
rewrite  unix  -      -      n      -      -      trivial-rewrite
bounce   unix  -      -      n      -      0      bounce
defer    unix  -      -      n      -      0      bounce
trace    unix  -      -      n      -      0      bounce
verify   unix  -      -      n      -      1      verify
flush    unix  n      -      n      1000?  0      flush
proxymap unix  -      -      n      -      -      proxymap
proxywrite unix -      -      n      -      1      proxymap
smtp     unix  -      -      n      -      -      smtp
relay    unix  -      -      n      -      -      smtp
#        -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq    unix  n      -      n      -      -      showq
error    unix  -      -      n      -      -      error
retry    unix  -      -      n      -      -      error
discard  unix  -      -      n      -      -      discard
local    unix  -      n      n      -      -      local
virtual  unix  -      n      n      -      -      virtual
lmtpl    unix  -      -      n      -      -      lmtpl
anvil    unix  -      -      n      -      1      anvil
scache   unix  -      -      n      -      1      scache
#
# =====
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# Many of the following services use the Postfix pipe(8) delivery
# agent. See the pipe(8) man page for information about ${recipient}
# and other message envelope options.
# =====
#
# maildrop. See the Postfix MAILDROP_README file for details.
# Also specify in main.cf: maildrop_destination_recipient_limit=1
#

```

```
#maildrop unix - n n - - pipe
# flags=DRhu user=vmail argv=/usr/local/bin/maildrop -d ${recipient}
#
# =====
#
# Recent Cyrus versions can use the existing "lmtp" master.cf entry.
#
# Specify in cyrus.conf:
# lmtp cmd="lmtpd -a" listen="localhost:lmtp" proto=tcp4
#
# Specify in main.cf one or more of the following:
# mailbox_transport = lmtp:inet:localhost
# virtual_transport = lmtp:inet:localhost
#
# =====
#
# Cyrus 2.1.5 (Amos Gouaux)
# Also specify in main.cf: cyrus_destination_recipient_limit=1
#
#cyrus unix - n n - - pipe
# user=cyrus argv=/usr/lib/cyrus-imapd/deliver -e -r ${sender} -m ${extension} ${user}
#
# =====
#
# Old example of delivery via Cyrus.
#
#old-cyrus unix - n n - - pipe
# flags=R user=cyrus argv=/usr/lib/cyrus-imapd/deliver -e -m ${extension} ${user}
#
# =====
#
# See the Postfix UUCP_README file for configuration details.
#
#uucp unix - n n - - pipe
# flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
#
# =====
#
# Other external delivery methods.
#
#ifmail unix - n n - - pipe
# flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
#
#bsmtp unix - n n - - pipe
# flags=Fq. user=bsmtp argv=/usr/local/sbin/bsmtp -f $sender $nexthop $recipient
#
#scalemail-backend unix - n n - 2 pipe
# flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store
```

```
# ${nexthop} ${user} ${extension}
#
#mailman unix - n n - - pipe
# flags=FR user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py
# ${nexthop} ${user}
dovecot unix - n n - - pipe
 flags=DRhu user=vmail:vmail argv=/usr/libexec/dovecot/deliver -f ${sender} -d ${recipient}
```

步骤 9 重启 postfix。

```
[root@mail ~]# systemctl restart postfix
```

步骤 10 测试邮箱服务。

```
[root@mail ~]# dnf -y install telnet
[root@mail ~]# telnet mail.huawei.com 25
#粗体字代表需要输入的内容
Trying 192.168.XX.XX...
Connected to mail.huawei.com.
Escape character is '^]'.
220 mail.huawei.com ESMTP Postfix
hello mail.huawei.com
250 mail.huawei.com
mail from:test1@huawei.com
250 2.1.0 Ok
rcpt to:test1@huawei.com
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
this is a test mail
.
250 2.0.0 Ok: queued as D7E56400107
quit
221 2.0.0 Bye
Connection closed by foreign host.
[root@mail ~]# mailq
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-----
D7E56400107      337 Mon xxx xx xx:xx:xx  test1@huawei.com
                                     (unknown mail transport error)
                                     test1@huawei.com

-- 0 Kbytes in 1 Request.
```

### 4.3.5 配置 dovecot

步骤 1 备份配置文件。

```
[root@mail ~]# cp /etc/dovecot/dovecot.conf /etc/dovecot/dovecot.conf.org
```

```
[root@mail ~]# cp /etc/dovecot/conf.d/10-mail.conf /etc/dovecot/conf.d/10-mail.conf.org
[root@mail ~]# cp /etc/dovecot/conf.d/10-auth.conf /etc/dovecot/conf.d/10-auth.conf.org
[root@mail ~]# cp /etc/dovecot/conf.d/10-master.conf /etc/dovecot/conf.d/10-master.conf.org
[root@mail ~]# cp /etc/dovecot/conf.d/10-ssl.conf /etc/dovecot/conf.d/10-ssl.conf.org
```

## 步骤 2 配置 dovecot 配置文件。

```
[root@mail ~]# vim /etc/dovecot/dovecot.conf
#按照如下要求，修改此配置文件
#第 24 行，取消注释
protocols = imap pop3 lmtp
#第 30 行，取消注释
listen = *, ::

[root@mail ~]# vim /etc/dovecot/conf.d/10-mail.conf #找到并修改如下内容
mail_location = maildir:/home/vmail/%d/%n/Maildir
mail_privileged_group = mail
```

## 步骤 3 创建 vmail 用户和组。

```
[root@mail ~]# groupadd -g 5000 vmail
[root@mail ~]# useradd -g vmail -u 5000 vmail
```

## 步骤 4 配置 10-auth.conf 文件。

```
[root@mail ~]# vim /etc/dovecot/conf.d/10-auth.conf #找到并修改如下内容
disable_plaintext_auth = no
auth_mechanisms = plain login
#!include auth-system.conf.ext #122 行左右，前面加上#号
!include auth-sql.conf.ext #123 行左右，前面删除#号
```

## 步骤 5 配置 10-ssl.conf 文件。

```
[root@mail ~]# vim /etc/dovecot/conf.d/10-ssl.conf #将 ssl 值改为 no，前面若有#号请删除
ssl = no
```

## 步骤 6 编辑连接数据库的文件。

```
[root@mail ~]# vim /etc/dovecot/dovecot-sql.conf.ext
driver = mysql
connect = host=127.0.0.1 dbname=mail user=mail_admin password=mypassword #创建邮件数据库管
理员 mail_admin 时配置的密码
default_pass_scheme = CRYPT
password_query = SELECT email as user, password FROM users WHERE email='%u';
user_query = SELECT ('5000') as 'uid','(5000)' as 'gid'
```

## 步骤 7 配置目录权限。

```
[root@mail ~]# chown -R vmail:dovecot /etc/dovecot
[root@mail ~]# chmod -R o-rwx /etc/dovecot
```

## 步骤 8 配置 10-master.conf 文件。

```
[root@mail ~]# vim /etc/dovecot/conf.d/10-master.conf
#建议清空配置，然后填入如下内容，有变更地方已加粗
#default_process_limit = 100
#default_client_limit = 1000

# Default VSZ (virtual memory size) limit for service processes. This is mainly
# intended to catch and kill processes that leak memory before they eat up
# everything.
#default_vsz_limit = 256M

# Login user is internally used by login processes. This is the most untrusted
# user in Dovecot system. It shouldn't have access to anything at all.
#default_login_user = dovenull

# Internal user is used by unprivileged processes. It should be separate from
# login user, so that login processes can't disturb other processes.
#default_internal_user = dovecot

service imap-login {
  inet_listener imap {
    port = 143
  }
  inet_listener imaps {
    port = 993
    ssl = no
  }
}

# Number of connections to handle before starting a new process. Typically
# the only useful values are 0 (unlimited) or 1. 1 is more secure, but 0
# is faster. <doc/wiki/LoginProcess.txt>
#service_count = 1

# Number of processes to always keep waiting for more connections.
#process_min_avail = 0

# If you set service_count=0, you probably need to grow this.
#vsz_limit = $default_vsz_limit
}

service pop3-login {
  inet_listener pop3 {
    port = 110
  }
  inet_listener pop3s {
    port = 995
    ssl = no
  }
}
```

```
}
}

service lmtp {
    unix_listener /var/spool/postfix/private/dovecot-lmtp {
        mode = 0600
        user = postfix
        group = postfix
    }

    # Create inet listener only if you can't use the above UNIX socket
    #inet_listener lmtp {
        # Avoid making LMTP visible for the entire internet
        #address =
        #port =
    #}
}

service imap {
    # Most of the memory goes to mmap()ing files. You may need to increase this
    # limit if you have huge mailboxes.
    #vsz_limit = $default_vsz_limit

    # Max. number of IMAP processes (connections)
    #process_limit = 1024
}

service pop3 {
    # Max. number of POP3 processes (connections)
    #process_limit = 1024
}

service auth {
    # auth_socket_path points to this userdb socket by default. It's typically
    # used by dovecot-lda, doveadm, possibly imap process, etc. Users that have
    # full permissions to this socket are able to get a list of all usernames and
    # get the results of everyone's userdb lookups.
    #
    # The default 0666 mode allows anyone to connect to the socket, but the
    # userdb lookups will succeed only if the userdb returns an "uid" field that
    # matches the caller process's UID. Also if caller's uid or gid matches the
    # socket's uid or gid the lookup succeeds. Anything else causes a failure.
    #
    # To give the caller full permissions to lookup all users, set the mode to
    # something else than 0666 and Dovecot lets the kernel enforce the
    # permissions (e.g. 0777 allows everyone full permissions).
    unix_listener auth-userdb {
        mode = 0666
    }
}
```

```
user = vmail
#group =
}

# Postfix smtp-auth
unix_listener /var/spool/postfix/private/auth {
    mode = 0666
    user = postfix
    user = postfix
}

# Auth process is run as this user.
#user = $default_internal_user
user= dovecot
}

service auth-worker {
    # Auth worker process is run as root by default, so that it can access
    # /etc/shadow. If this isn't necessary, the user should be changed to
    # $default_internal_user.
    #user = root
    user = vmail
}

service dict {
    # If dict proxy is used, mail processes should have access to its socket.
    # For example: mode=0660, group=vmail and global mail_access_groups=vmail
    unix_listener dict {
        #mode = 0600
        #user =
        #group =
    }
}
```

步骤 9 启动 dovecot。

```
[root@mail ~]# systemctl restart dovecot.service
[root@mail ~]# systemctl enable dovecot.service
```

### 4.3.6 使用邮件客户端测试

步骤 1 下载邮件客户端软件 Thunderbird，并安装在 PC 中。

步骤 2 配置电脑 hosts 文件，添加域名记录到 hosts 文件。（文件位置在 C:\Windows\System32\drivers\etc 目录，建议修改网卡 DNS 配置）

```
192.168.xxx.xxx mail.huawei.com    #这里 ip 地址填写成 openEuler 操作系统 IP，下同
192.168.xxx.xxx huawei.com
```

步骤 3 打开 Thunderbird 客户端，填写 test1 用户信息，然后点击“手动配置”。（密码是 123456）

### 设置现有的电子邮件地址

若要使用您现有的电子邮件地址，请填写您的凭据。  
Thunderbird 将自动搜索可用的和推荐的服务器配置。

您的全名

电子邮件地址

密码

  
 记住密码

您的登录凭据只会存储在您的计算机本地。

步骤 4 按照如下图填写配置信息，配置完成后点击“重新测试”。

电子邮件地址  
test1@huawei.com

密码  
•••••

记住密码

**服务器设置**

**收件服务器**

协议: POP3

主机名: mail.huawei.com

端口: 110

连接安全性: 无

验证方式: 自动检测

用户名: test1@huawei.com

**发件服务器**

主机名: mail.huawei.com

端口: 25

连接安全性: 无

验证方式: 自动检测

用户名: test1@huawei.com

[高级配置](#)

Thunderbird 将尝试自动检测留白的字段。

步骤 5 等待提示“探测制定服务器找到下列配置”，然后点击右下角“完成”。

电子邮件地址  
test1@huawei.com

密码  
●●●●●●

记住密码

探测指定服务器找到下列配置:

### 服务器设置

**收件服务器**

协议: POP3

主机名: mail.huawei.com

端口: 110

连接安全性: 无

验证方式: 普通密码

用户名: test1@huawei.com

**发件服务器**

主机名: mail.huawei.com

端口: 25

连接安全性: 无

验证方式: 普通密码

用户名: test1@huawei.com

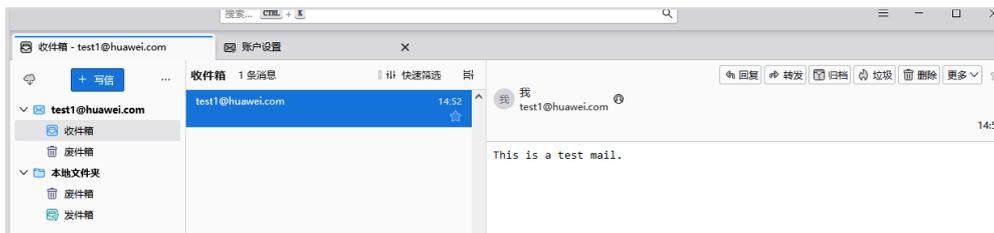
[高级配置](#)

Thunderbird 将尝试自动检测留白的字段。

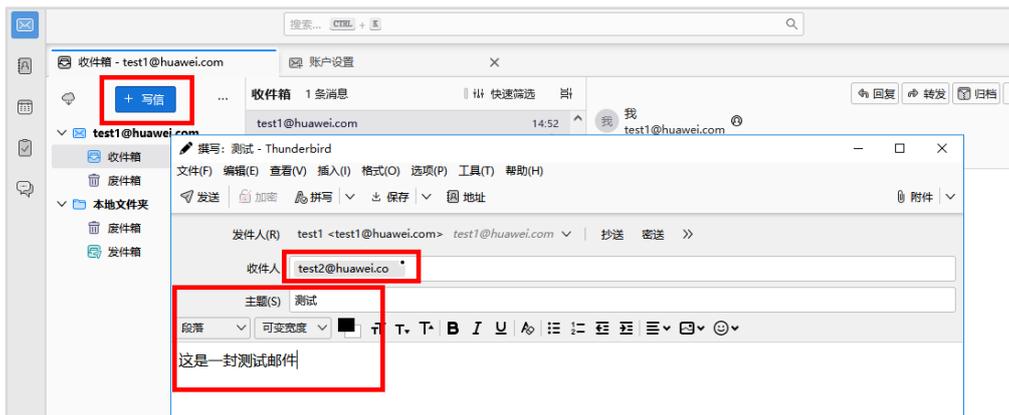
步骤 6 因为未配置 SSL 加密，此时会有告警弹窗，勾选“我已了解相关风险”，然后点击“确认”，即可。



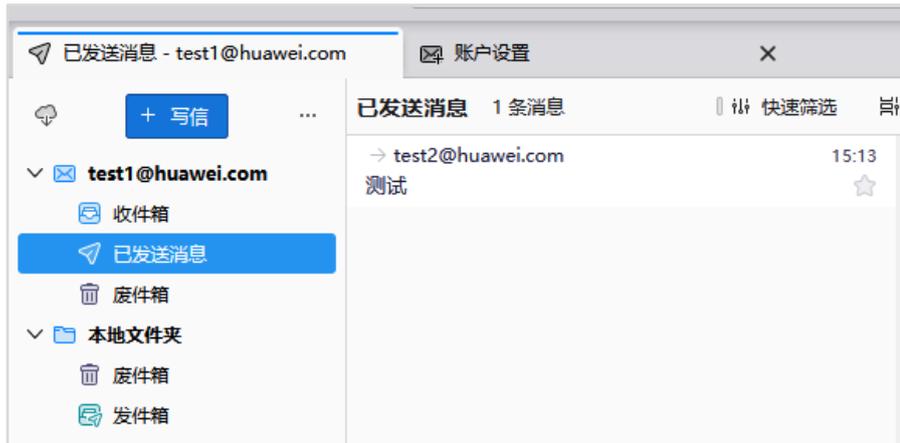
步骤 7 此时即可看到在安装 postfix 步骤时发送的测试邮件。（初次配置，需要等待较久）



步骤 8 也可以测试给 test2@huawei.com 发送邮件，完成后点击“发送”。



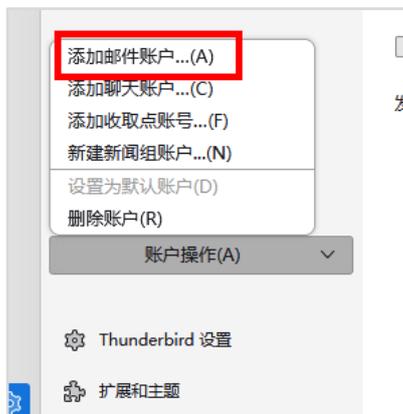
步骤 9 在发件箱可以看到已发送的邮件。



步骤 10 点击邮件客户端左下角的齿轮图标，选中“账户设置”。



步骤 11 在“账户操作(A)”中选择“添加邮件账户...(A)”。



步骤 12 参照前面的步骤，使用 test2@huawei.com 登录邮箱，密码也是 123456。

您的全名

电子邮件地址

密码

  
 记住密码**服务器设置****收件服务器**

协议: POP3

主机名: mail.huawei.com

端口: 110

连接安全性: 无

验证方式: 自动检测

用户名: test2@huawei.com

**发件服务器**

主机名: mail.huawei.com

端口: 25

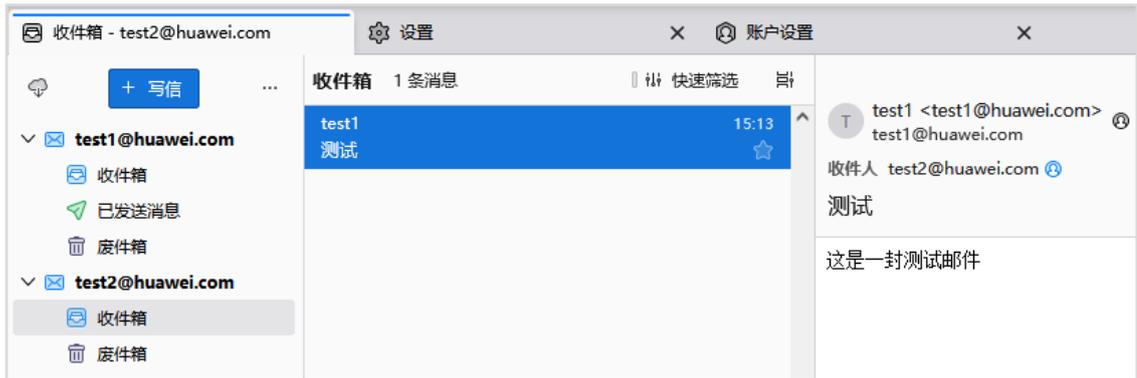
连接安全性: 无

验证方式: 自动检测

用户名: test2@huawei.com

[高级配置](#)

步骤 13 登录成功后即可在收件箱看到刚刚 test1 发给 test2 的邮件。



## 4.4 思考题

本实验中用的是 openEuler 操作系统 Everything 镜像，无法同步更新；在实际部署时，可以通过哪些方式及时同步开源镜像站中的软件包？

答案：可以用一台服务器连接外网，使用 RClone 同步外部镜像站资源。

# 5 (选做) Atlas 800(3000)推理场景 AI 软件部署

## 5.1 实验介绍

### 5.1.1 关于本实验

本实验介绍在物理机中安装 NPU 驱动和 AI 软件的方法，通过本次上机实验操作，你可以熟悉怎么使用 SmartKit 工具安装 NPU 驱动和 AI 软件（CANN 和 ToolBox）。

**\*注意：**本实验为选做实验，如实验环境使用 Taishan 200 系列服务器，则无法进行本实验操作，可通过阅读本实验手册具体步骤了解安装 NPU 驱动的具体方法。

### 5.1.2 实验目的

- 掌握使用 SmartKit 工具安装 NPU 驱动的方法；
- 掌握使用 SmartKit 工具；
- 了解 AI 算力的使用。

### 5.1.3 实验所需软件

表5-1 实验所需软件

软件	下载地址
SmartKit_23.0.0.6_zh.zip	<a href="https://support.huawei.com/enterprise/zh/distributed-storage/smartkit-pid-8576706/software/261548176?idAbsPath=fixnode01%7C7919749%7C251366268%7C250389224%7C251366263%7C8576706">https://support.huawei.com/enterprise/zh/distributed-storage/smartkit-pid-8576706/software/261548176?idAbsPath=fixnode01%7C7919749%7C251366268%7C250389224%7C251366263%7C8576706</a>
SmartKit_func_server_exp.tar	从 HClA-Computing 认证官网下载
Ascend_Package-V3.0.zip	从 HClA-Computing 认证官网下载
sample-master.zip	<a href="https://gitee.com/ascend/samples">https://gitee.com/ascend/samples</a>
resnet50.prototxt	<a href="https://obs-9be7.obs.cn-east-2.myhuaweicloud.com/003_Atch_Models/AE/ATC%20Model/resnet50/resnet50.prototxt">https://obs-9be7.obs.cn-east-2.myhuaweicloud.com/003_Atch_Models/AE/ATC%20Model/resnet50/resnet50.prototxt</a>

Ascend-hdk-310p-npu-driver-23.0.1-1.aarch64.rpm Ascend-hdk-310p-npu-firmware-7.1.0.4.220-1.noarch.rpm	<a href="https://support.huawei.com/enterprise/zh/ascend-computing/ascend-hdk-pid-252764743/software/261964445?idAbsPath=fixnode01%7C23710424%7C251366513%7C22892968%7C252764743">https://support.huawei.com/enterprise/zh/ascend-computing/ascend-hdk-pid-252764743/software/261964445?idAbsPath=fixnode01%7C23710424%7C251366513%7C22892968%7C252764743</a>
resnet50.caffemodel	<a href="https://obs-9be7.obs.cn-east-2.myhuaweicloud.com/003_Atch_Models/AE/ATC%20Model/resnet50/resnet50.caffemodel">https://obs-9be7.obs.cn-east-2.myhuaweicloud.com/003_Atch_Models/AE/ATC%20Model/resnet50/resnet50.caffemodel</a>

## 5.2 实验任务配置

### 5.2.1 安装 NPU 驱动

步骤 1 使用 putty 登录 openEuler 操作系统，执行如下命令安装依赖。

```
[root@mail ~]# dnf install -y gcc gcc-c++ make cmake unzip zlib-devel libffi-devel openssl-devel pciutils net-tools sqlite-devel lapack-devel gcc-gfortran python3-devel dkms kernel-devel-$(uname -r)
```

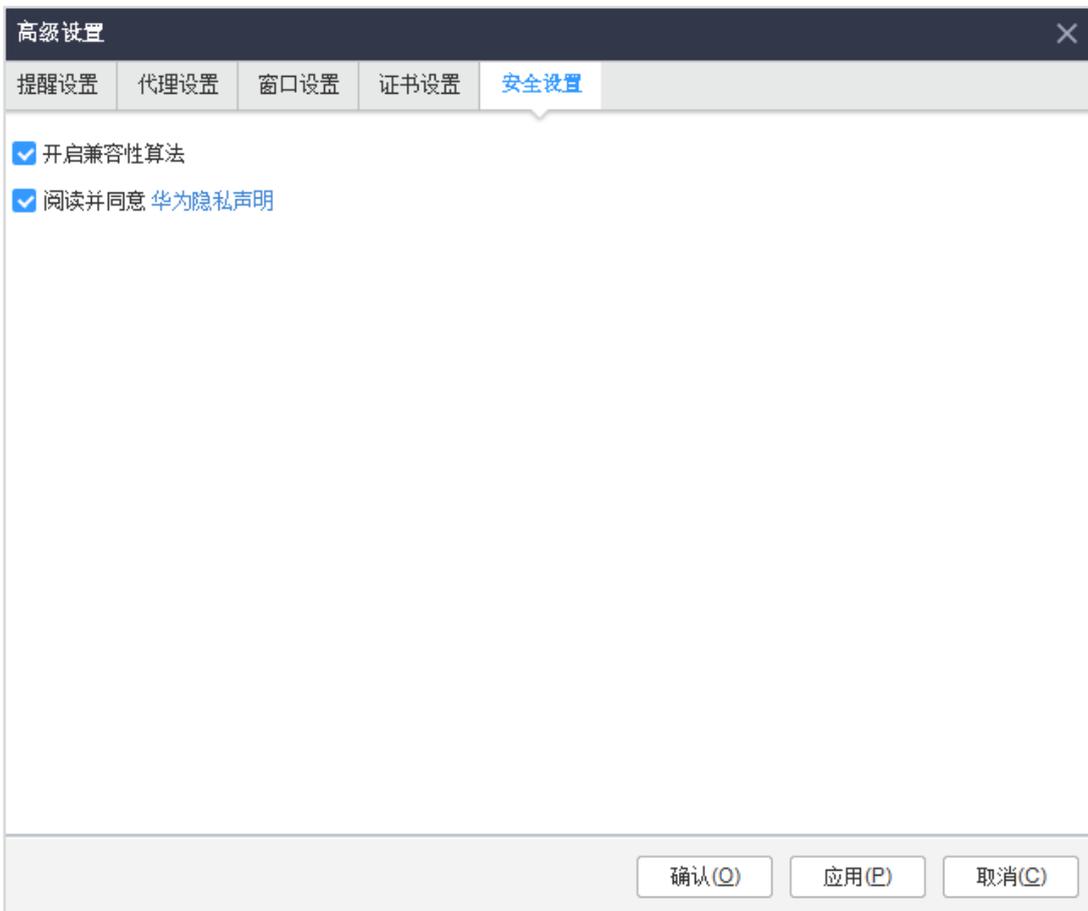
步骤 2 在 PC 内解压并安装 SmartKit 工具。打开此工具，如出现认证界面，请直接关闭该界面。在 SmartKit 首页，会出现操作安全注意事项，单击“确认”继续，进入 SmartKit 首页。



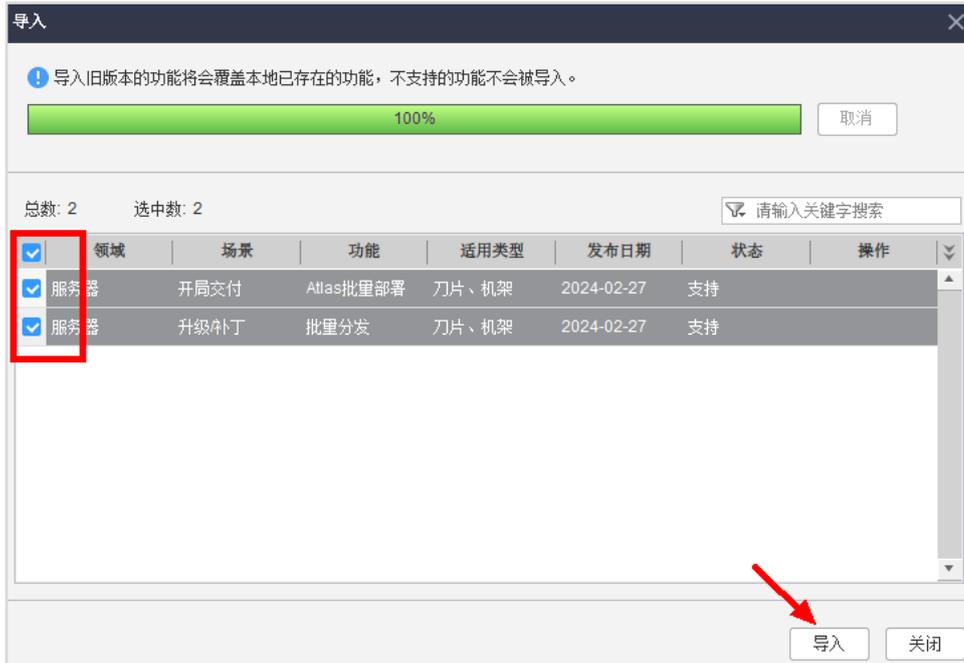
步骤 3 点击右上角的齿轮图标，进入设置页面。



步骤 4 点击右下角的“高级设置”->“安全设置”，勾选“阅读并同意华为隐私声明”。然后点击“确认”。



步骤 5 回到软件首页，点击“服务器”->“升级/补丁”模块里的“批量分发”，此时会弹出缺失工具模块的提醒，点击“导入”，导入 SmartKit\_func\_server\_exp.tar 工具包。若提示无校验，请忽略，直接点击“确定”，再勾选两个工具，点击“导入”，直到导入成功。



步骤 6 关闭导入界面，再次点击“批量分发”，



步骤 7 点击“选择设备”->“OS”->“增加”。



步骤 8 参照如下图示填写相关信息。（起始 IP 和结束 IP 填写 openEuler 操作系统 IP，密码为操作系统的 root 密码，其他保持默认）

**增加设备**

\* 起始IP:  \* 结束IP:

\* 用户名:  \* 密码:

SNMPV3加密码:  (如果不填则使用上面的用户密码, 如果SNMPV3加密码和用户密码不一致, 则SNMPV3加密码)

高级设置

OS/IES选项

root 密码:  (对于OS类型设备, 非root用户如无root权限必须输入root密码)

交换模块选项 (E9000 健康检查、日志收集、升级涉及到交换模块的场景需要)

计算节点信息 (E9000 MM920/MM921或E9000H SMM 涉及到计算节点的场景需要)

确定 取消

步骤 9 点击“确定”后, 等待增加设备成功。

**增加设备**

执行中... 请稍候。

10%

总数: 1 选中数: 0

对象	状态	进度	处理建议
192.168.10.101	执行中	10 %	

取消

步骤 10 勾选已添加的设备, 然后点击“确定”。

**选择设备**

iBMC HMM OS 1 IES E9000H(TCE)

总数: 1 选中数: 1

请输入关键字搜索

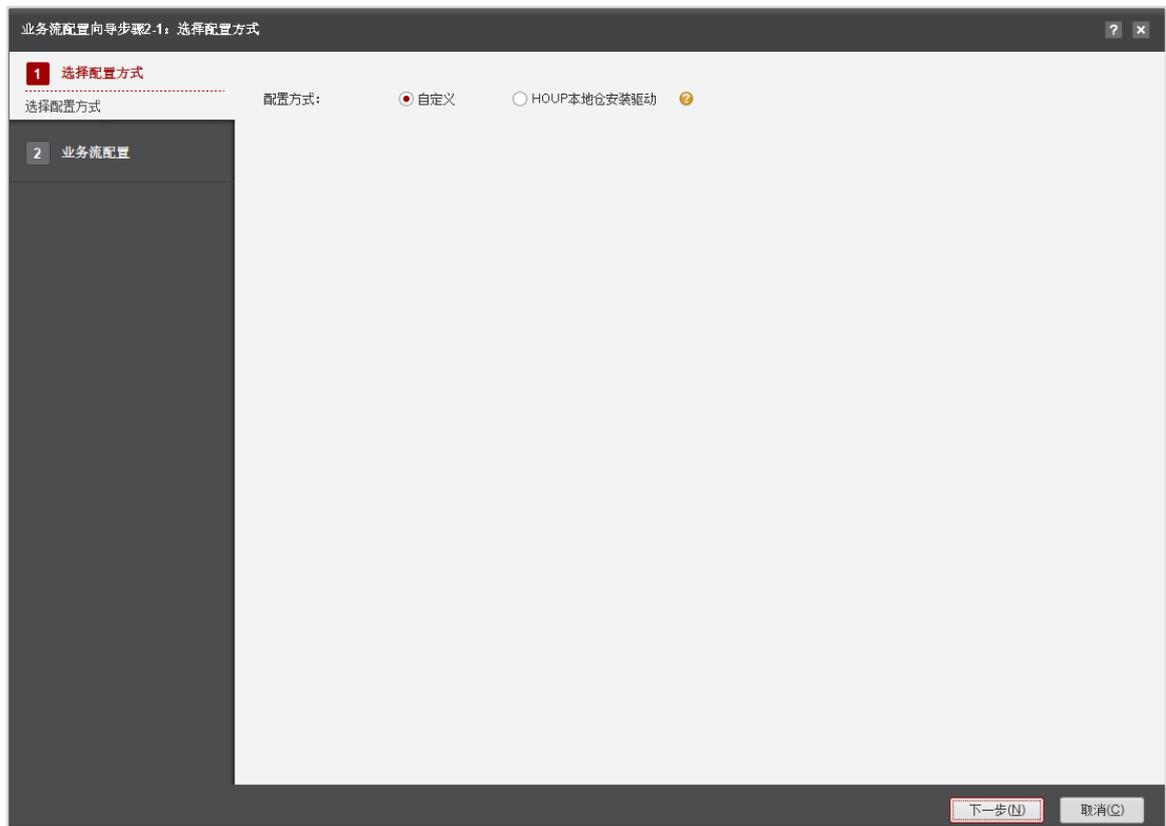
IP	类型	用户名	kernelVersion
<input checked="" type="checkbox"/> 192.168.10.101	openEuler 22.03 LTS	root	5.10.0-60.18.0.50.oe2203.aarch64

确定(D) 取消(O)

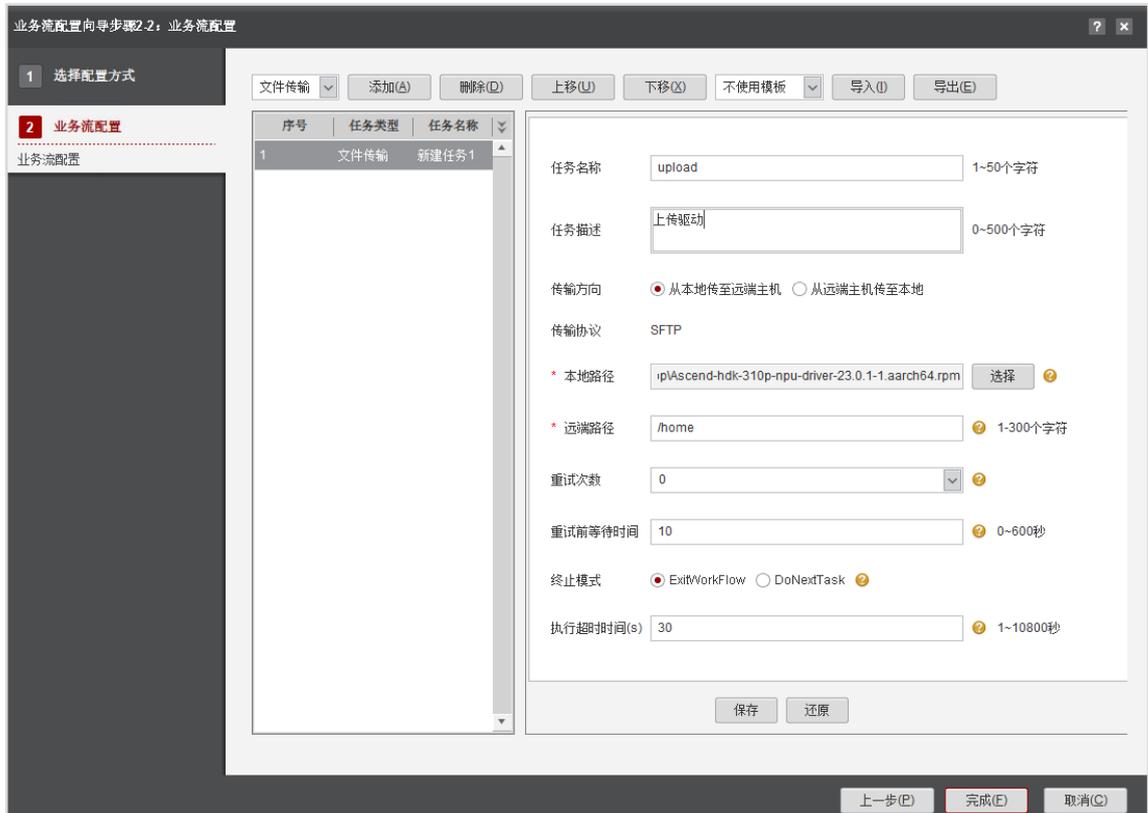
步骤 11 点击“批量分发”->勾选步骤 10 中添加的设备->“配置业务流”，进入业务流配置向导式界面。



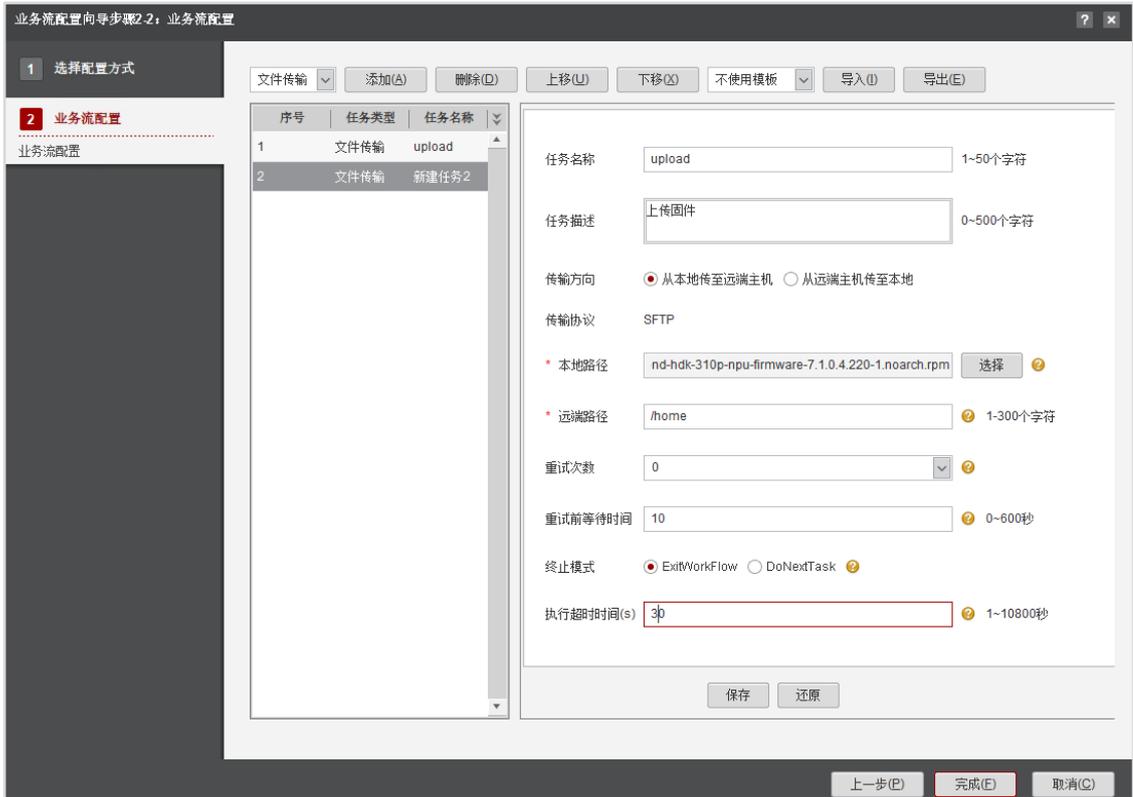
步骤 12 保持默认选项，点击“下一步”，进入业务流配置界面。



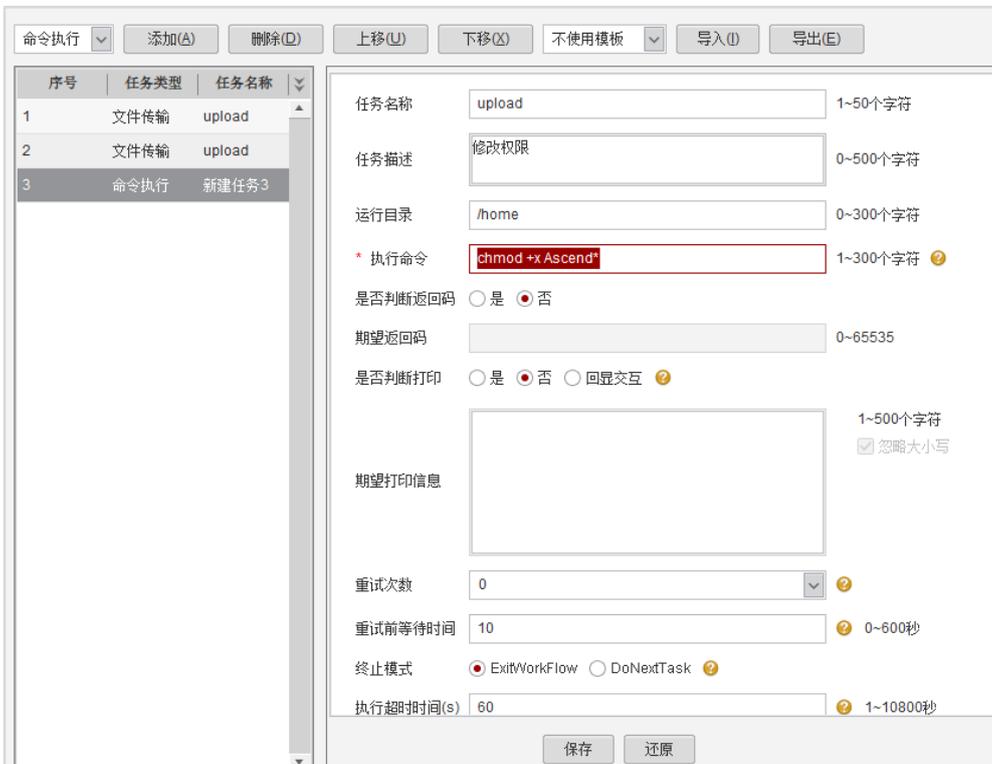
- 步骤 13 选择“文件传输”，点击“添加”。输入任务名称和任务描述，传输方向保持默认选择（从本地传至远端主机），点击本地路径“选择”，选择本地下载的 Ascend-hdk-310p-npu-driver-23.0.1-1.aarch64.rpm 包，远端路径输入“/home”，将 NPU 驱动包上传到“/home”目录下。执行超时时间可以修改成“30”，点击“保存”完成文件传输配置。



- 步骤 14 参照上述步骤，创建上传固件 Ascend-hdk-310p-npu-firmware-7.1.0.4.220-1.noarch.rpm 的任务，同样上传到“/home”目录。



步骤 15 选择“命令执行”，点击“添加”。输入任务名称和任务描述，运行目录输入“/home”，执行命令输入“chmod +x Ascend\*”，点击“保存”，完成命令执行配置。



**步骤 16** 执行安装命令。选择“命令执行”，点击“添加”。输入任务名称和任务描述，运行目录输入“/home”，执行命令输入“rpm -ivh Ascend-hdk\*.rpm”，点击“保存”，点击“确认”保存成功，完成命令执行配置。点击最下方的“完成”，完成业务流配置，业务流配置界面会自动关闭，并返回批量分发页面。

序号	任务类型	任务名称
1	文件传输	upload
2	文件传输	upload
3	命令执行	upload
4	命令执行	新建任务4

任务名称: install (1~50个字符)

任务描述: 安装驱动 (0~500个字符)

运行目录: /home (0~300个字符)

\* 执行命令: rpm -ivh Ascend-hdk\*.rpm (1~300个字符)

是否判断返回码:  是  否

期望返回码: (0~65535)

是否判断打印:  是  否  回显交互

期望打印信息: (1~500个字符, 忽略大小写)

重试次数: 0

重试前等待时间: 10 (0~600秒)

终止模式:  ExitWorkFlow  DoNextTask

执行超时时间(s): 60 (1~10800秒)

保存 还原

**步骤 17** 点击“执行业务流”，开始执行配置的业务流操作待执行状态变成成功后，则安装成功结束。

批量分发

请点击“配置业务流”按钮自定义业务流程，实现添加新任务，调整任务执行顺序等操作。业务流配置完成后，点击“执行业务流”按钮开始批量分发。

提示：建议先对一台设备执行业务流，成功后再批量执行。

配置业务流(C) 编辑业务流(D) **执行业务流(E)** 停止业务流(S) 导出报告(O) 添加其他设备(O)

总数: 1 选中数: 1

IP	类型	任务进度	消息	执行状态	操作
192.168.10.101	openEuler 22.03 LTS(OS)			已配置	

任务名称 任务类型 任务描述 开始时间 结束时间 状态 执行详情

步骤 18 使用 putty 登录 openEuler 操作系统，输入如下命令验证驱动安装是否成功。若能正常显示 NPU 卡信息，则表示安装成功。

```
[root@mail ~]# npu-smi info
```

npu-smi 23.0.1		Version: 23.0.1					
NPU Chip	Name Device	Health Bus-Id	Power (W) AICore (%)	Temp (C) Memory-Usage (MB)	Hugepages-Usage (page)		
10	310P3 0	OK 0000:01:00.0	NA 0	52 1776 / 21527	0	0	
20	310P3 1	OK 0000:02:00.0	NA 0	52 1775 / 21527	0	0	
40	310P3 2	OK 0000:81:00.0	NA 0	56 1780 / 21527	0	0	
50	310P3 3	OK 0000:82:00.0	NA 0	56 1785 / 21527	0	0	
60	310P3 6	OK 0000:85:00.0	NA 0	62 1783 / 21527	0	0	
70	310P3 4	OK 0000:83:00.0	NA 0	57 1784 / 21527	0	0	
80	310P3 5	OK 0000:84:00.0	NA 0	57 1784 / 21527	0	0	

步骤 19 输入 reboot 命令，重启服务器。

```
[root@mail ~]# reboot
```

步骤 20 返回 SmartKit 工具批量分发执行界面，点击关闭批量分发页面，点击“确定”，返回批量分发功能首页，点击“返回首页”，点击“确定”，返回 SmartKit 工具箱首页，结束 NPU 驱动的安装。





## 5.2.2 安装 AI 软件

- 步骤 1 等待服务器重启后，打开 SmartKit 工具，点击“开局交付”模块里的“Atlas 批量部署”，进入 Atlas 批量部署首页。



步骤 2 点击“选择设备”，勾选已添加的 OS 设备，点击“确定”完成设备选择。



步骤 3 待完成后，点击“Atlas 批量部署”，进入 Atlas 批量部署配置界面。等待进度到 100%，点击“设置部署策略”，进入部署策略设置向导界面。



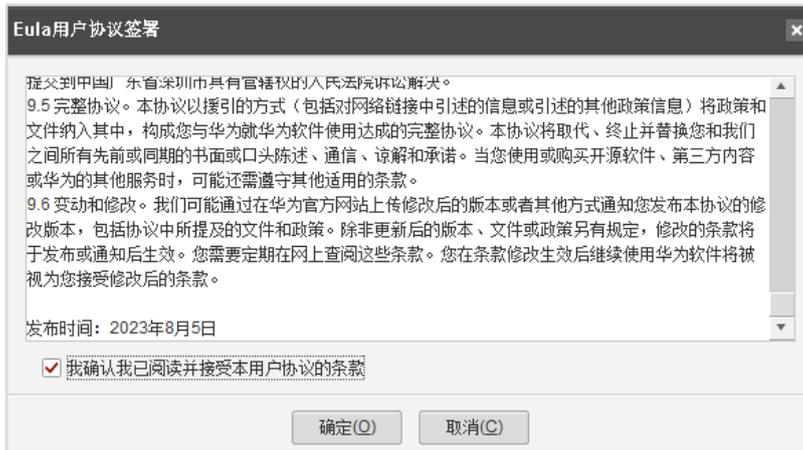
步骤 4 点击本地包路径后的“选择”，在打开的文件选择窗中选择下载的离线安装包 Ascend\_Package-V3.0.zip，点击“确定”读取大包里的安装软件。此过程需要等待三分分钟左右，请耐心等待。



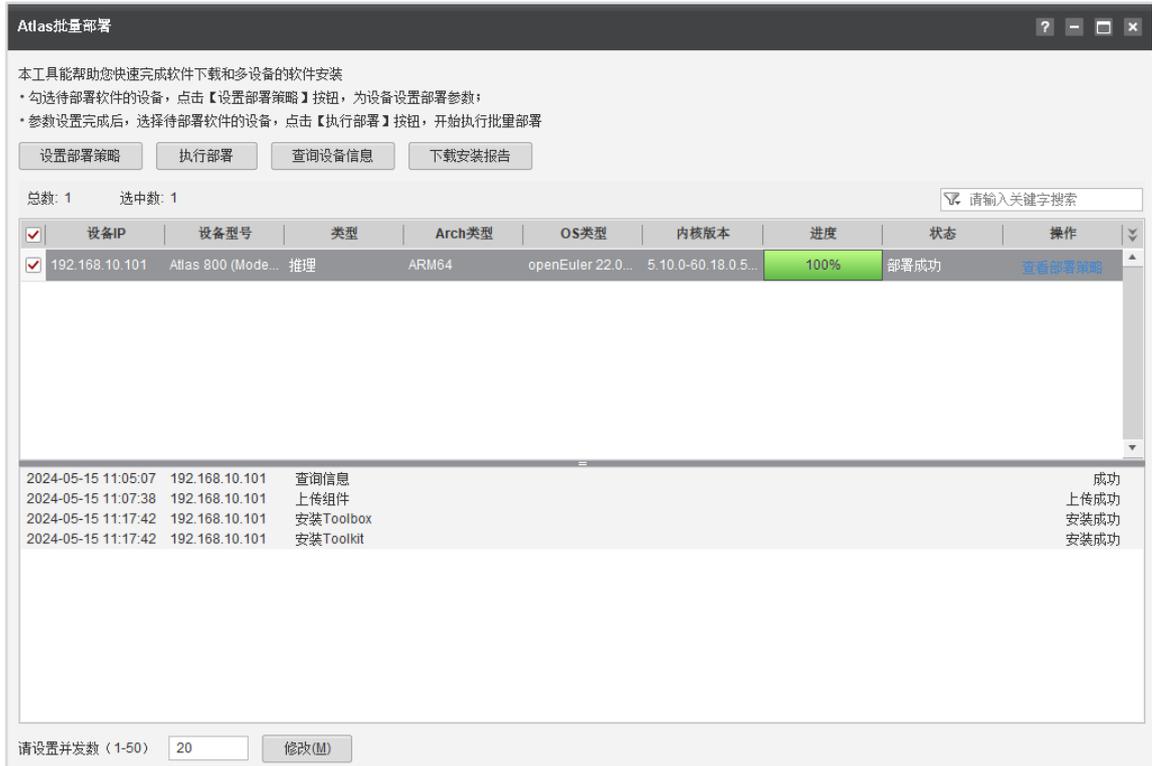
步骤 5 点击“下一步”，进入设置参数界面。勾选“安装完后自动重启设备”，点击“完成”，回到 Atlas 批量部署页面。



步骤 6 点击“执行部署”，用户协议滑到最下方，勾选“我确认我已阅读...”，输入“yes”确认执行危险后，然后点击“确定”，工具开始执行部署。部署时长约 1 小时，请耐心等待。状态变成完成后，即表示完成部署。



步骤 7 等待一段时间后可以看见状态变为“部署成功”，关闭 SmartKit 软件。



**步骤 8 修改 python 默认版本为 3.7.5。**

```
[root@mail ~]# rm -f /usr/bin/python
[root@mail ~]# ln -s /usr/local/python3.7.5/bin/python3 /usr/bin/python
```

**步骤 9 执行 vim /etc/profile，在文件末尾新增如下内容，修改完成后执行 source /etc/profile 更新环境变量。**

```
export LD_LIBRARY_PATH=/usr/local/python3.7.5/lib:$LD_LIBRARY_PATH
export PATH=/usr/local/python3.7.5/bin:$PATH
export
LD_LIBRARY_PATH=/usr/local/Ascend/driver/lib64:/usr/local/Ascend/driver/lib64/common:/usr/local/Ascend/driver/lib64/driver:$LD_LIBRARY_PATH
export ASCEND_TOOLKIT_HOME=/usr/local/Ascend/ascend-toolkit/latest
export
LD_LIBRARY_PATH=${ASCEND_TOOLKIT_HOME}/lib64:${ASCEND_TOOLKIT_HOME}/lib64/plugin/opskernel:${ASCEND_TOOLKIT_HOME}/lib64/plugin/nnengine:${ASCEND_TOOLKIT_HOME}/opp/built-in/op_impl/ai_core/tbe/op_tiling:$LD_LIBRARY_PATH
export PYTHONPATH=${ASCEND_TOOLKIT_HOME}/python/site-packages:${ASCEND_TOOLKIT_HOME}/opp/built-in/op_impl/ai_core/tbe:$PYTHONPATH
export
PATH=${ASCEND_TOOLKIT_HOME}/bin:${ASCEND_TOOLKIT_HOME}/compiler/ccec_compiler/bin:$PATH
export ASCEND_AICPU_PATH=${ASCEND_TOOLKIT_HOME}
export ASCEND_OPP_PATH=${ASCEND_TOOLKIT_HOME}/opp
export TOOLCHAIN_HOME=${ASCEND_TOOLKIT_HOME}/toolkit
export ASCEND_HOME_PATH=${ASCEND_TOOLKIT_HOME}
export PATH=$PATH:/usr/local/Ascend/toolbox/latest/Ascend-DMI/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/dcmi
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/Ascend/driver/lib64
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/Ascend/driver/lib64/driver
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/Ascend/toolbox/5.0.RC3/Ascend-DMI/lib64/
```

步骤 10 执行如下命令验证工具是否安装正确。

```
[root@mail ~]# ascend-dmi -c
```

若能显示如下内容，则表示配置正确。

```
[root@mail ~]# ascend-dmi -c
=====
                        System Information
=====
Architecture           | aarch64
-----
Type                    | Atlas 300I Pro
=====
Compatibility Check Result: Compatible
=====
Package                 | Version           | Status      | Innerversion      | Dependencies
-----
npd-driver              | 23.0.1           | OK          | V100R001C15SPC004B220 | NA
npd-firmware           | 7.1.0.4.220     | OK          | NA                 | NA
toolkit                 | 7.0.RC1          | OK          | V100R001C13SPC005B246 | NA
toolbox                 | 5.0.RC3.3       | OK          | NA                 | NA
=====
```

## 5.2.3 AI 算力测试

步骤 1 执行如下命令查询推理卡的 DeviceID。

```
[root@mail ~]# ascend-dmi --info
```

```
[root@mail ~]# ascend-dmi --info
```

ascend-dmi		5.0.RC3.3		Brief Information	
Card	Type	NPU Count		Real-time Card Power	
Chip Name	Health	Used Memory		Temperature	Voltage
Device ID	Bus ID	AI Core Usage			
1	Atlas 300I Pro	1		17.1W	
0	Ascend 310P3	OK	1774MB/21527MB	52C	0.79V
0		0000:01:00.0	0%		
2	Atlas 300I Pro	1		16.3W	
0	Ascend 310P3	OK	1774MB/21527MB	53C	0.79V
1		0000:02:00.0	0%		
4	Atlas 300I Pro	1		18.0W	
0	Ascend 310P3	OK	1778MB/21527MB	57C	0.79V
2		0000:81:00.0	0%		
5	Atlas 300I Pro	1		17.0W	
0	Ascend 310P3	OK	1775MB/21527MB	56C	0.79V
3		0000:82:00.0	0%		
7	Atlas 300I Pro	1		16.5W	
0	Ascend 310P3	OK	1775MB/21527MB	57C	0.79V
4		0000:83:00.0	0%		
8	Atlas 300I Pro	1		16.0W	
0	Ascend 310P3	OK	1777MB/21527MB	57C	0.79V
5		0000:84:00.0	0%		
6	Atlas 300I Pro	1		18.5W	
0	Ascend 310P3	OK	1775MB/21527MB	62C	0.79V
6		0000:85:00.0	0%		

步骤 2 执行 `ascend-dmi -f -t int8 -d 0` 命令来测试昇腾推理卡 0 号卡的整数精度 (INT8) 算力。若返回如下图所示信息, 表示工具运行正常。

```
[root@mail ~]# ascend-dmi -f -t int8 -d 0
```

```
[root@mail ~]# ascend-dmi -f -t int8 -d 0
```

Device	Execute Times	Duration(ms)	TOPS@INT8	Power (W)
0	80,000,000	1185	141.580	60.2000008

步骤 3 执行 `ascend-dmi -f -d 0` 命令来测试昇腾推理卡 0 号卡的半精度 (FP16) 算力。

```
[root@mail ~]# ascend-dmi -f -d 0
```

```
[root@mail ~]# ascend-dmi -f -d 0
```

Device	Execute Times	Duration(ms)	TFLOPS@FP16	Power (W)
0	80,000,000	2370	70.790	68.8000031

## 5.2.4 推理应用示例

步骤 1 将 samples-master.zip 包上传至服务器/home 目录。

步骤 2 执行 unzip samples-master.zip 解压 sample 代码仓 zip 包。

```
[root@mail ~]# cd /home
[root@mail home]# unzip samples-master.z ip
```

步骤 3 执行如下命令进入样例目录。

```
[root@mail home]# cd /home/samples-
master/cplusplus/level2_simple_inference/1_classification/resnet50_imagenet_classification
```

说明：下文中的“样例目录”均指“/home/samples-master/cplusplus/level2\_simple\_inference/1\_classification/resnet50\_imagenet\_classification”目录。

步骤 4 执行 mkdir -p caffe\_model 创建 caffe\_model 目录。

```
[root@mail resnet50_imagenet_classification]# mkdir -p caffe_model
```

步骤 5 将 resnet50.caffemodel 和 resnet50.prototxt 2 个 ResNet-50 网络的模型文件（\*.prototxt）、权重文件（\*.caffemodel）上传至 caffe\_model 目录。

步骤 6 在“样例目录”下执行如下命令，将 ResNet-50 原始模型转换为适配昇腾 AI 处理器的离线模型（\*.om 文件）。

```
[root@mail resnet50_imagenet_classification]# atc --model=caffe_model/resnet50.prototxt --
weight=caffe_model/resnet50.caffemodel --framework=0 --output=model/resnet50 --
soc_version=Ascend310P3 --input_format=NCHW --input_fp16_nodes=data --output_type=FP32 --
out_nodes=prob:0
```

步骤 7 将样例测试图片 dog1\_1024\_683.jpg 和 dog2\_1024\_683.jpg 上传至/home/samples-master/cplusplus/level2\_simple\_inference/1\_classification/resnet50\_imagenet\_classification/data 目录下。（测试样例图为手册后面的两张图片，请按顺序保存并命名）

步骤 8 执行如下命令进入 data 目录，执行如下命令将\*.jpg 转换为\*.bin。

```
[root@mail resnet50_imagenet_classification]# cd data/
[root@mail data]# python3 ../script/transferPic.py
```

```
[root@mail resnet50_imagenet_classification]# cd data/
[root@mail data]# ls
dog1_1024_683.jpg dog2_1024_683.jpg
[root@mail data]# python3 ../script/transferPic.py
start to process image dog1_1024_683.jpg...
process image dog1_1024_683.jpg successfully
start to process image dog2_1024_683.jpg...
process image dog2_1024_683.jpg successfully
2 images in total, 2 images process successfully
```

步骤 9 执行如下命令，进入“样例目录”下。

```
[root@mail data]# cd /home/samples-  
master/cplusplus/level2_simple_inference/1_classification/resnet50_imagenet_classification
```

步骤 10 执行如下命令配置环境变量。

```
[root@mail resnet50_imagenet_classification]# export DDK_PATH=/usr/local/Ascend/ascend-  
toolkit/latest  
[root@mail resnet50_imagenet_classification]# export NPU_HOST_LIB=$DDK_PATH/runtime/lib64/stub
```

步骤 11 在“样例目录”下，执行如下命令创建存放编译文件的目录。

```
[root@mail resnet50_imagenet_classification]# mkdir -p build/intermediates/host
```

步骤 12 在“样例目录”下，执行如下命令生成可执行文件，生成的可执行文件在样 out 目录下。

```
[root@mail resnet50_imagenet_classification]# cd build/intermediates/host  
[root@mail host]# cmake ../../src -DCMAKE_CXX_COMPILER=g++ -DCMAKE_SKIP_RPATH=TRUE  
[root@mail host]# make
```

```
[root@mail resnet50_imagenet_classification]# cd build/intermediates/host  
[root@mail host]# cmake ../../src -DCMAKE_CXX_COMPILER=g++ -DCMAKE_SKIP_RPATH=TRUE  
-- The C compiler identification is GNU 10.3.1  
-- The CXX compiler identification is GNU 10.3.1  
-- Detecting C compiler ABI info  
-- Detecting C compiler ABI info - done  
-- Check for working C compiler: /usr/bin/cc - skipped  
-- Detecting C compile features  
-- Detecting C compile features - done  
-- Detecting CXX compiler ABI info  
-- Detecting CXX compiler ABI info - done  
-- Check for working CXX compiler: /usr/bin/g++ - skipped  
-- Detecting CXX compile features  
-- Detecting CXX compile features - done  
-- env INC_PATH: /usr/local/Ascend/ascend-toolkit/latest  
-- env LIB_PATH: /usr/local/Ascend/ascend-toolkit/latest/runtime/lib64/stub  
-- Configuring done  
-- Generating done  
-- Build files have been written to: /home/samples-master/cplusplus/level2_simple_inference/1_classification/resnet50_imagenet_classification/build/intermediates/host  
[root@mail host]# make  
[ 20%] Building CXX object CMakeFiles/main.dir/utlis.cpp.o  
[ 40%] Building CXX object CMakeFiles/main.dir/model_process.cpp.o  
[ 60%] Building CXX object CMakeFiles/main.dir/sample_process.cpp.o  
[ 80%] Building CXX object CMakeFiles/main.dir/main.cpp.o  
[100%] Linking CXX executable /home/samples-master/cplusplus/level2_simple_inference/1_classification/resnet50_imagenet_classification/out/main  
[100%] Built target main
```

步骤 13 执行如下命令运行可执行文件。

```
[root@mail host]# cd /home/samples-  
master/cplusplus/level2_simple_inference/1_classification/resnet50_imagenet_classification/out  
[root@mail out]# ./main
```

```
[root@mail host]# cd /home/samples-master/cplusplus/level2_simple_inference/1_classification/resnet50_imagenet_classification/out
[root@mail out]# ./main
[INFO] acl init success
[INFO] set device 0 success
[INFO] create context success
[INFO] create stream success
[INFO] get run mode success
[INFO] load model ../model/resnet50.om success
[INFO] create model description success
[INFO] create model input success
[INFO] create model output success
[INFO] start to process file:../data/dog1_1024_683.bin
[INFO] model execute success
[INFO] top 1: index[161] value[0.766602]
[INFO] top 2: index[162] value[0.155762]
[INFO] top 3: index[167] value[0.038452]
[INFO] top 4: index[163] value[0.021576]
[INFO] top 5: index[166] value[0.011642]
[INFO] output data success
[INFO] start to process file:../data/dog2_1024_683.bin
[INFO] model execute success
[INFO] top 1: index[267] value[0.936035]
[INFO] top 2: index[266] value[0.041107]
[INFO] top 3: index[265] value[0.018982]
[INFO] top 4: index[219] value[0.002628]
[INFO] top 5: index[160] value[0.000297]
[INFO] output data success
[INFO] destroy model input success
[INFO] destroy model output success
[INFO] unload model success, modelId is 1
[INFO] destroy model description success
[INFO] execute sample success
[INFO] end to destroy stream
[INFO] end to destroy context
[INFO] end to reset device 0
[INFO] end to finalize acl
```

说明:类别标签和类别的对应关系与训练模型时使用的数据集有关, 本样例使用的模型是基于 imagenet 数据集进行训练的。图中圈中的计算结果中, index 值为不同犬编号, value 值为相似概率。

dog1\_1024\_683.jpg:



表5-2 dog1\_1024\_683.jpg 推理结果

类别标签	类别 (推理结果)
161	巴吉特, 巴吉特猎犬
162	小猎犬
167	英国猎狐犬

163	猎犬, 侦查犬
166	沃克猎犬, 沃克猎犬

dog2\_1024\_683.jpg:



表5-3 dog2\_1024\_683.jpg 推理结果

类别标签	类别 (推理结果)
267	标准贵宾犬
266	微型贵宾犬
265	玩具贵宾犬
219	可卡犬, 英国可卡犬, 可卡犬
160	阿富汗猎犬, 阿富汗

### 5.3 思考题

如果涉及到多台服务器, 能否使用 SmartKit 批量部署 Atlas 环境?

答案: 可以。

华为认证 Computing 系列教程

# HCIA-Computing

## 计算产品基础运维

### 实验指导手册

版本：3.0



华为技术有限公司

版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

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。

## 华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://e.huawei.com>

---

## 华为认证体系介绍

华为认证是华为公司基于“平台+生态”战略，围绕“云-管-端”协同的新ICT技术架构，打造的覆盖ICT（Information and Communications Technology，信息通信技术）全技术领域的认证体系，包含ICT基础设施认证、基础软硬件认证、云平台及云服务认证三类认证。

根据ICT从业者的学习和进阶需求，华为认证分为工程师级别、高级工程师级别和专家级别三个认证等级。

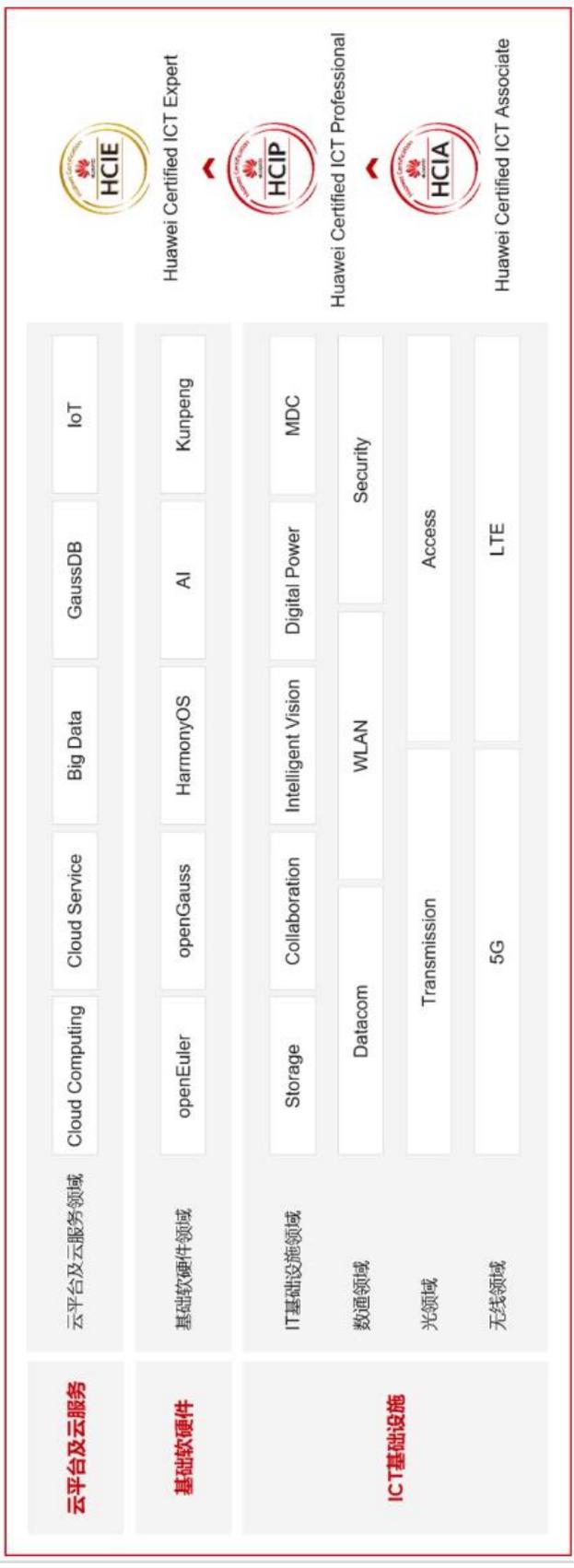
华为认证覆盖ICT全领域，符合ICT融合的技术趋势，致力于提供领先的人才培养体系和认证标准，培养数字化时代新型ICT人才，构建良性ICT人才生态。

华为认证HCIA-Computing V3.0定位于培养与认证具备计算产品安装与基础维护能力的计算工程师。

通过HCIA-Computing V3.0认证，您将了解计算产业的发展历史及产业前景，理解华为计算产业的愿景和目标，掌握计算系统及服务器产品相关知识，具备计算系统及业务系统搭建、基础维护能力，胜任计算产品售前工程师、计算产品交付工程师、IT工程师等岗位。

---

# 华为职业认证



## 简介

本书为 HClA-Computing 认证培训教程，适用于准备参加 HClA-Computing 考试的学员或者希望了解 iBMA、iBMC、Smart Provisioning、SmartKit、FusionDirector 计算产品运维工具使用的读者。

## 内容描述

本实验指导书共包含 4 个实验，逐一介绍了 iBMA、iBMC、Smart Provisioning、SmartKit、FusionDirector 计算产品运维工具的安装与使用。

- 实验一为 iBMC 使用，包括 iBMC 安装 iBMA、iBMC 管理 Smart Provisioning。
- 实验二为 Smart Provisioning 使用，包括固件升级、硬件诊断与日志收集。
- 实验三为 SmartKit 安装与使用，包括安装 SmartKit、初始化 SmartKit、添加服务器设备、配置服务器 iBMC、配置服务器 BIOS、服务器电源控制、服务器配置检查、服务器健康检查、服务器日志收集。
- 实验四为 FusionDirector 安装与使用，包括安装 FusionDirector、初始化 FusionDirector、上传根证书及二级 CA 证书、添加服务器、服务器基本操作、服务器监控信息。

## 读者知识背景

本课程为华为认证基础课程，为了更好地掌握本书内容，阅读本书的读者应首先具备以下基本条件：

- 具有服务器基本知识，了解服务器架构及部件。
- 熟悉 Linux 基本命令行。
- 熟悉操作系统基本原理。
- 了解虚拟化基本原理。

## 实验环境说明

组网说明

---

本实验环境面向准备 HCIA-Computing 考试的 IT 工程师。每套实验环境包括 1 台 Atlas 800 ( Model 3000 ) 或 TaiShan 200 ( 2280 ) 服务器。

#### 设备介绍

为了满足 HCIA-Computing 实验需要，建议每套实验环境采用以下配置：

设备名称、型号与版本的对应关系如下：

设备名称	设备型号	软件版本
Atlas 800	Model 3000	OS: openEuler 22.03 LTS

【注】如无 Atlas 800 ( Model 3000 ) 服务器，也可用 TaiShan 200 ( 2280 ) 服务器完成本实验。

## 准备实验环境

### 检查设备

实验开始之前请每组学员检查自己的实验设备是否齐全，实验清单如下：

设备名称	数量	备注
Atlas 800	1台	1名学员1台
PC	1台	1名学员1台，Windows系统

# 目录

---

<b>前 言</b> .....	<b>3</b>
简介.....	3
内容描述.....	3
读者知识背景.....	3
实验环境说明.....	3
准备实验环境.....	4
<b>1 实验环境介绍</b> .....	<b>7</b>
1.1 实验介绍.....	7
1.1.1 关于本实验.....	7
1.1.2 实验组网介绍.....	7
1.1.3 实验设备介绍.....	8
1.1.4 实验软件介绍.....	8
<b>2 计算产品基础运维</b> .....	<b>9</b>
2.1 iBMC 使用.....	9
2.1.1 iBMC 安装 iBMA.....	9
2.1.2 iBMC 管理 Smart Provisioning.....	14
2.2 Smart Provisioning 使用.....	18
2.2.1 固件升级.....	18
2.2.2 硬件诊断.....	25
2.2.3 日志收集.....	30
2.3 SmartKit 安装与使用.....	37
2.3.1 安装 SmartKit (可选).....	37
2.3.2 初始化 SmartKit (可选).....	41
2.3.3 添加服务器设备.....	44
2.3.4 配置服务器 iBMC.....	46
2.3.5 配置服务器 BIOS.....	57
2.3.6 服务器电源控制.....	63
2.3.7 服务器配置检查.....	71
2.3.8 服务器健康检查.....	79
2.3.9 服务器日志收集.....	90
2.4 FusionDirector 安装与使用.....	95

---



2.4.1 安装 FusionDirector .....	95
2.4.2 初始化 FusionDirector .....	102
2.4.3 上传根证书及二级 CA 证书.....	109
2.4.4 添加服务器 .....	112
2.4.5 服务器基本操作 .....	115
2.4.6 服务器监控信息 .....	118
<b>3 思考题.....</b>	<b>123</b>
<b>4 缩略语.....</b>	<b>124</b>

---

# 1 实验环境介绍

## 1.1 实验介绍

### 1.1.1 关于本实验

华为计算产品运维工具包含了单机带内管理的 iBMA，单机带外管理的 iBMC 和 Smart Provisioning，集中管理的 FusionDirector 和 eSight，近端运维的 SmartKit，远端运维的 eService，本实验将聚焦在 iBMA、iBMC、Smart Provisioning、SmartKit 和 FusionDirector 这 5 个运维工具，涉及到工具的安装与使用。

本实验通过 PC 直连 Atlas800 服务器（openEuler 22.03 LTS），对服务器进行 iBMC、iBMA 和 Smart Provisioning 实验操作，在 PC 上安装 SmartKit 对 Atlas800 服务器进行近端运维，在服务器上通过虚拟化技术创建 FusionDirector 虚拟机，并通过 FusionDirector 对服务器进行集中管理。

### 1.1.2 实验组网介绍



图1-1 计算产品基础运维实验拓扑图

### 1.1.3 实验设备介绍

设备名称、型号与版本的对应关系如下：

表1-1 实验设备配套关系

设备名称	设备型号	软件版本
Atlas 800	Model 3000	OS: openEuler 22.03 LTS

【注】如无 Atlas 800 ( Model 3000 ) 服务器，也可用 TaiShan 200 ( 2280 ) 服务器完成本实验。

### 1.1.4 实验软件介绍

软件名称与版本对应关系如下：

表1-2 实验软件配套关系

设备名称	软件版本
iBMA	2.11.0
iBMC	3.11.00.09
Smart Provisioning	1.9.2
SmartKit	23.0.0.6_zh
FusionDirector	23.1.0.SPC3_ENT

【注】相关软件可以在华为人才在线 HCIA-Computing V3.0 认证界面进行下载。

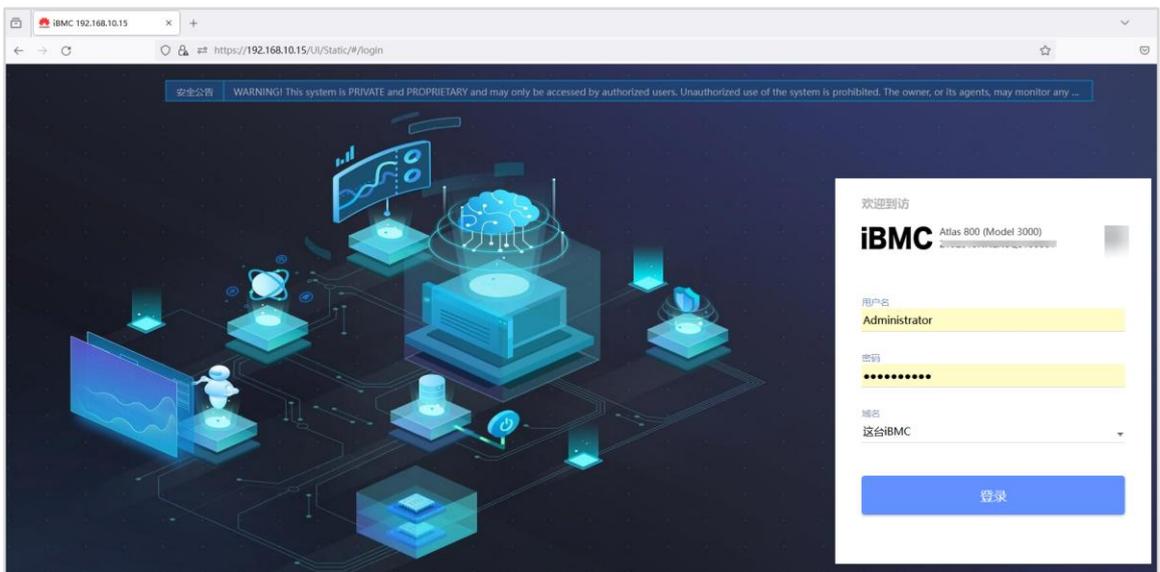
---

# 2 计算产品基础运维

## 2.1 iBMC 使用

### 2.1.1 iBMC 安装 iBMA

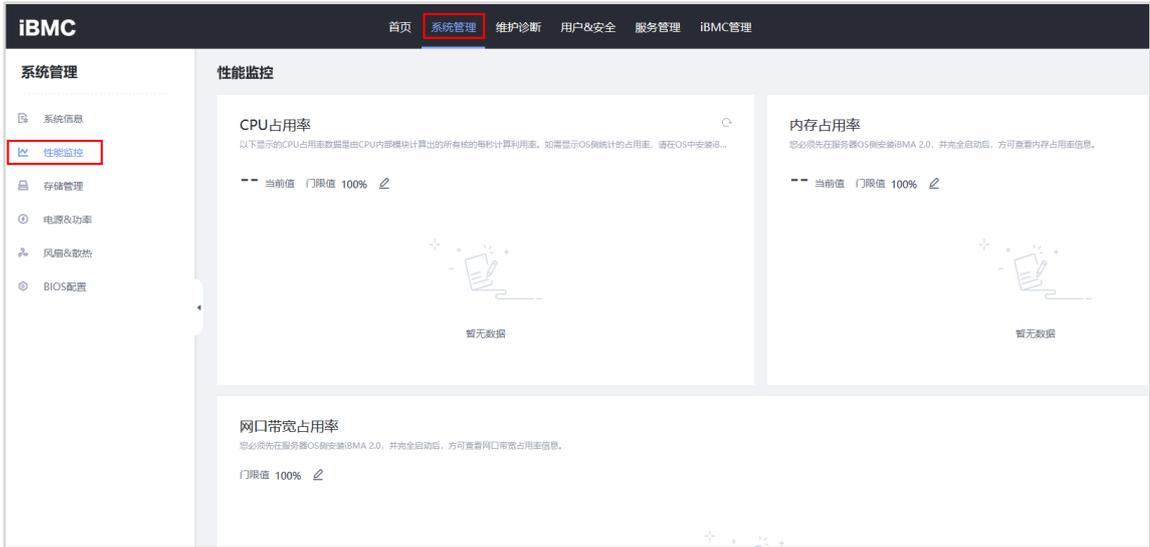
步骤 1 在浏览器里输入服务器 iBMC 地址，并使用 Administrator 用户进行登录。



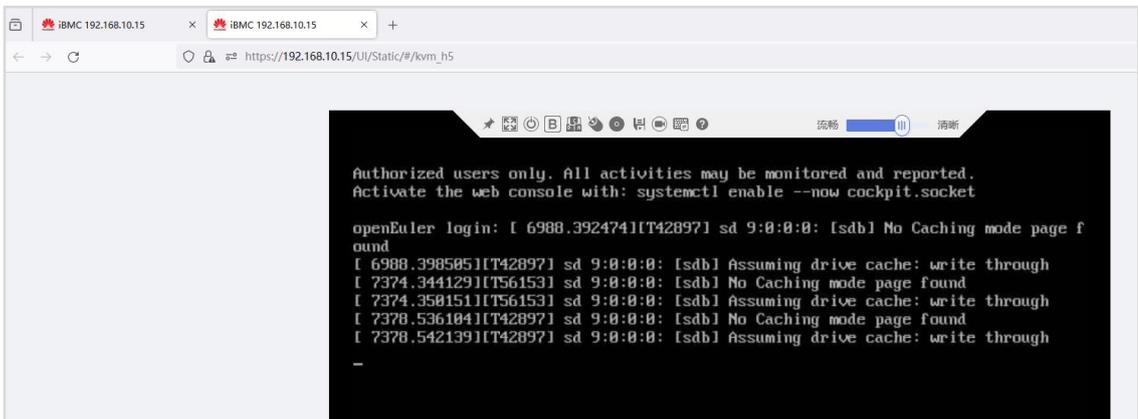
步骤 2 由于缺失 iBMA 模块，在“首页”里，系统监控无法监控 CPU、内存、磁盘利用率。



步骤 3 在“系统管理 > 性能监控”界面，同样也无法查看相关检测数据。



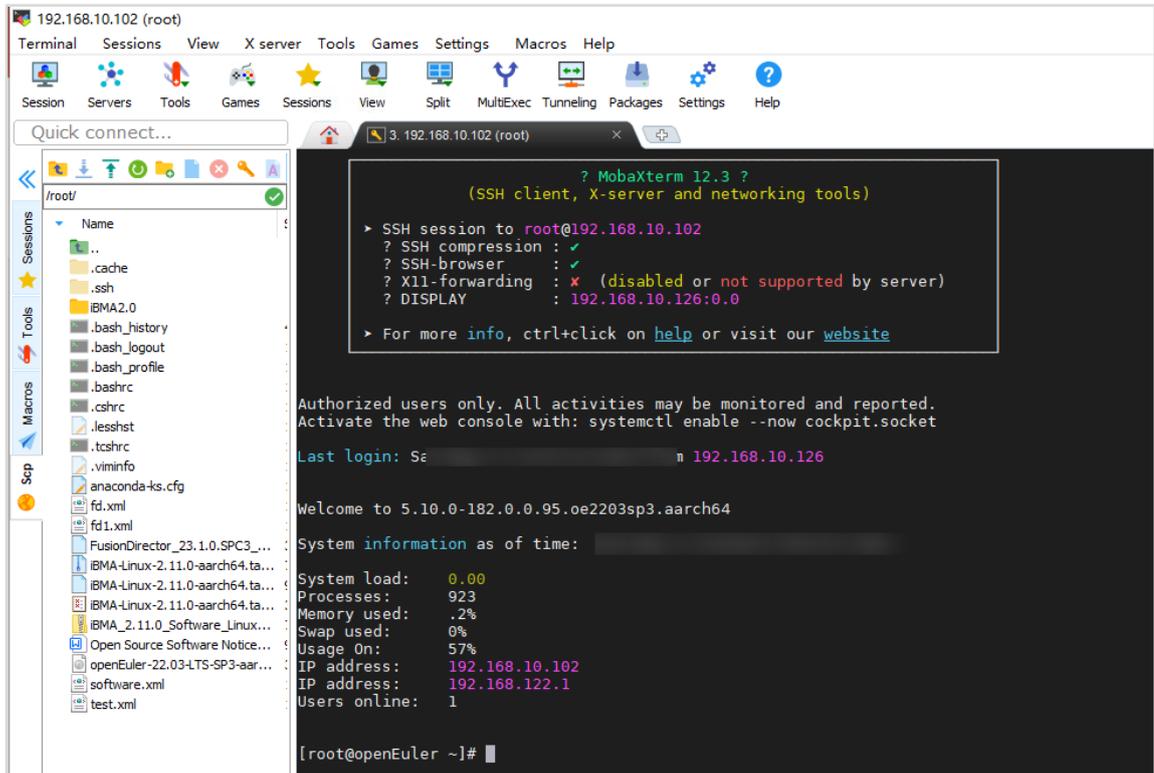
步骤 4 在“iBMC 管理 > iBMA 管理”界面，点击“安装 iBMA”，系统会自动弹出“虚拟控制台”界面。



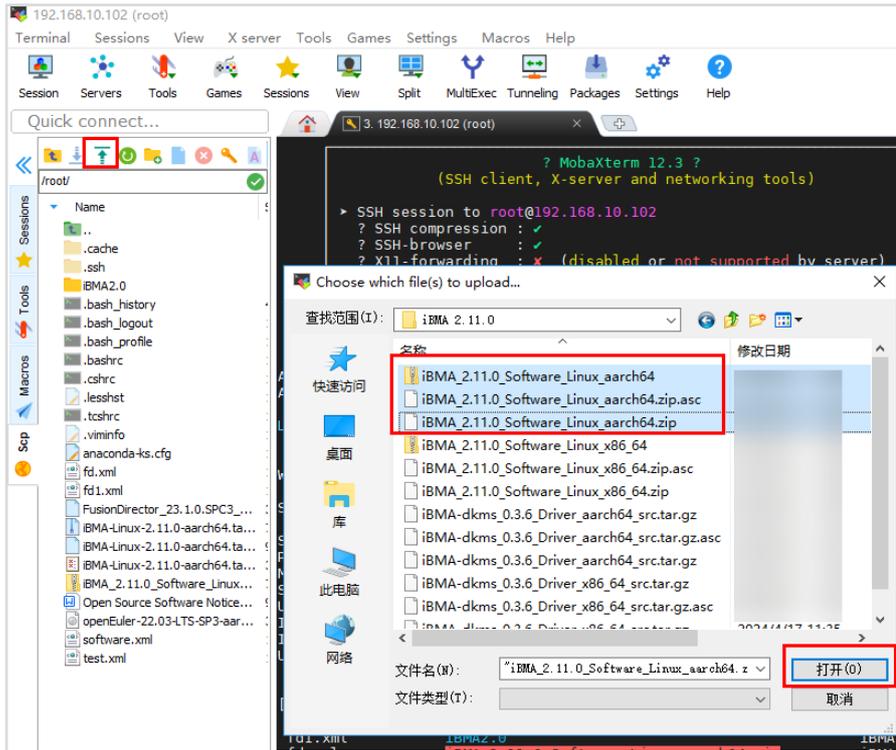
【注】

- 由于服务器有管理 IP 可以登录，本次实验不通过虚拟控制台，使用 SSH 工具进行登录，操作起来会更加方便。
- 默认的 iBMA 版本过低，本次实验采用上传 2.11.0 版本的 iBMA 离线安装包的形式进行安装。

### 步骤 5 使用 MobaXterm 工具 SSH 登录服务器 OS 后台。



### 步骤 6 使用 MobaXterm 的 SFTP 功能上传 “iBMA\_2.11.0\_Software\_Linux\_aarch64.zip” 及校验文件到服务器上。



步骤 7 等待文件上传完毕，运行如下命令进行解压。

```
unzip iBMA_2.11.0_Software_Linux_aarch64.zip
```

```
[root@openEuler ~]# unzip iBMA_2.11.0_Software_Linux_aarch64.zip
Archive:  iBMA_2.11.0_Software_Linux_aarch64.zip
 inflating: iBMA-Linux-2.11.0-aarch64.tar.gz
 inflating: iBMA-Linux-2.11.0-aarch64.tar.gz.cms
 inflating: iBMA-Linux-2.11.0-aarch64.tar.gz.crl
 inflating: Open Source Software Notice.doc
 inflating: software.xml
[root@openEuler ~]#
```

【注】

- 解压 zip 文件后得到的 cms 和 crl 用于对文件包做 CMS 校验。
- iBMA 自带的 CMS 校验文件放在“./lib/Linux/upgrade”目录下，可以使用此工具对下载的安装包进行校验。
- “software.xml”文件用来描述安装包的版本信息等。
- 解压后的“iBMA-Linux-2.3.3.tar.gz”为安装包文件。
- “Open Source Software Notice.doc”为开源软件使用声明。

步骤 8 运行如下命令，解压安装包。

```
tar xzf iBMA-Linux-2.11.0-aarch64.tar.gz
```

```
[root@openEuler ~]# tar xzf iBMA-Linux-2.11.0-aarch64.tar.gz
[root@openEuler ~]# ls
anaconda-ks.cfg  FusionDirector_23.1.0.SPC3_ENT_aarch64.qcow2  iBMA-Linux-2.11.0-aarch64.tar.gz  openEuler-22.03-LTS-SP3-aarch64-dvd.iso  test.xml
fd1.xml         iBMA2.0                                         iBMA-Linux-2.11.0-aarch64.tar.gz.cms  'Open Source Software Notice.doc'
fd.xml         iBMA_2.11.0_Software_Linux_aarch64.zip        iBMA-Linux-2.11.0-aarch64.tar.gz.crl  software.xml
[root@openEuler ~]#
```

步骤 9 运行如下命令，安装 iBMA。

```
cd iBMA2.0/
ls
./install.sh -s
```

```
[root@openEuler ~]# cd iBMA2.0/
[root@openEuler iBMA2.0]# ls
app config drivers install.sh README.TXT script
[root@openEuler iBMA2.0]# ./install.sh -s
-----
Starting to install iBMA in silent mode.
System is openEuler
Kernel version is 5.10.0-182.0.0.95.oe2203sp3.aarch64
Driver package version is 0.3.6
Your environment already has a pre-installed iBMA driver
Installing iBMA ...
-----
iBMA installed successfully.
-----
Starting iBMA service.
Start iBMA service successfully.
-----
[root@openEuler iBMA2.0]# █
```

【注】

- ./install.sh -s [--enable-iBMC\_event=true|false --enable-iBMA\_https=true|false]为静默安装（推荐使用此模式），--enable-iBMC\_event 用于控制是否记录 iBMC 事件，不配置时默认为 false，--enable-iBMA\_https 用于控制 iBMA 的 Redfish 服务是否以 HTTPS 协议启动。
- ./install.sh -c 为自定义安装。
- ./install.sh -f 为强制安装（需配合-s 或-c 参数使用）。
- ./install.sh -u 为升级安装。

步骤 10 进入 “/opt/huawei/ibma” 目录查看安装后的文件。

```
cd /opt/huawei/ibma
ls
```

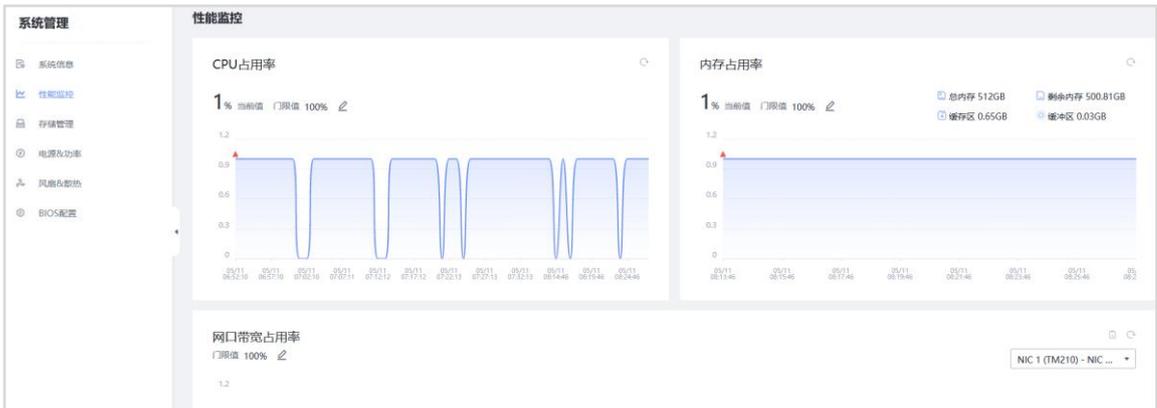
```
[root@openEuler iBMA2.0]# cd /opt/huawei/ibma/
[root@openEuler ibma]# ls
bin bob.sh config ibmacli iBMA.sh lib lib64 log recovery script tools uninstall.sh
[root@openEuler ibma]# █
```

【注】“iBMA.sh” 为 iBMA 服务脚本，“bob.sh” 为 BOB 服务脚本，“uninstall.sh” 为卸载脚本。

步骤 11 iBMA 安装完毕后，返回 iBMC webUI，在 “iBMC 管理 > iBMA 管理” 里，查看 iBMA 状态信息。



步骤 12 在“系统管理 > 性能监控”界面，可以查看详细的监控指标。



步骤 13 在“首页”里，“系统监控”信息同样有监控数据显示。



## 2.1.2 iBMC 管理 Smart Provisioning

步骤 1 在“iBMC 管理 > SP 管理”界面，可对 SP 进行管理。



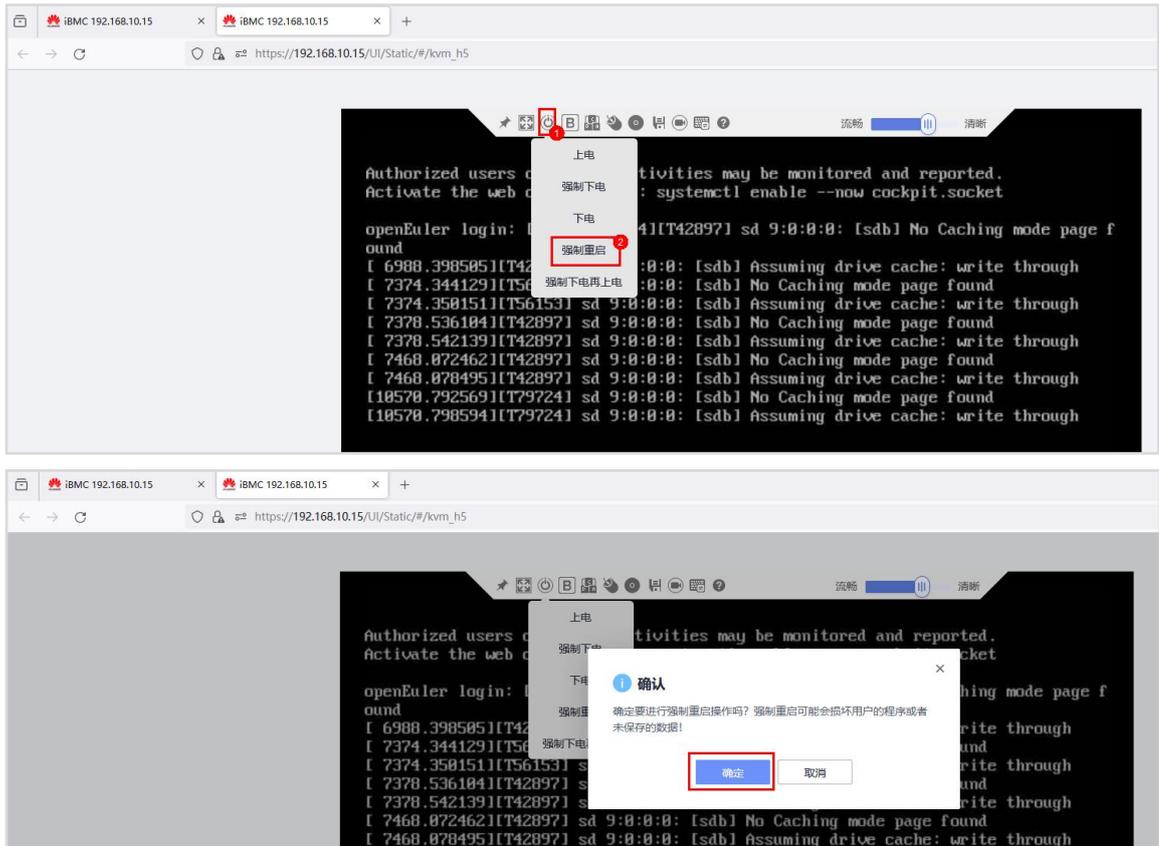
步骤 2 开启 SP 管理的“设备信息收集使能”和“OS 从 SP 启动”两个功能。



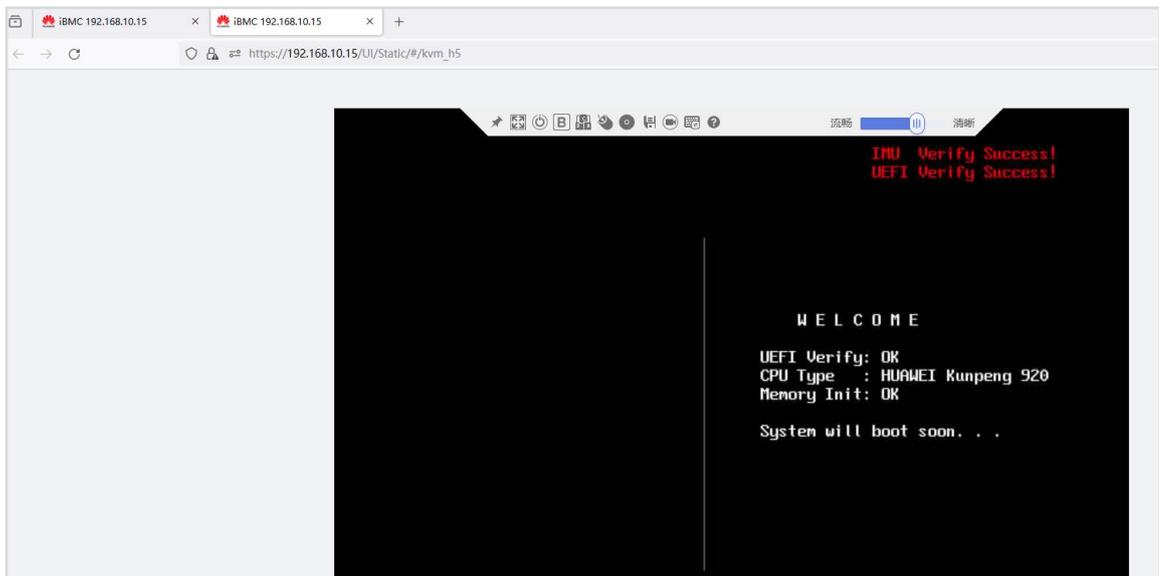
步骤 3 在“首页”里，找到“虚拟控制台”，点击“启动虚拟控制台 > HTML5 集成远程控制台（独占）”。

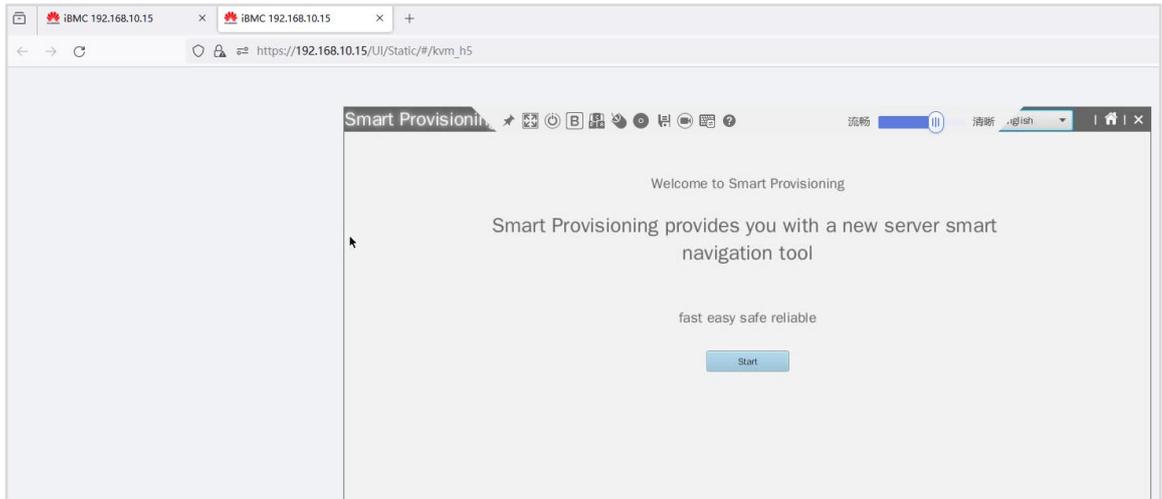


步骤 4 在“虚拟控制台”，点击“电源控制 > 强制重启”，在弹出的“确认”窗口，点击“确定”。



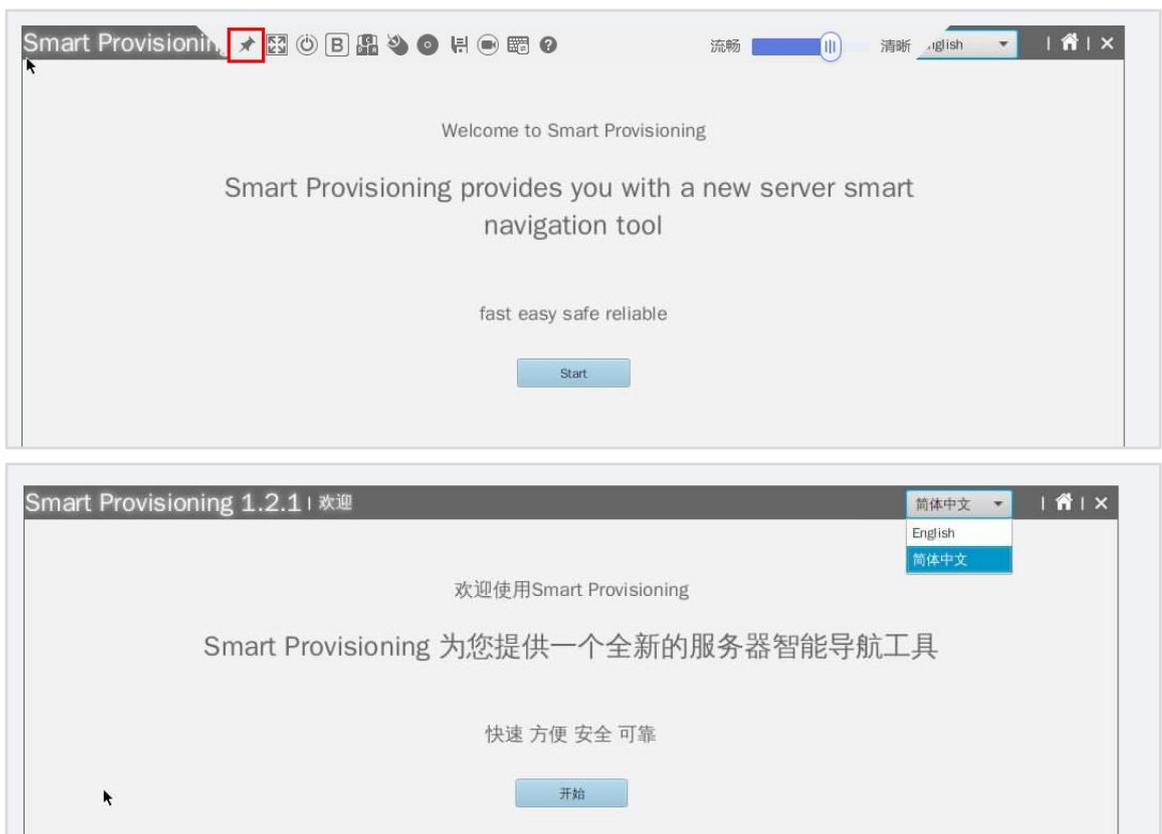
步骤 5 等待片刻，服务器启动后会进入 SmartProvisioning 界面。





【注】如果未自动进入 SP 界面，可在启动界面键盘按“F6”进入 SP，进入 SP 密码默认为“Admin@9000”。

步骤 6 取消虚拟控制台功能栏锁定，并切换 SmartProvisioning 为中文。



## 2.2 Smart Provisioning 使用

### 2.2.1 固件升级

步骤 1 在 SP 欢迎界面，点击“开始”。



步骤 2 点击“升级”，进入升级向导界面。





步骤 3 获取 Smart Provisioning 固件包 “SmartProvisioning\_1.9.2\_aarch64.iso”。

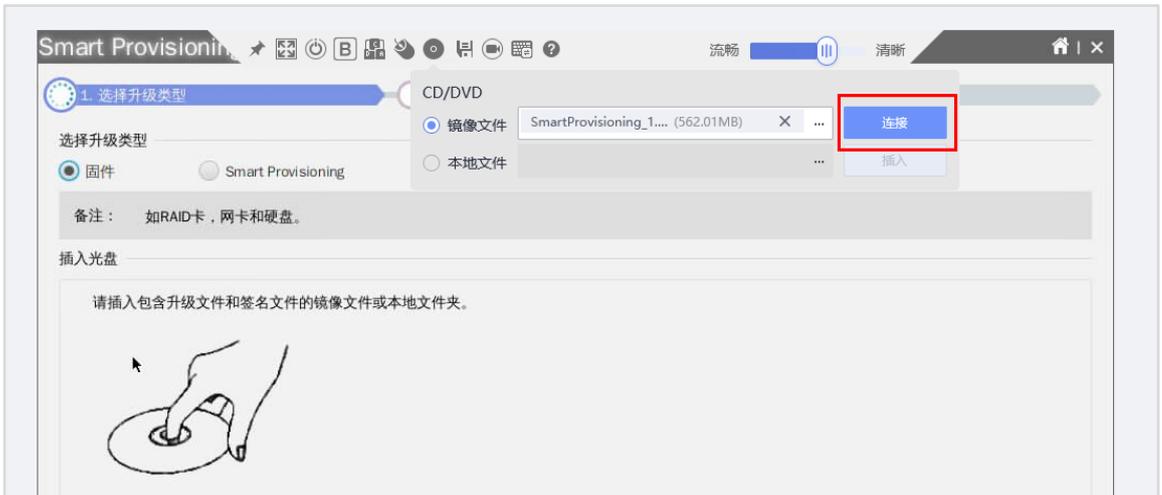
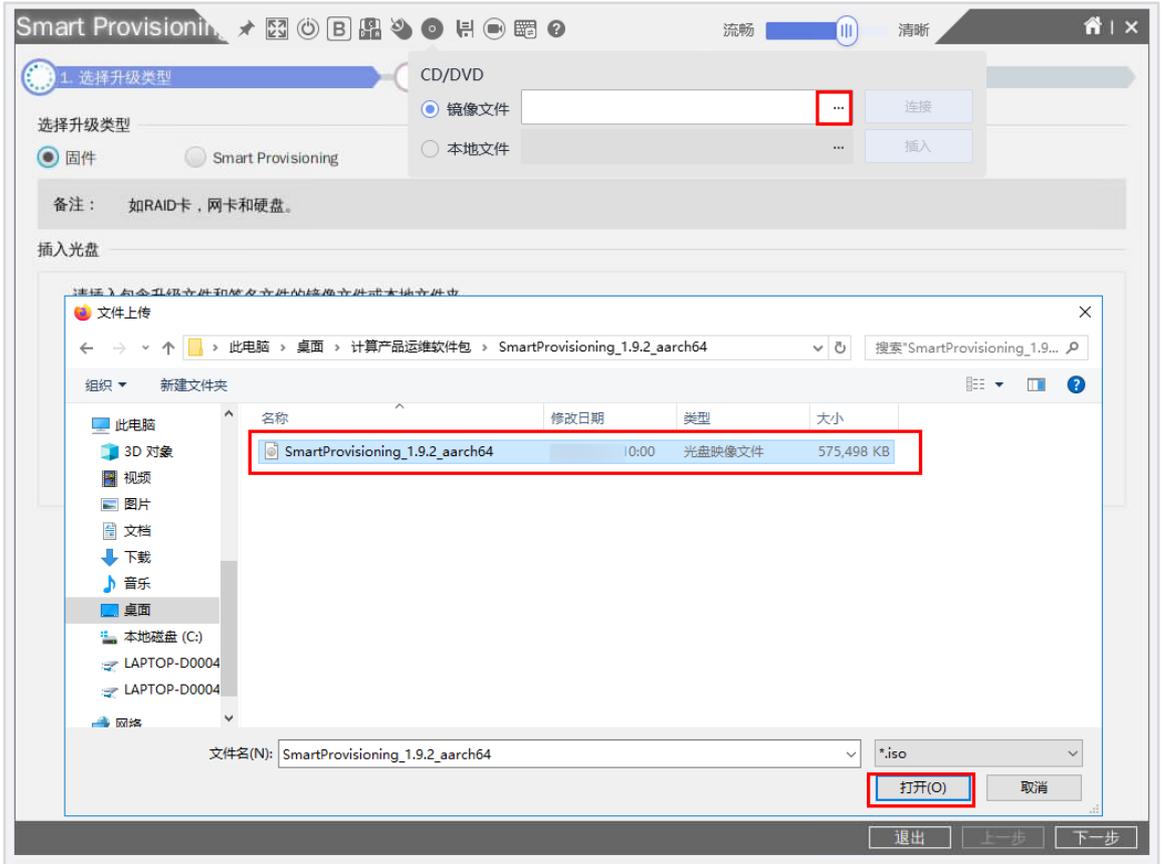


【注】固件包为 zip 格式的压缩包，需进行解压，获取到 iso 的包。

步骤 4 在“虚拟控制台”里，鼠标移到界面上面，功能栏会自动显示，点击“光驱”。



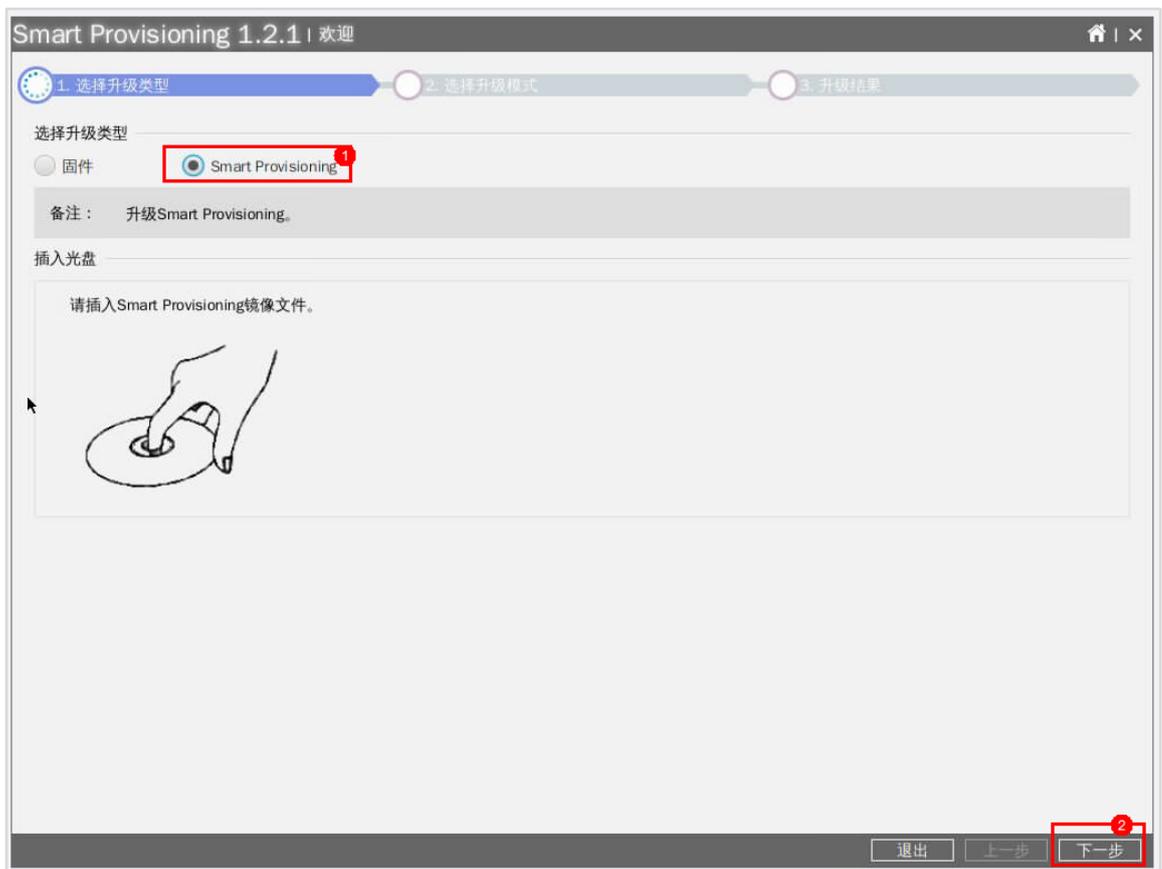
步骤 5 点击“镜像文件”，选择 SP 固件包 iso 文件，点击“连接”。



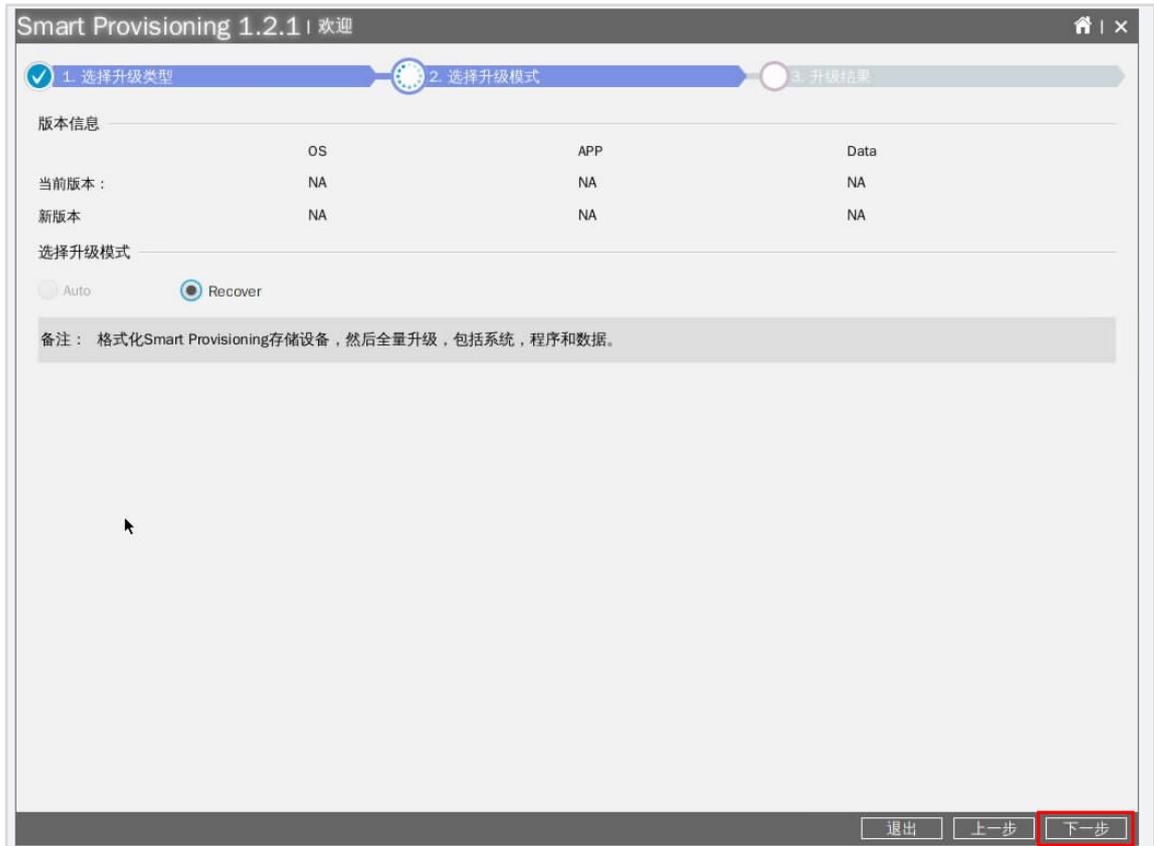
步骤 6 等待片刻，当出现“断开”按钮，说明 iso 文件连接成功。



步骤 7 在“选择升级类型”界面，选择“Smart Provisioning”，点击“下一步”。



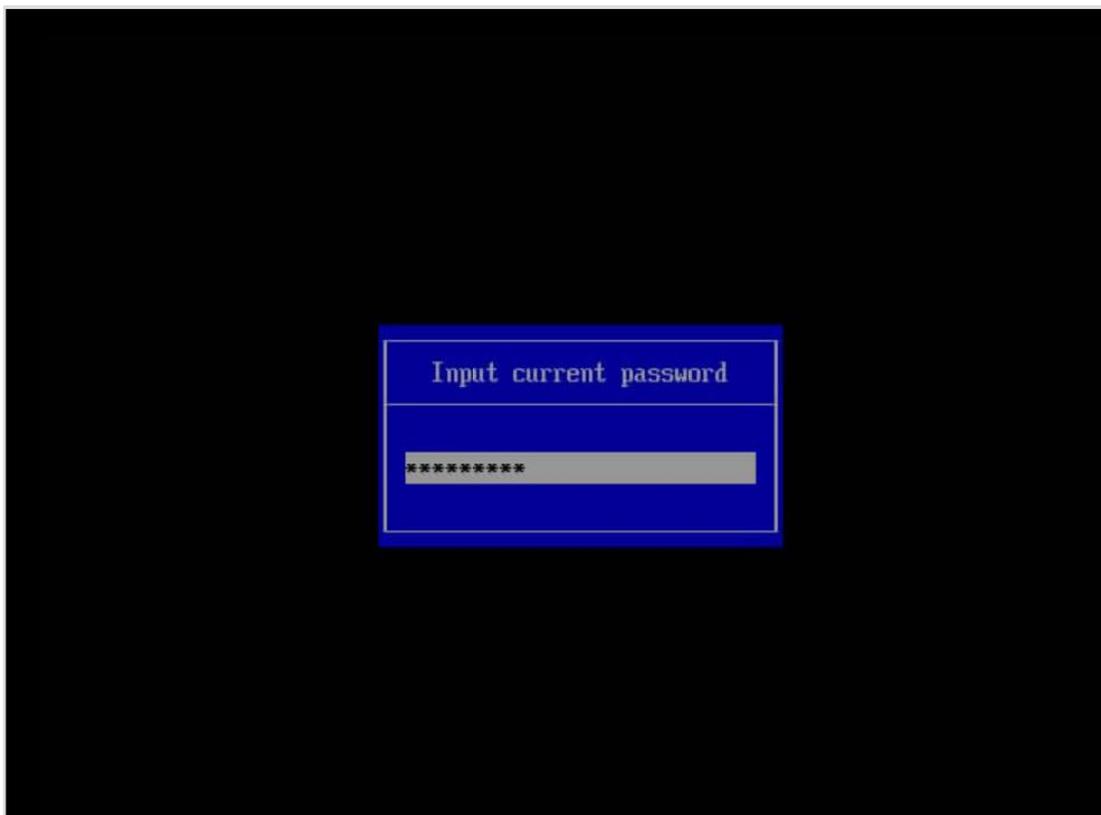
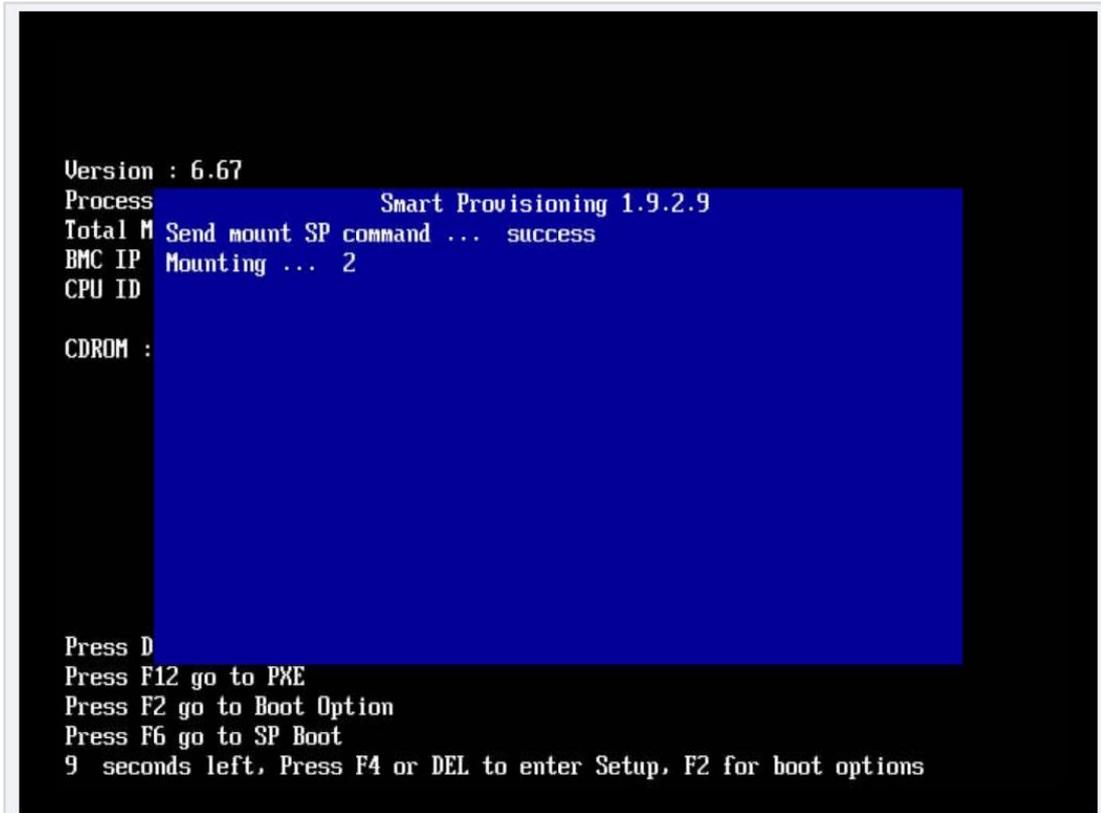
步骤 8 在“选择升级模式”界面，保持默认配置，点击“下一步”。



步骤 9 在“升级结果”界面，等待 SP 固件升级完毕，服务器会自动重启。



步骤 10 在服务器启动界面键盘按“F6”进入 SP，进入 SP 密码默认为“Admin@9000”。





【注】也可以在 iBMC 的“iBMC 管理 > SP 管理”页面，重新开启“OS 从 SP 启动”，服务器启动时会自动进入 SP，单次有效。

步骤 11 在 SP 主页可以看到版本号已经为 1.9.2，修改语言为简体中文，点击“开始”。



## 2.2.2 硬件诊断

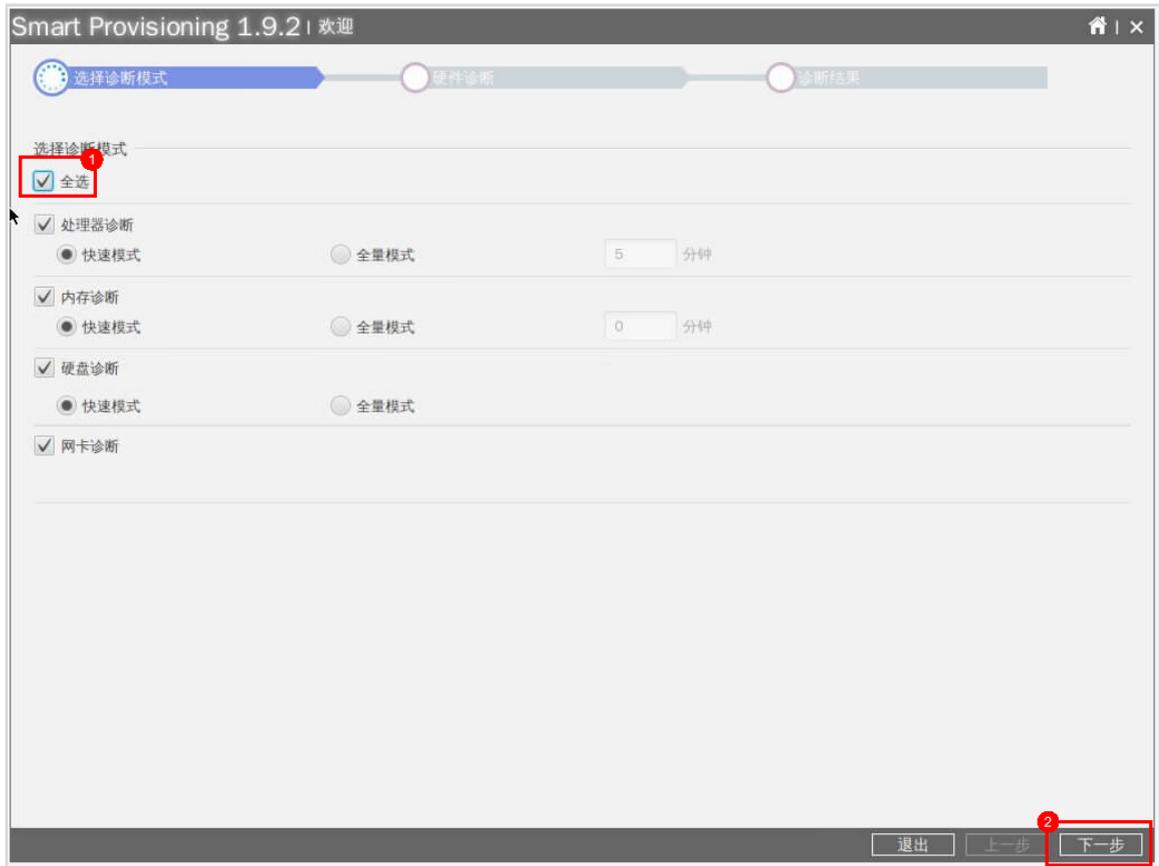
步骤 1 在 SP 功能界面，点击右上角工具箱按钮。



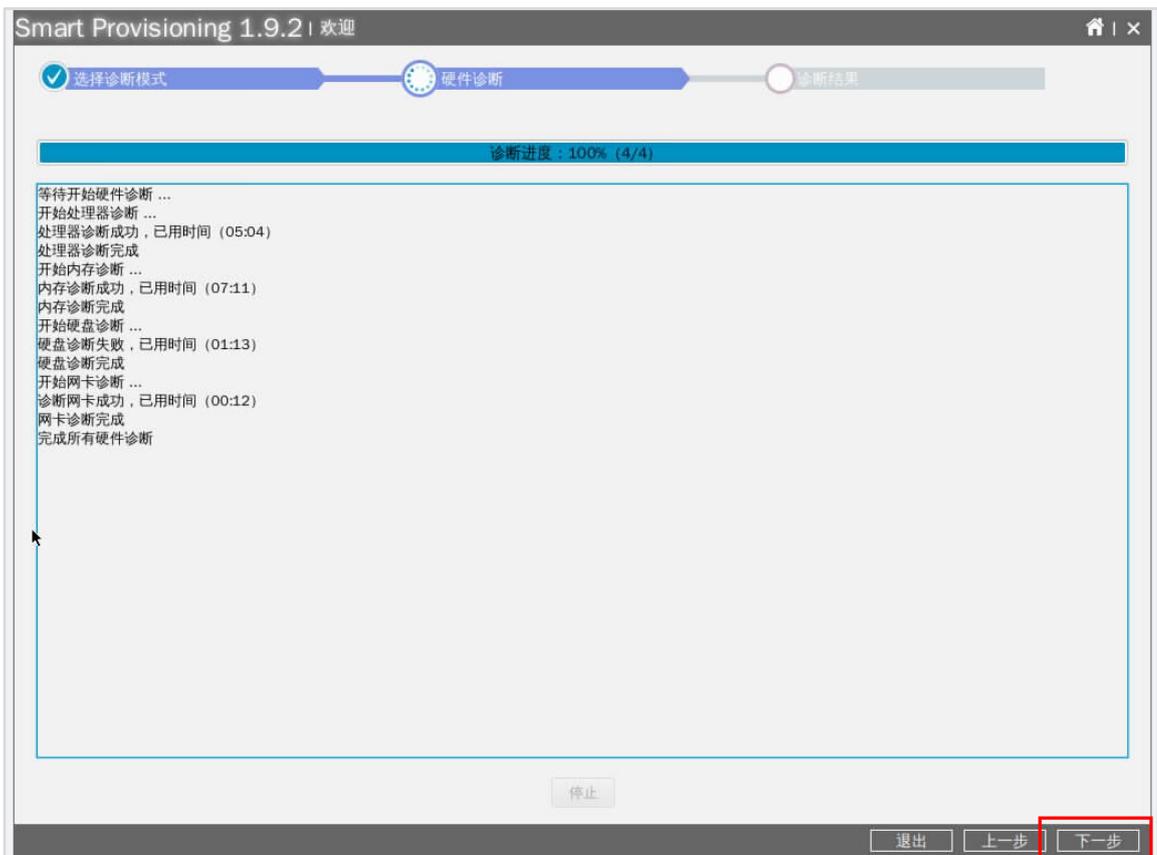
步骤 2 在 SP 工具箱界面，点击“硬件诊断”。



步骤 3 在“选择诊断模式”界面，全选所有诊断项，参数保持“快速模式”，点击“下一步”。

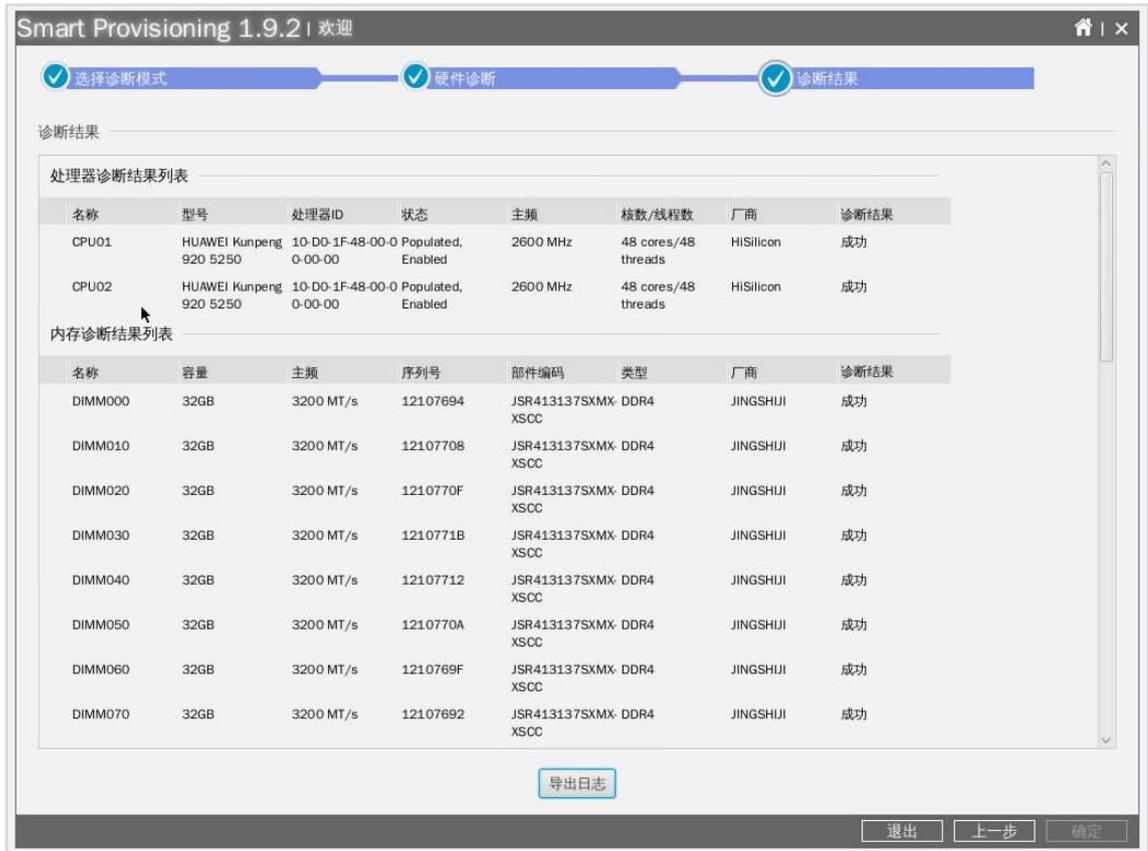


步骤 4 在“硬件诊断”界面，等待诊断进度 100%后，点击“下一步”。

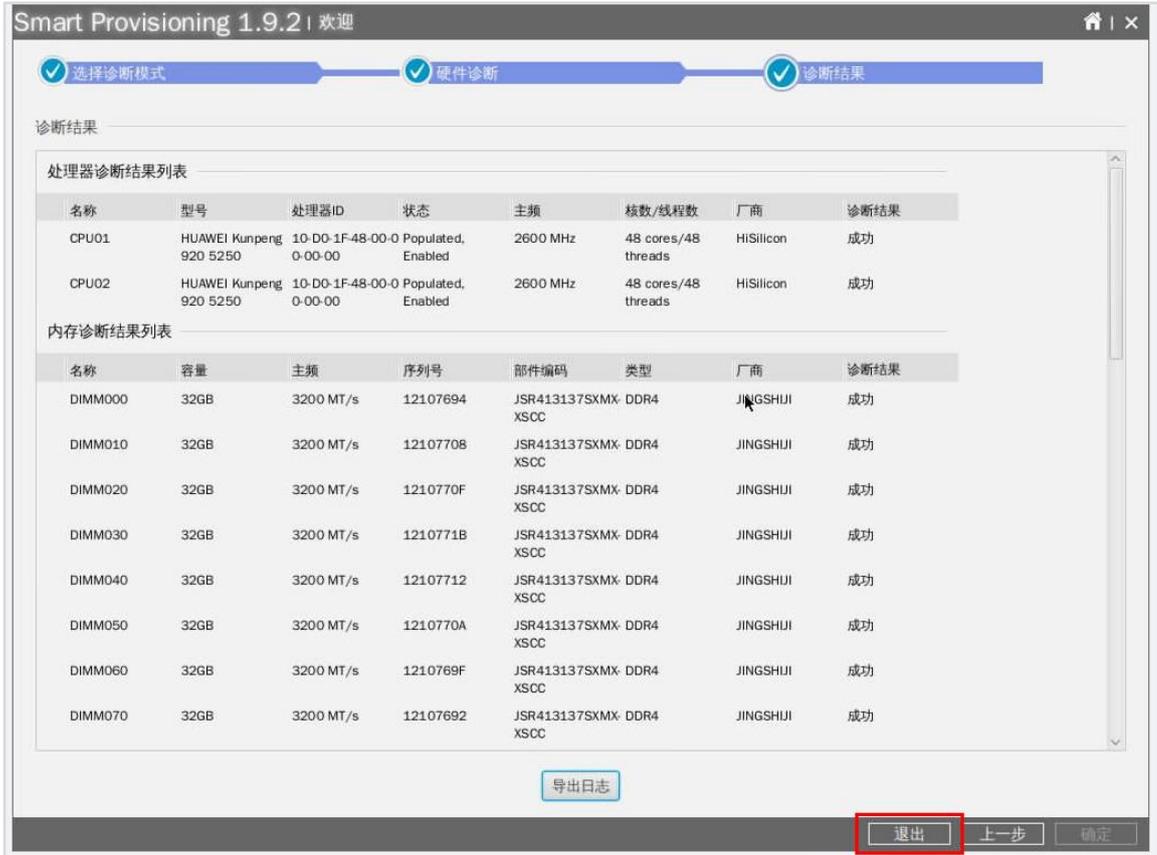


【注】由于全选了处理器、内存、硬盘和网卡的诊断，诊断时间会比较久。

步骤 5 在“诊断结果”界面，可以查看诊断结果详情。



步骤 6 点击“退出”，系统会弹出确认窗口，点击“确定”，结束硬件诊断任务。

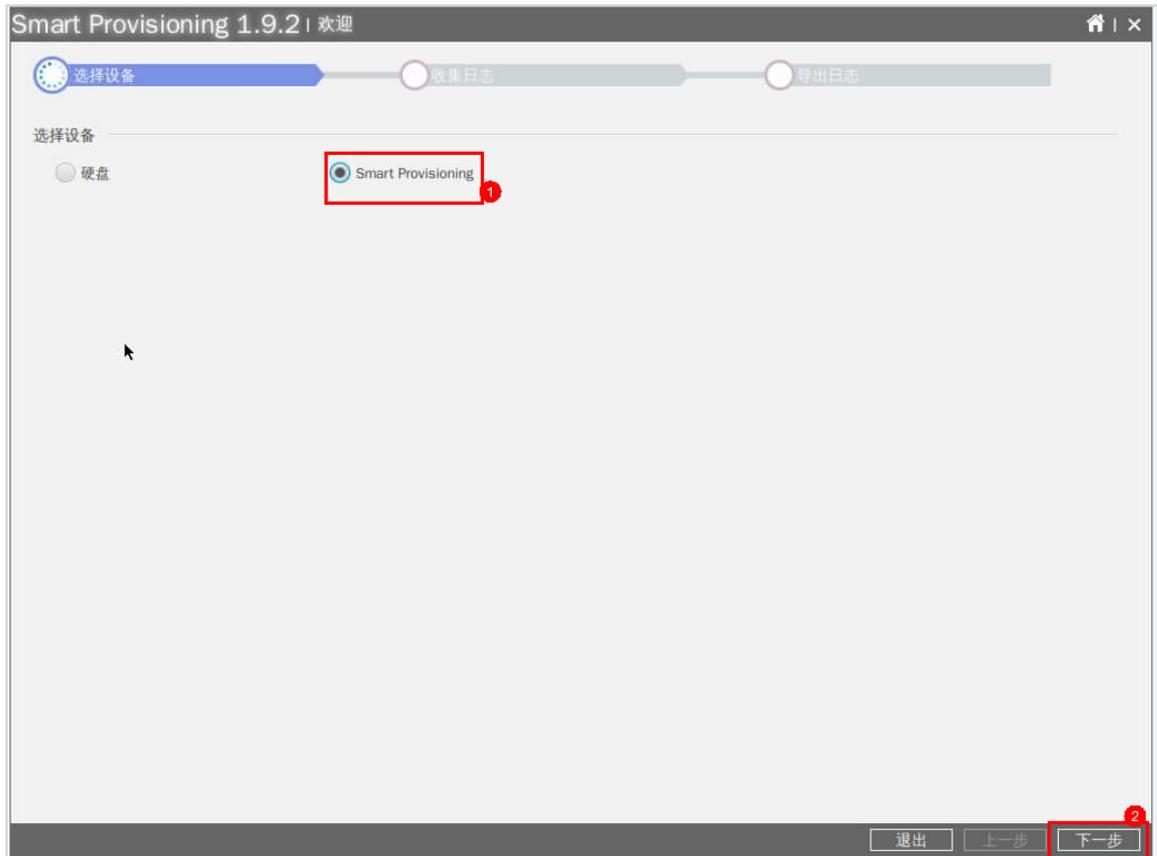


## 2.2.3 日志收集

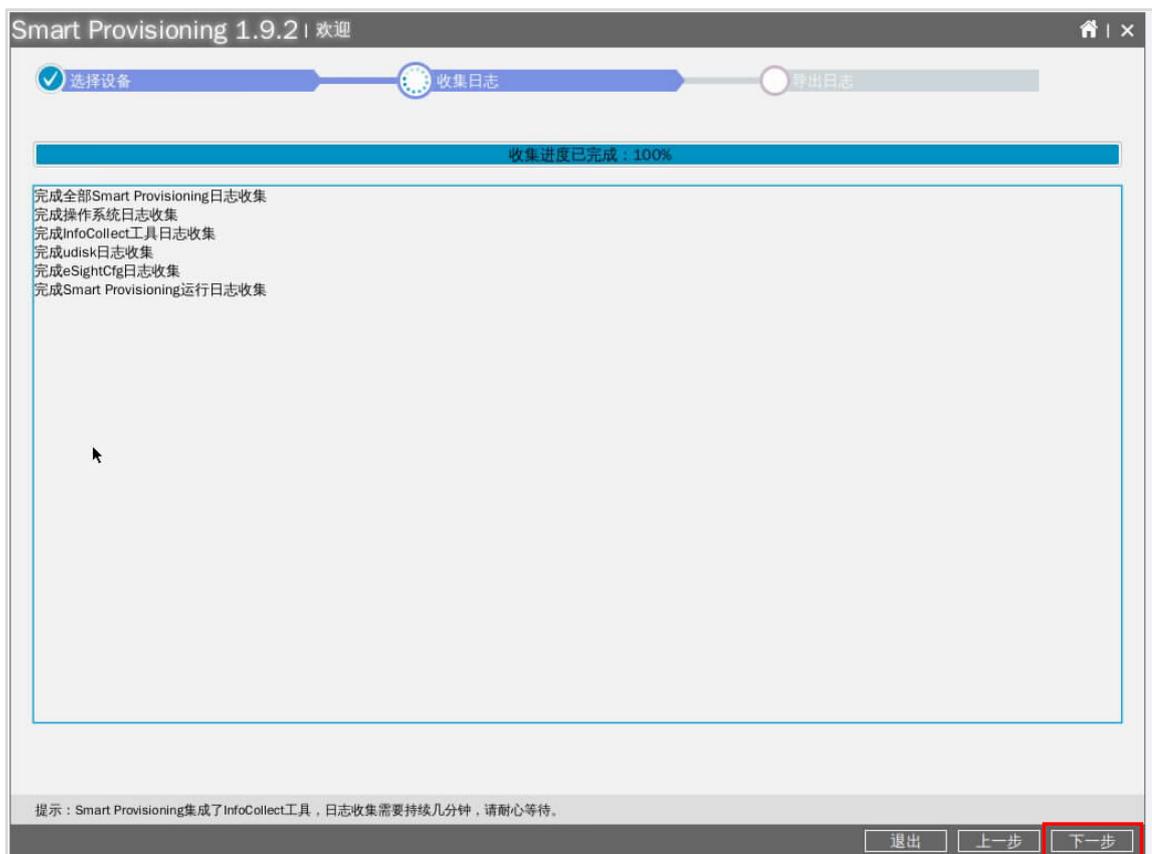
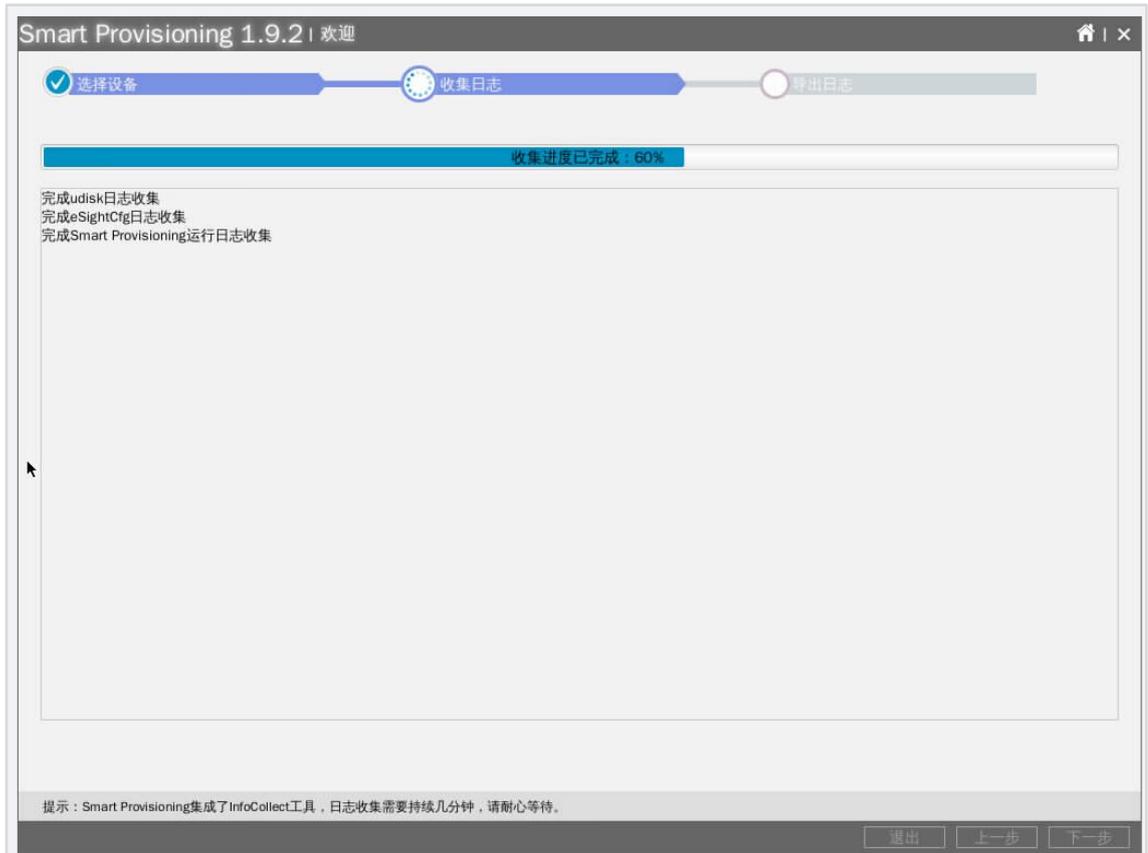
步骤 1 在 SP 工具箱界面，点击“日志收集”。



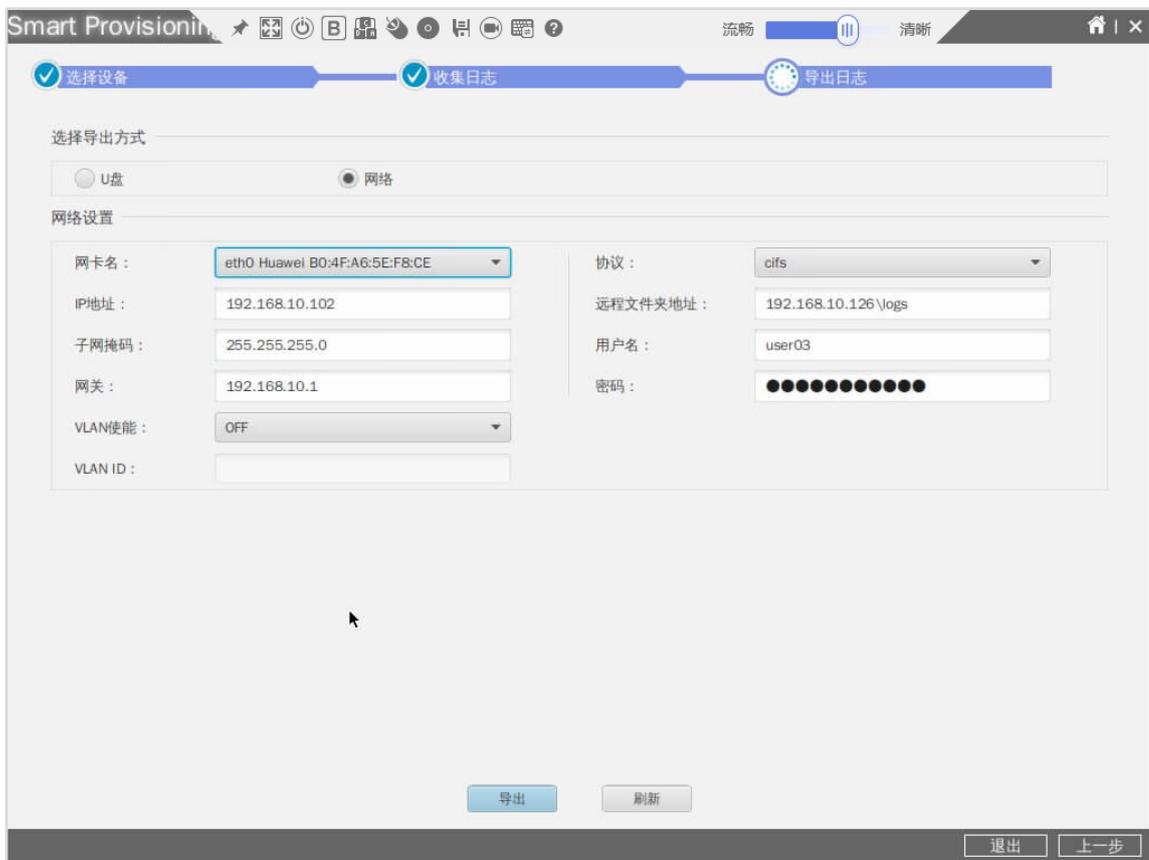
步骤 2 在“选择设备”界面，选择“Smart Provisioning”，点击“下一步”。



步骤 3 在“收集日志”界面，等待收集进度 100%后，点击“下一步”。



步骤 4 （可选）在“导出日志”界面，可选择“U 盘”或“网络”两种方式，可自行尝试把日志通过网络的形式导出到 PC 上。



### 【注】

#### 参数说明：

- 网卡名：可选，显示插入网线的业务侧网口对应的网卡名称。
- IP 地址：给本台服务器配置的操作系统 IP 地址。
- 子网掩码：给本台服务器配置的操作系统 IP 地址的子网掩码。
- 网关：为所选网卡配置的网关。
- 协议：Windows 系统选择“cifs”；Linux 系统选择“scp”。
- 远程文件夹地址：Windows 系统：远程服务器操作系统 IP 地址\共享的文件夹名称；Linux 系统：远程服务器操作系统 IP 地址\导出的文件夹名称。
- 用户名：Windows 系统：网络共享的用户名；Linux 系统：远程服务器操作系统用户名。
- 密码：Windows 系统：网络共享的密码；Linux 系统：远程服务器操作系统密码。
- VLAN 使能：设置 VLAN 使能状态。ON：使能 VLAN；OFF：禁用 VLAN。
- VLAN ID：范围 1 ~ 4094。

#### 步骤提示：

- 在本地 PC 上，打开 CMD 查看 IP 地址。

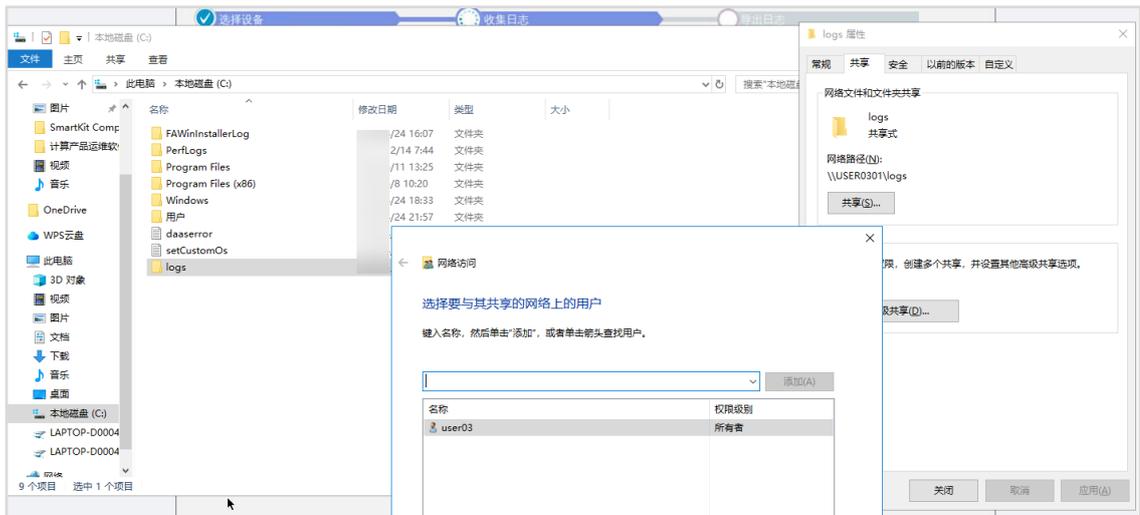
```
C:\Users\user03>ipconfig

Windows IP 配置

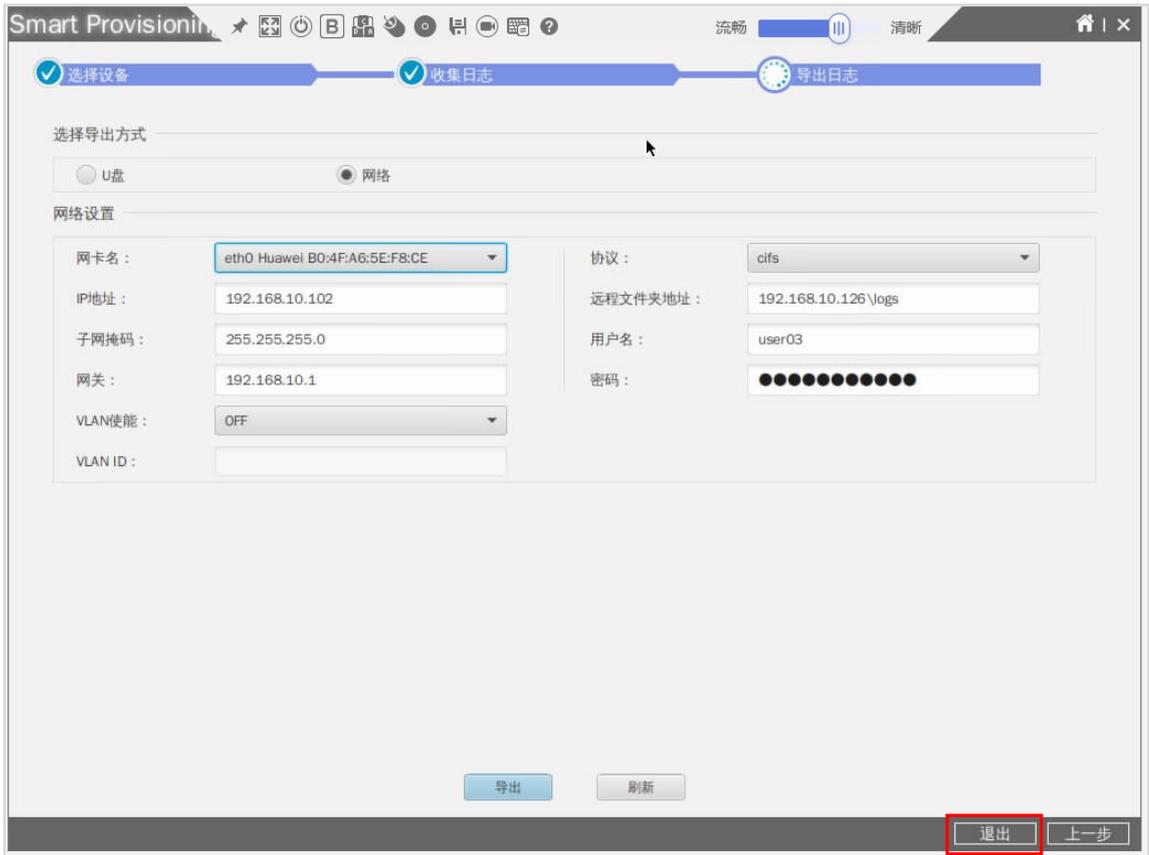
以太网适配器 以太网:

    连接特定的 DNS 后缀 . . . . . : vdesktop.huawei.com
    本地连接 IPv6 地址. . . . . : fe80::90b4:bb04:9b6f:3360%11
    IPv4 地址 . . . . . : 192.168.10.126
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : 192.168.10.1
```

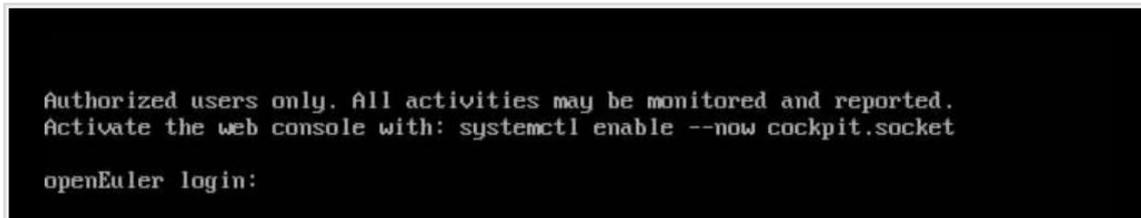
- 在本地 PC 的 C 盘，新建 logs 文件夹，并配置文件共享。



步骤 5 点击“退出”，系统会弹出确认窗口，点击“确定”，结束日志收集任务。



步骤 6 关闭 SP，弹出确认框，点击“确定”，服务器会重新启动，无需人工干预进入操作系统界面。



## 2.3 SmartKit 安装与使用

确认本地 PC 里 SmartKit 是否已经安装并初始化，如果已经安装并初始化，从 2.3.1 章节开始实验任务；如果已经安装未初始化，从 2.3.2 章节开始实验任务；如果未安装，从 2.3.3 章节开始实验任务。

### 2.3.1 安装 SmartKit（可选）

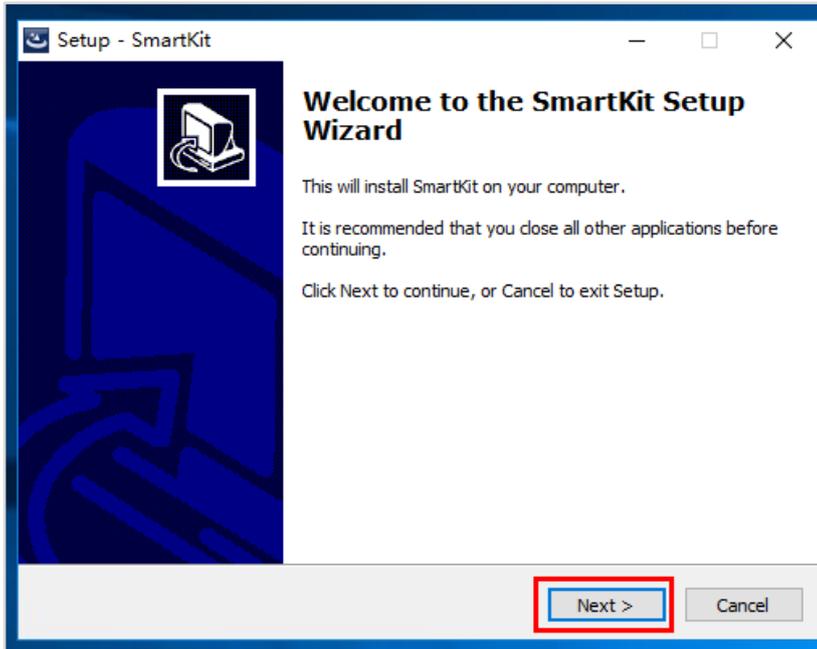
步骤 1 获取 SmartKit 安装包。



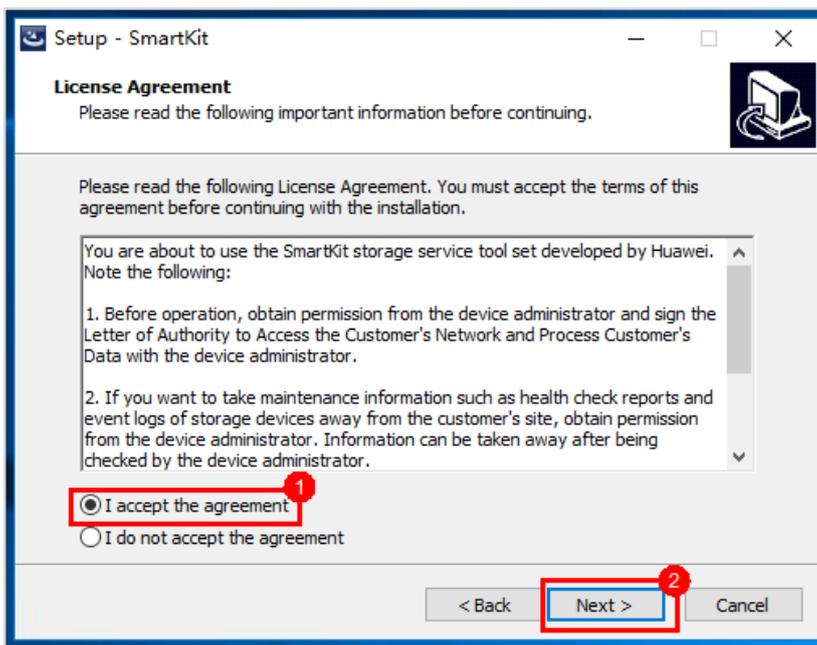
步骤 2 鼠标双击安装 SmartKit 软件包，系统会弹出“用户帐户控制”弹窗，点击“是”。



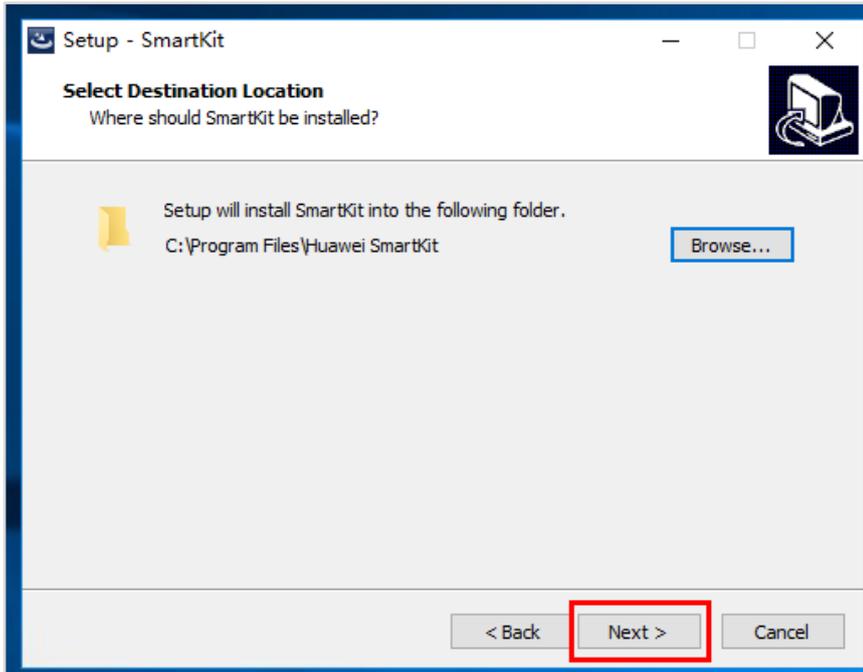
步骤 3 进入 SmartKit 安装界面，点击“Next”。



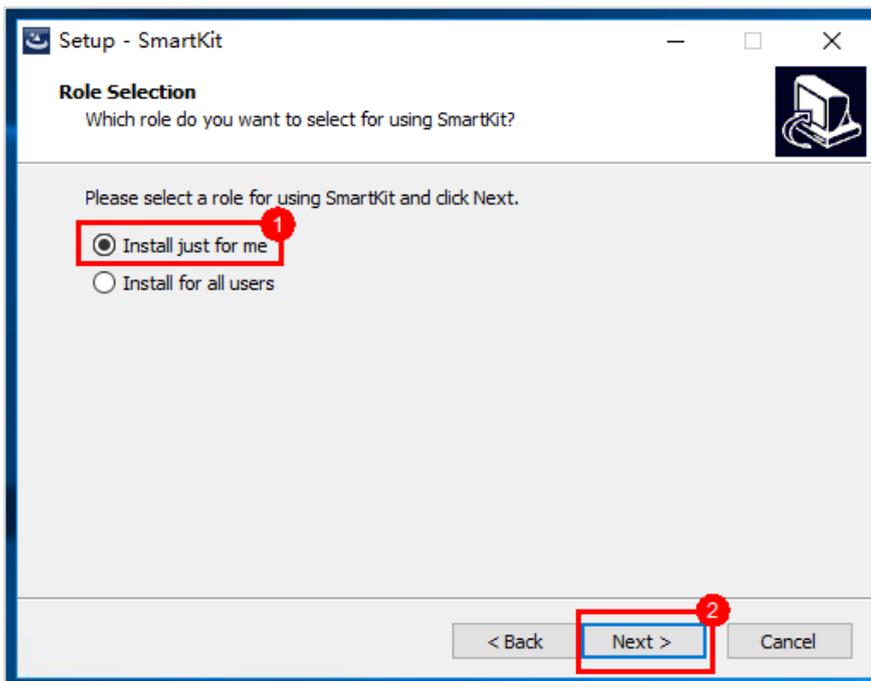
步骤 4 在“License Agreement”界面，选择“I accept the agreement”，点击“Next”。



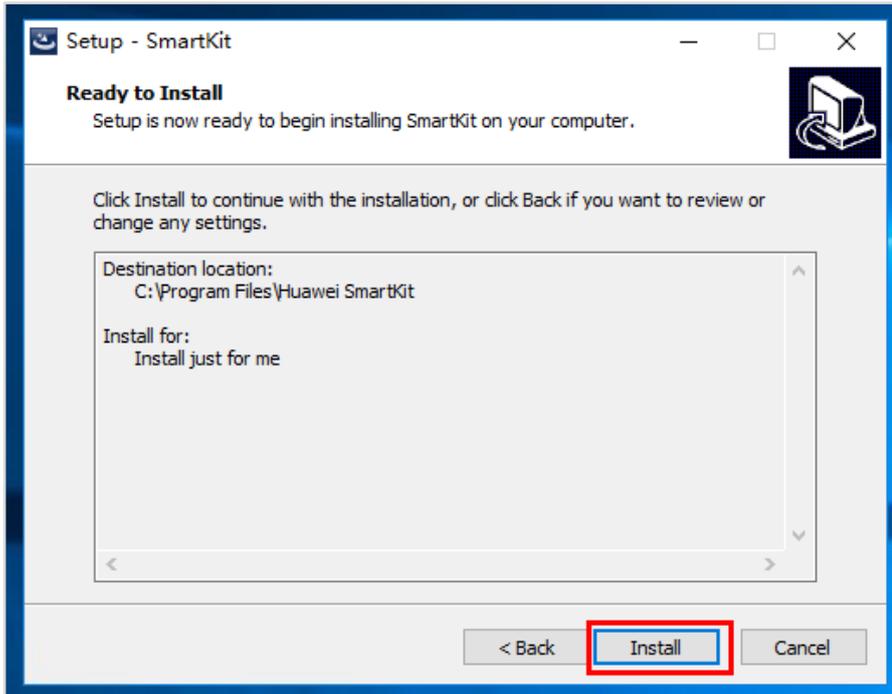
步骤 5 在“Select Destination Location”，保持软件默认安装路径不修改，点击“Next”。



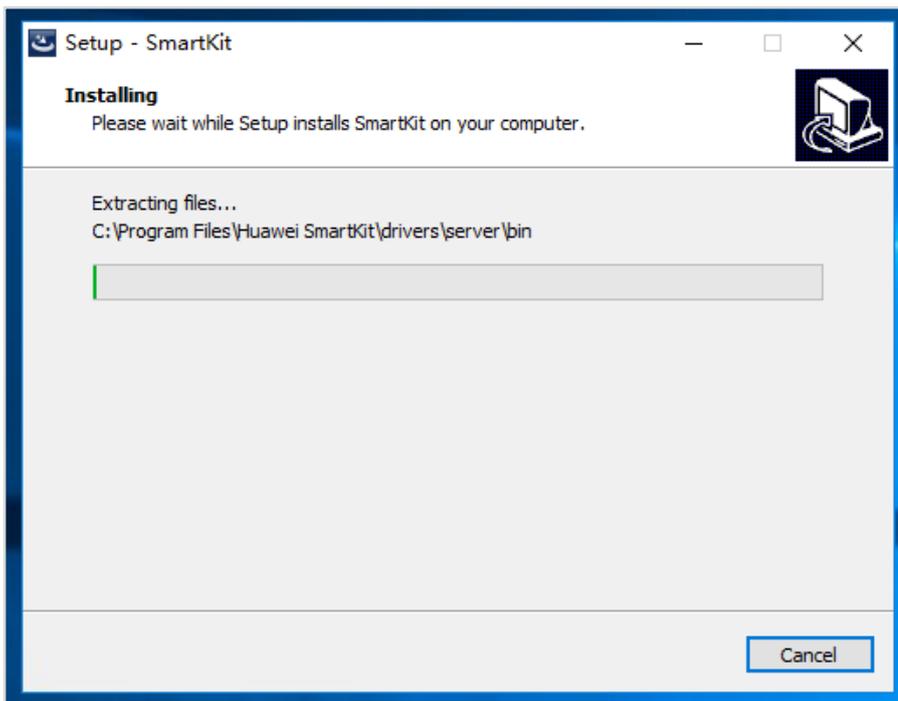
步骤 6 在“Role Selection”界面，选择“Install just for me”，点击“Next”。

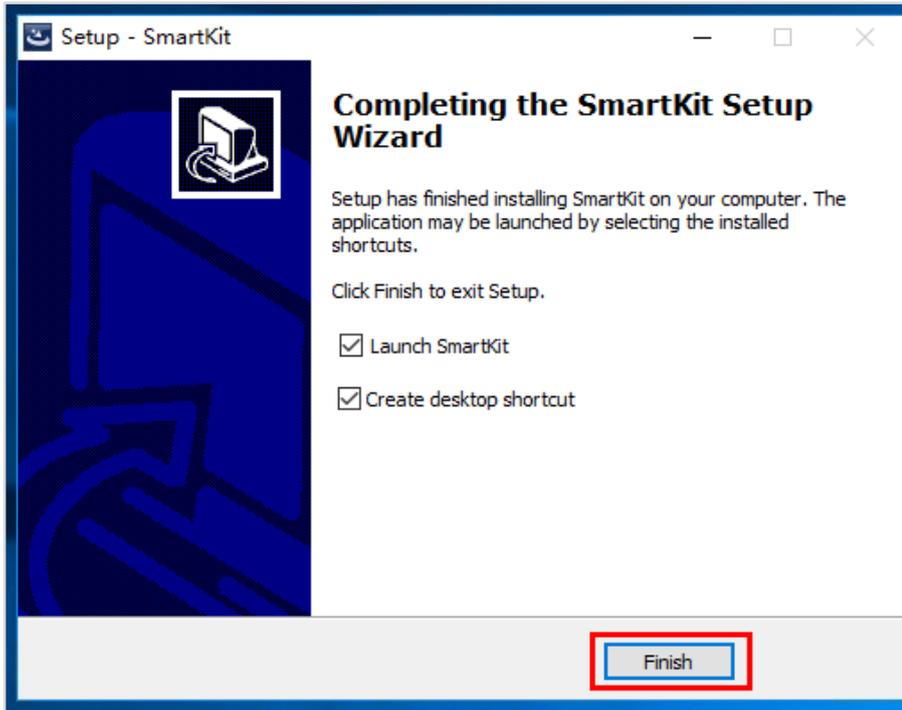


步骤 7 在“Ready to Install”界面，点击“Install”。



步骤 8 等待 2~5 分钟左右，SmartKit 安装完成，点击“Finish”。





## 2.3.2 初始化 SmartKit (可选)

步骤 1 SmartKit 安装完成后，勾选了“Launch SmartKit”，会自动运行软件，等待片刻后，会出现“登录认证”界面，点击“暂不认证”。



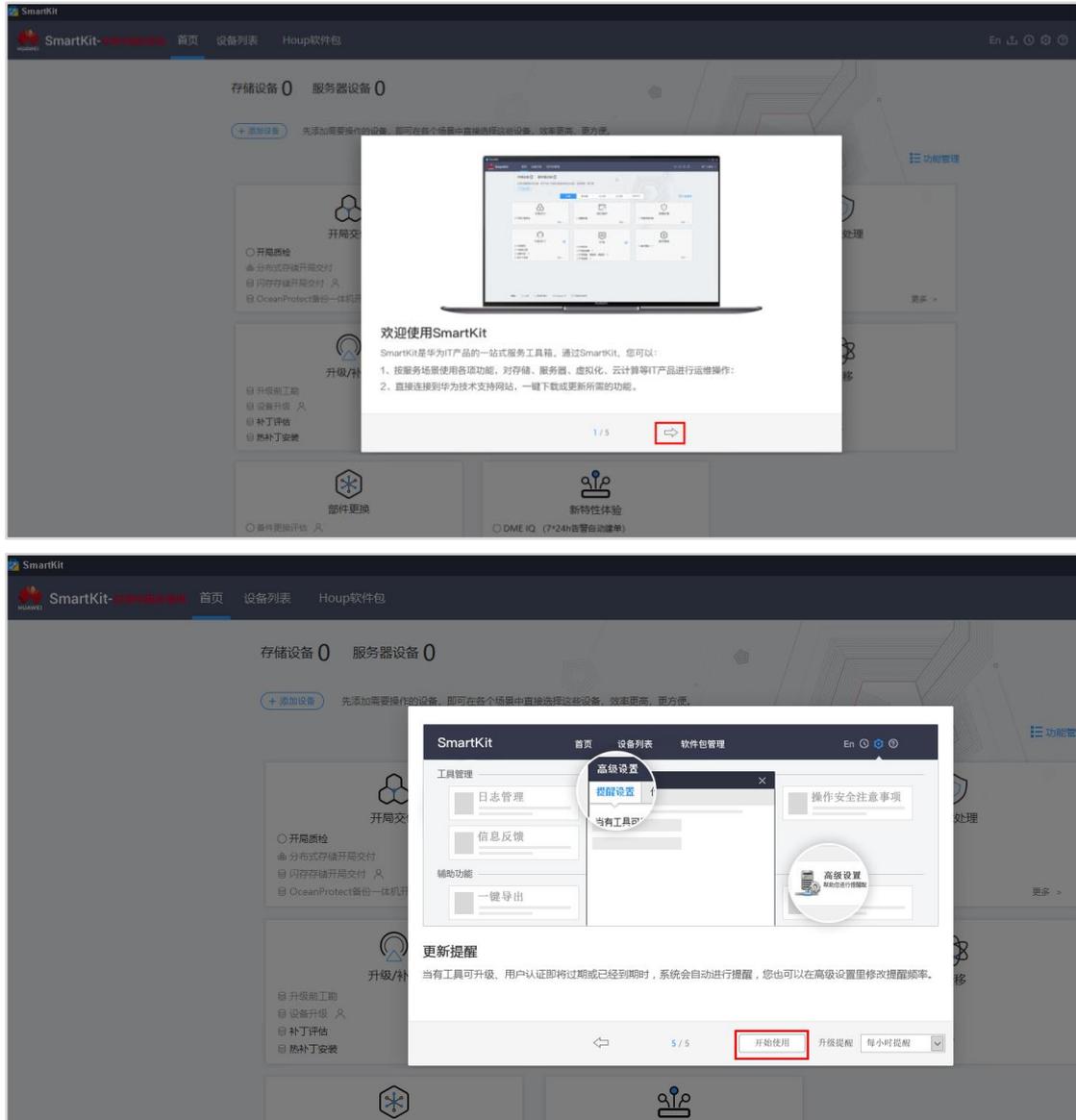
步骤 2 系统会弹出“用户账户控制”，点击“是”。



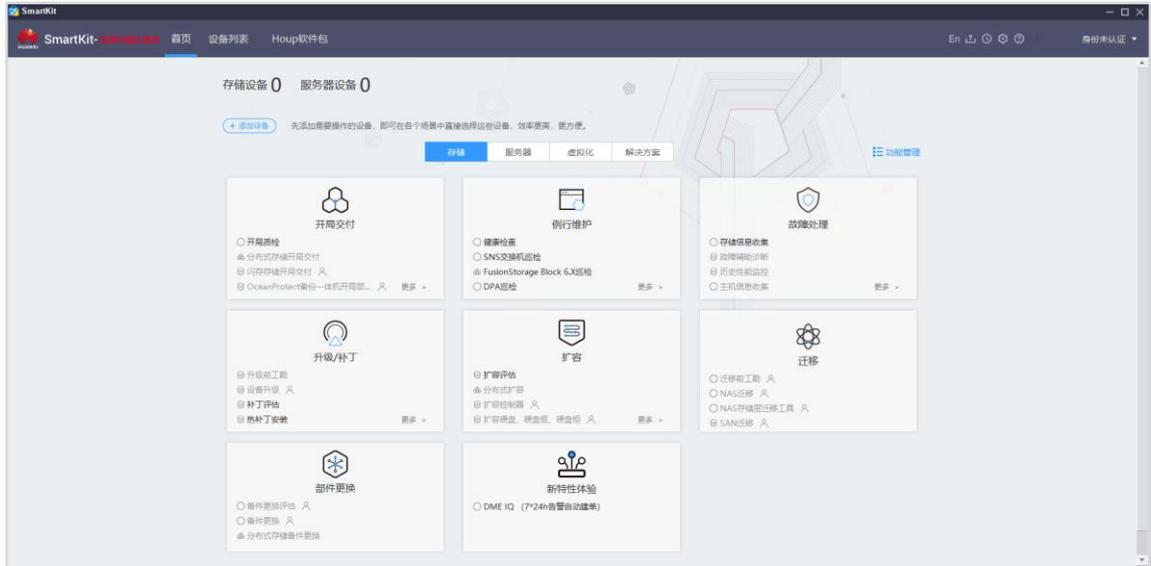
步骤 3 在“操作安全注意事项”界面，按下图进行操作，点击“继续”。



步骤 4 进入 SmartKit 主页，软件会弹出使用指导窗口，按下图进行操作，点击“开始使用”。



步骤 5 SmartKit 初始化完成，正式进入首页。



### 2.3.3 添加服务器设备

步骤 1 在 SmartKit 首页，点击“添加设备”，进入设备列表界面。



步骤 2 左侧选择“服务器 > iBMC”，点击右上角“增加”。



步骤 3 按下图进行参数配置，点击“确定”。

【注】只需配置 iBMC 的 IP、用户名、密码即可，其余参数保持默认。

增加设备

\* 起始IP: 192.168.10.15 \* 结束IP: 192.168.10.15

\* 用户名: Administrator \* 密码: \*\*\*\*\*

SNMPv3加密密码: (如果不填则使用上面的用户密码, 如果SNMPv3加密密码和用户密码不一致, 则SNMPv3加密...)

高级设置

\* SNMP端口: 161 \* IPMI端口: 623

\* SSH / SFTP端口: 22 \* VMM端口: 8208

\* HTTPS端口: 443 是否支持帐号解耦:

通道加密算法:  开启兼容性算法 是否为代理场景:

OS/IES选项

root 密码: (对于OS类型设备, 非root用户如无root权限必须输入root密码)

交换模块选项 (E9000 健康检查、日志收集、升级涉及到交换模块的场景需要)

计算节点信息 (E9000 MM920/MM921或E9000H SMM 涉及到计算节点的场景需要)

确定 取消

步骤 4 等待 1 分钟左右, SmartKit 完成服务器设备添加, 点击“确定”。

增加设备

执行中... 请稍候。

10%

总数: 1 选中数: 0

对象	状态	进度	处理建议
192.168.10.15	执行中	10%	

取消

增加设备

添加完成。iBMC: 1, HMM: 0, OS: 0, CMC: 0, IES: 0, E9000H: 0, SWITCH: 0;  
成功: 1, 失败: 0。

100%

总数: 1 选中数: 0

对象	状态	进度	处理建议
192.168.10.15	完成	100%	

确定

步骤 5 添加的服务器已存在设备列表中。

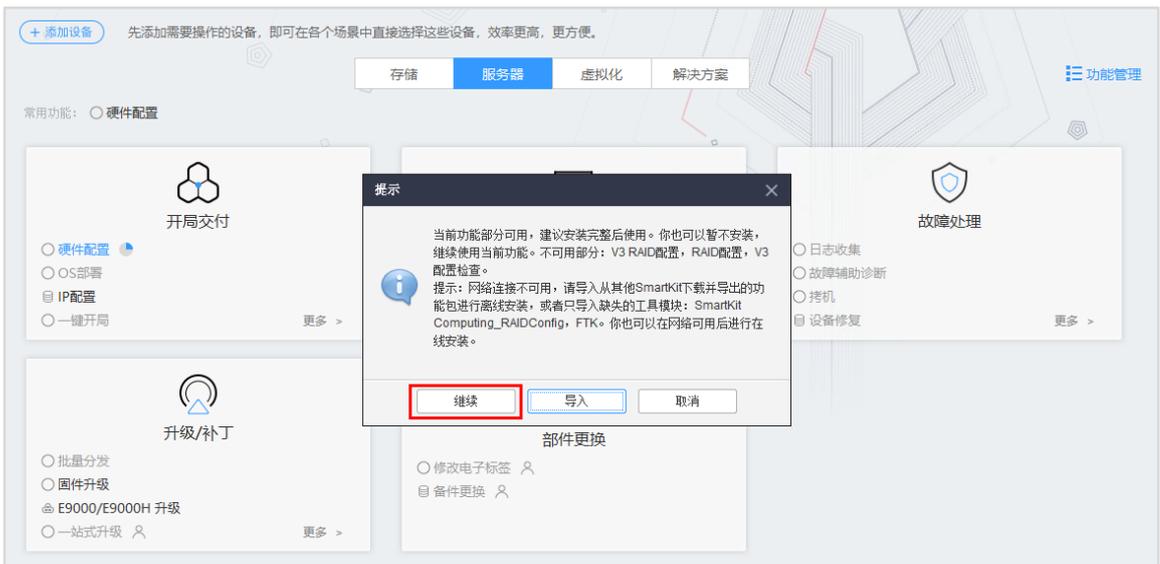


## 2.3.4 配置服务器 iBMC

步骤 1 在 SmartKit “首页”，点击“服务器 > 开局交付 > 硬件配置”。



步骤 2 弹出提示框，直接点击“继续”。

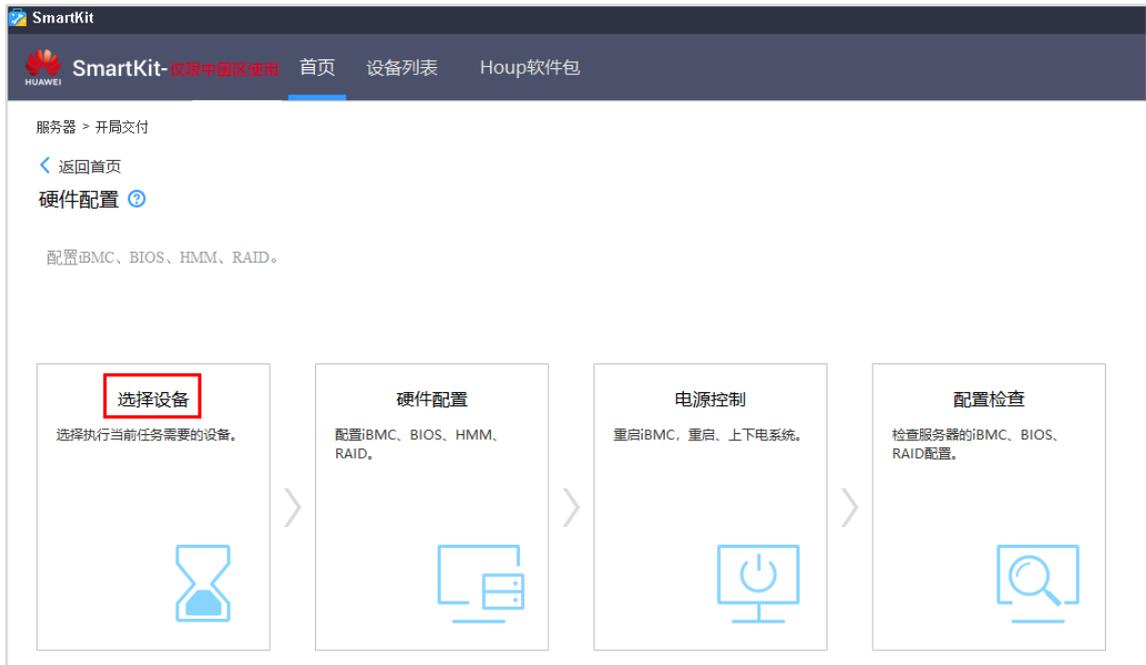


【说明】SmartKit 工具安装完毕后，默认会安装一些基础功能，部分功能需要额外安装软件包，可以通过联网下载，或者下载离线功能包导入两种形式添加功能，由于实验平台不连接互

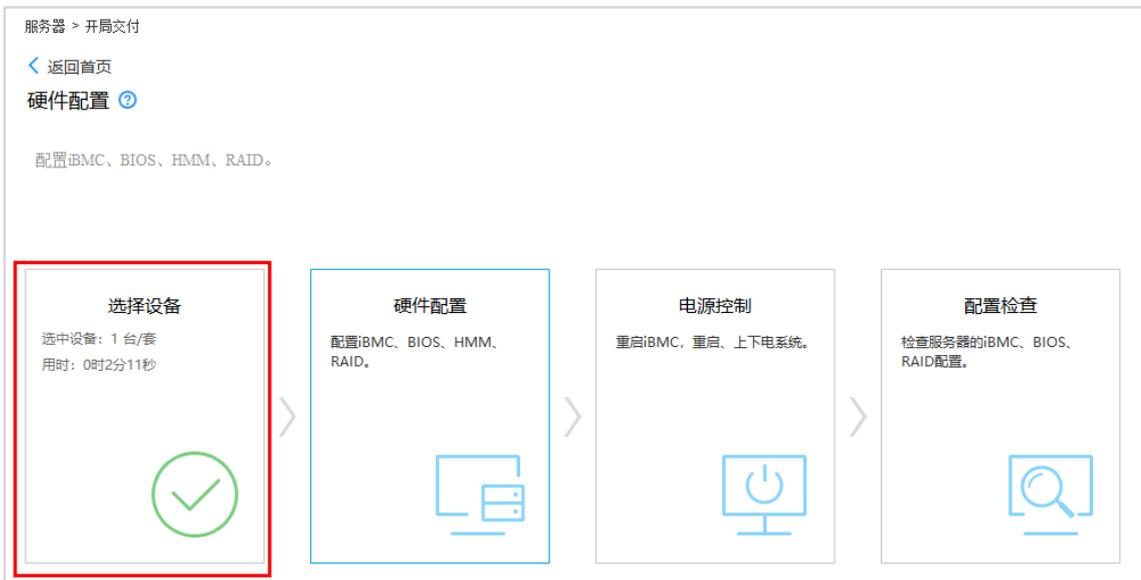
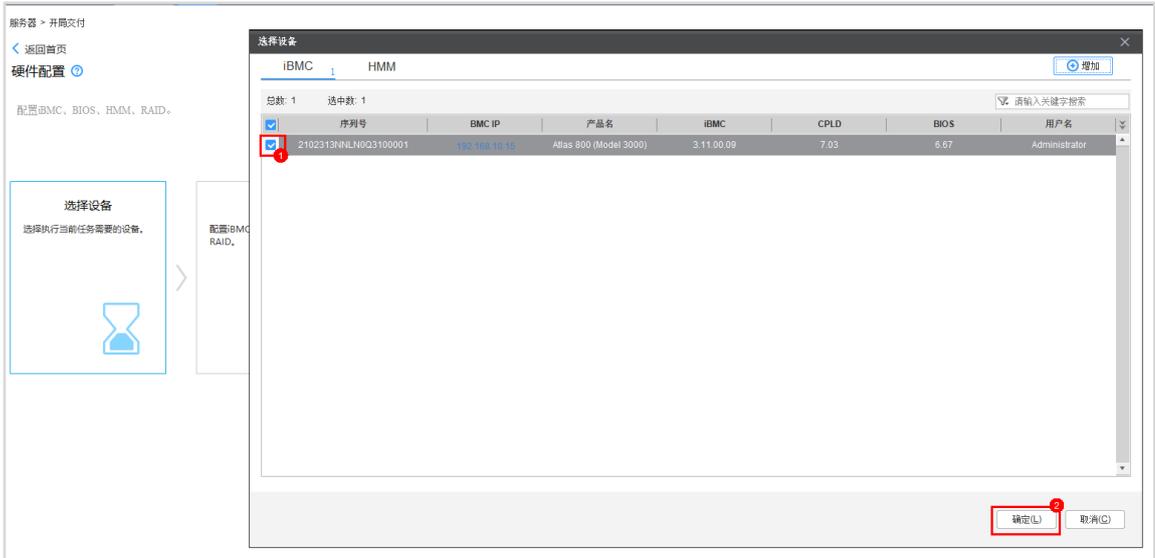
联网，只能通过下载离线包导入的形式，但配置 iBMC 功能属于 SmartKit 基础功能，无需导入离线功能包，这里直接忽略提示框即可。



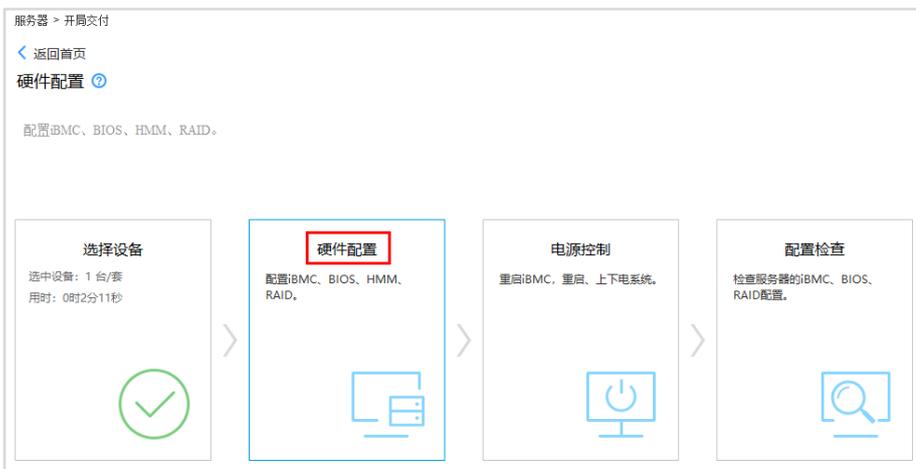
步骤 3 在“硬件配置”界面，点击“选择设备”。

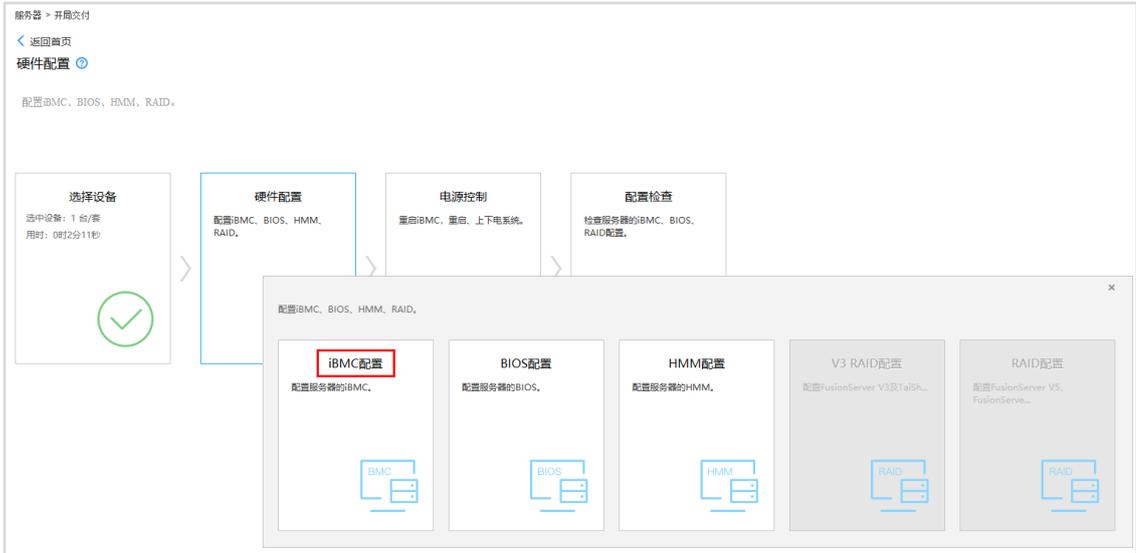


步骤 4 在“选择设备”弹出框，选择前面步骤添加的服务器，点击“确定”。



步骤 5 点击“硬件配置”，在弹出框，点击“iBMC 配置”。





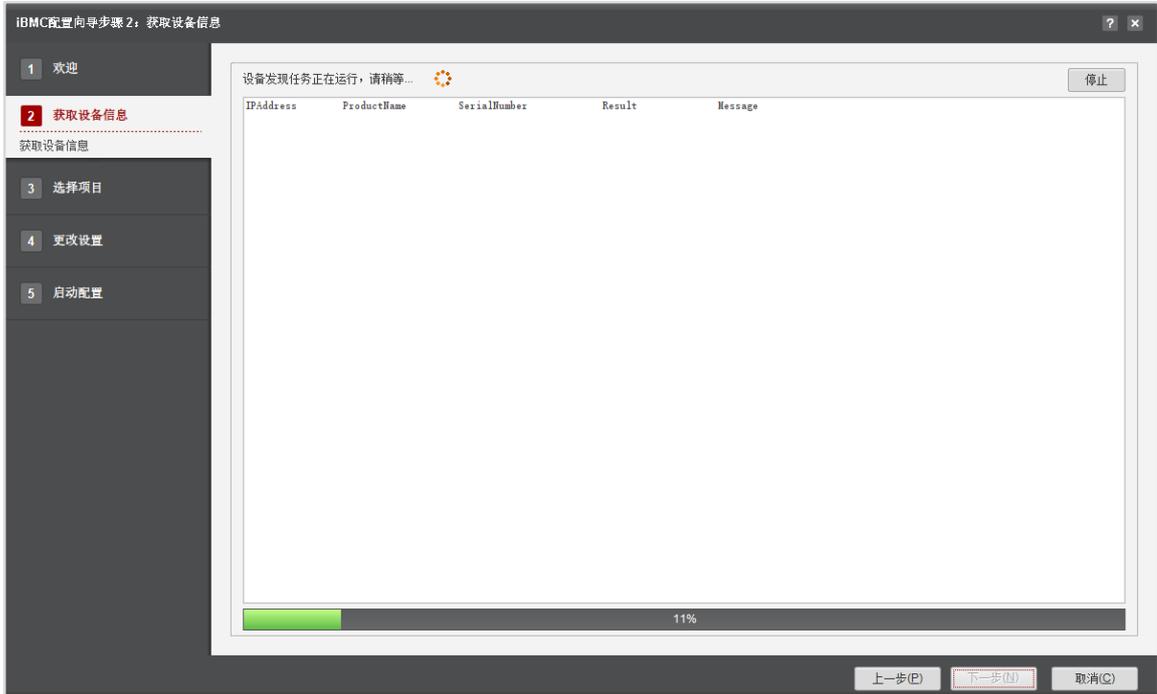
步骤 6 在“iBMC 配置向导步骤 1：欢迎”界面，点击“下一步”。

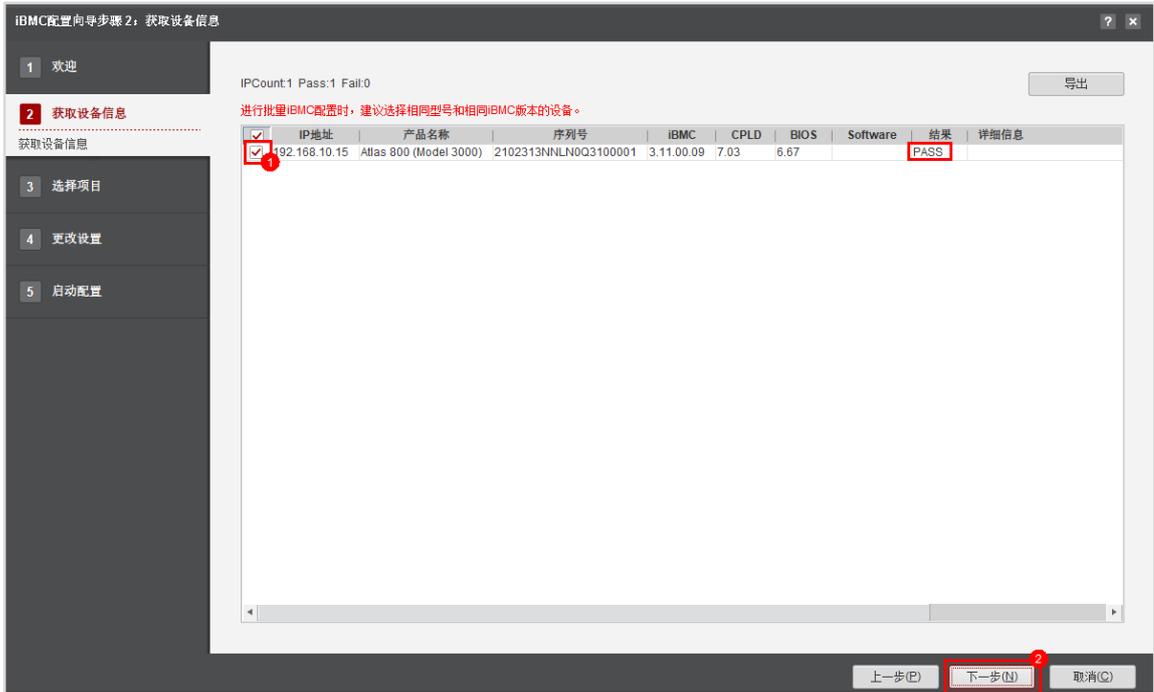


步骤 7 在“iBMC 配置向导步骤 2：获取设备信息”界面，勾选服务器，点击“下一步”



步骤 8 SmartKit 进行设备发现任务，等待片刻，看到结果是“PASS”，勾选服务器，点击“下一步”。

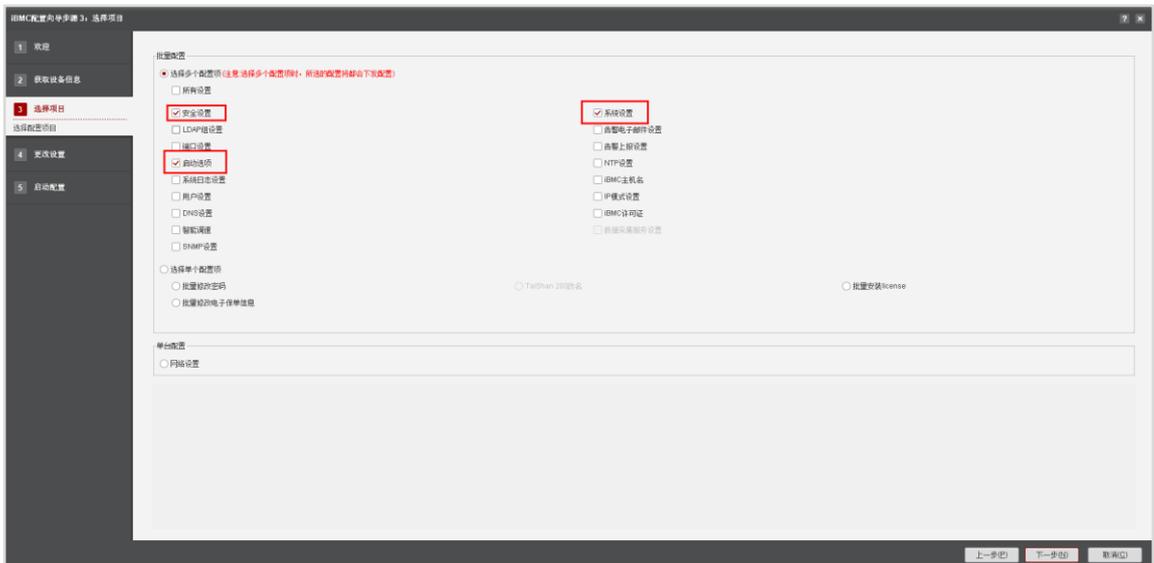




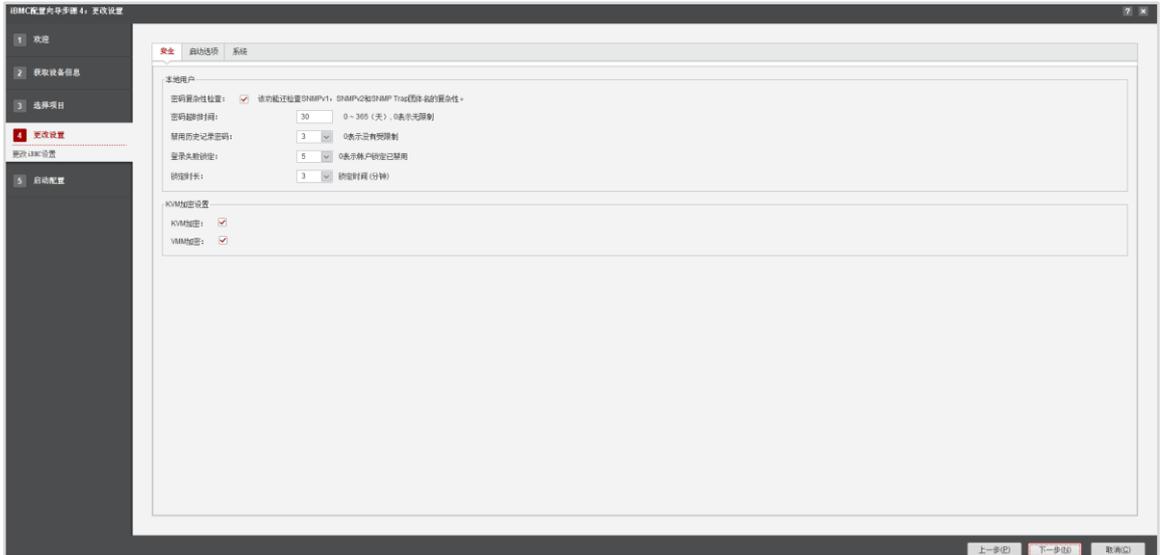
步骤 9 在“iBMC 配置向导步骤 3: 选择项目”界面，按下图勾选配置项，点击“下一步”。

配置项如下：

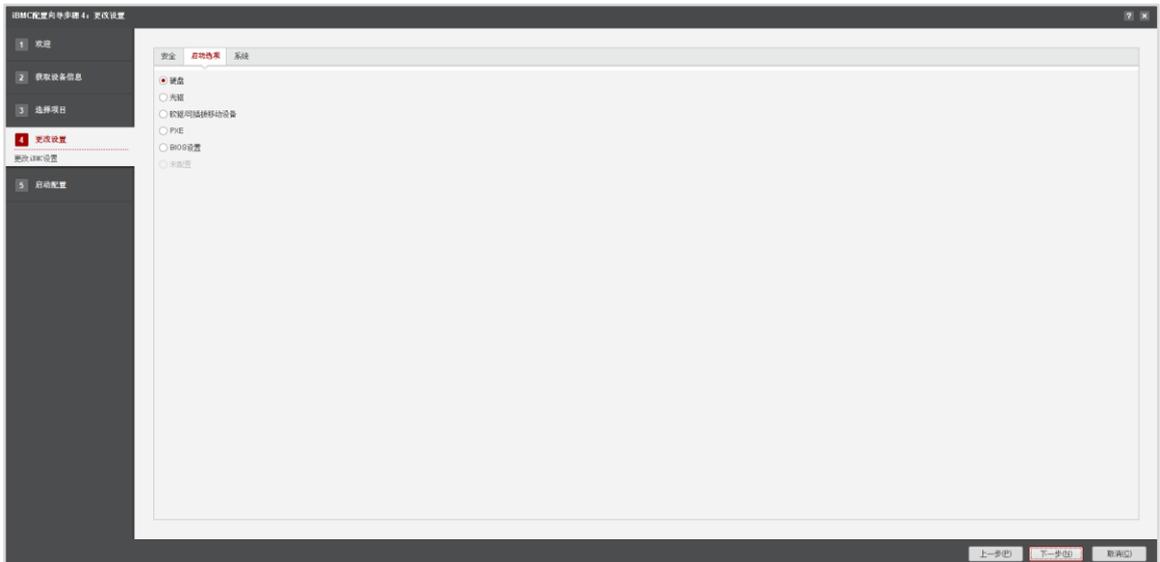
- 安全设置
- 系统设置
- 启动选项



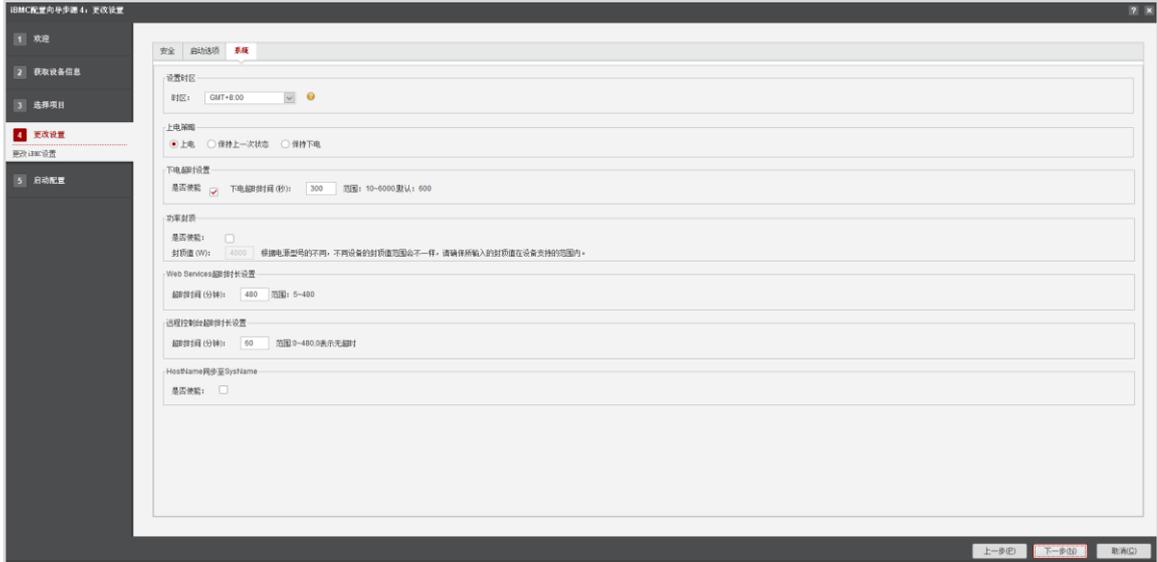
步骤 10 在“iBMC 配置向导步骤 4: 更改设置”界面，按下图修改安全配置。



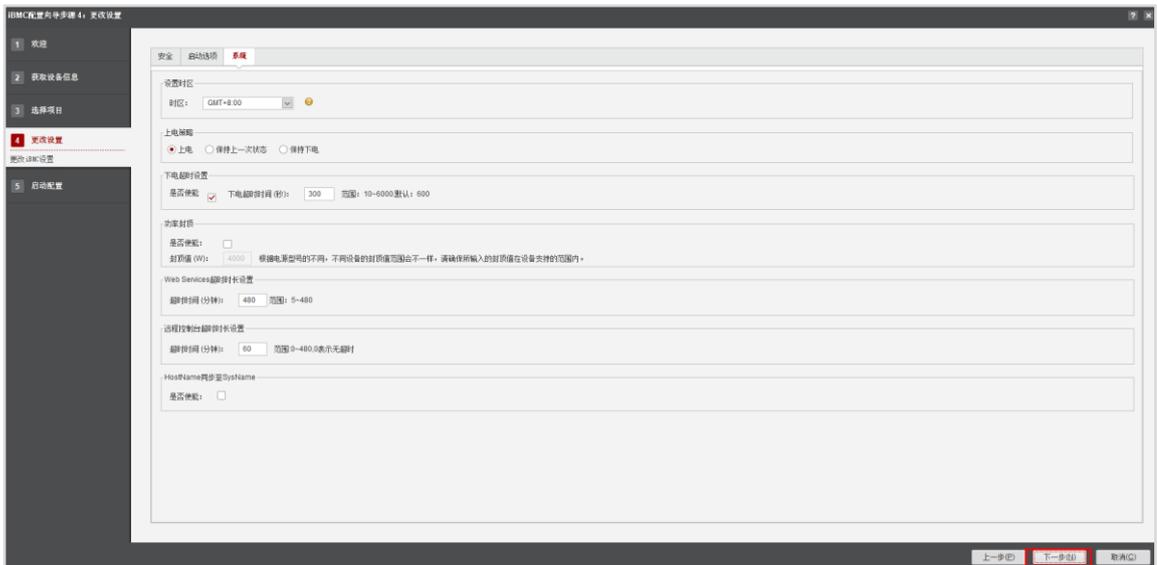
步骤 11 在“iBMC 配置向导步骤 4：更改设置”界面，按下图修改启动选项。



步骤 12 在“iBMC 配置向导步骤 4：更改设置”界面，按下图修改系统配置。

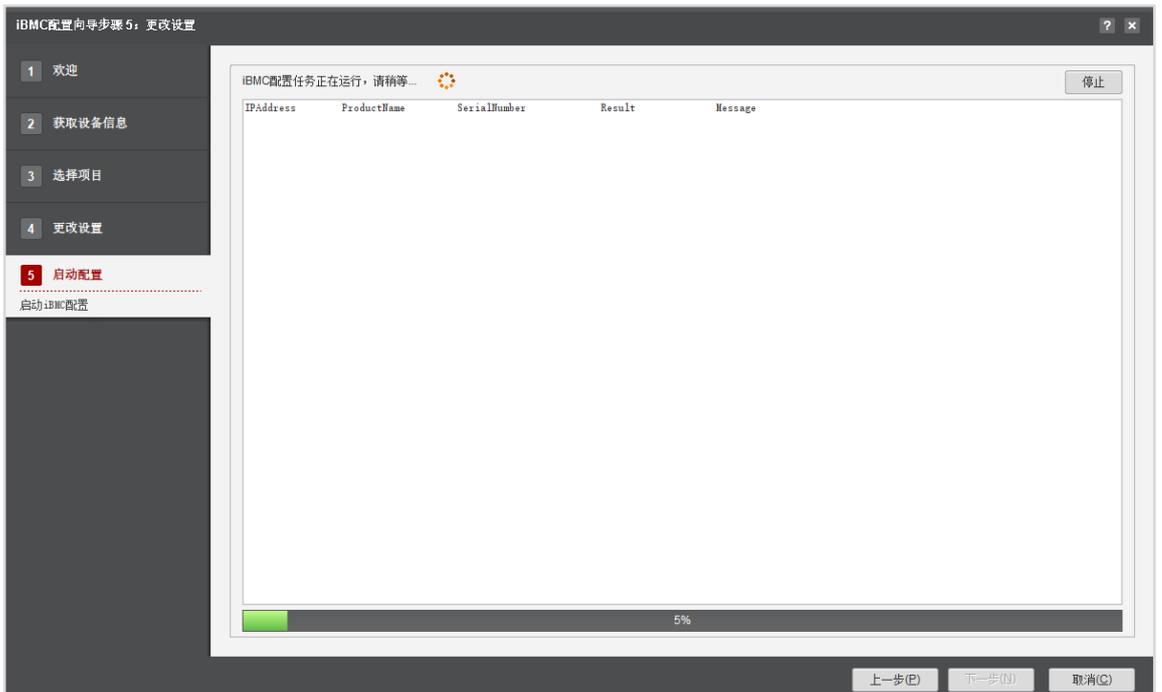


步骤 13 所有都配置完毕后, 点击“下一步”, 弹出危险弹窗, 输入“yes”按下回车, 并点击“确定”。

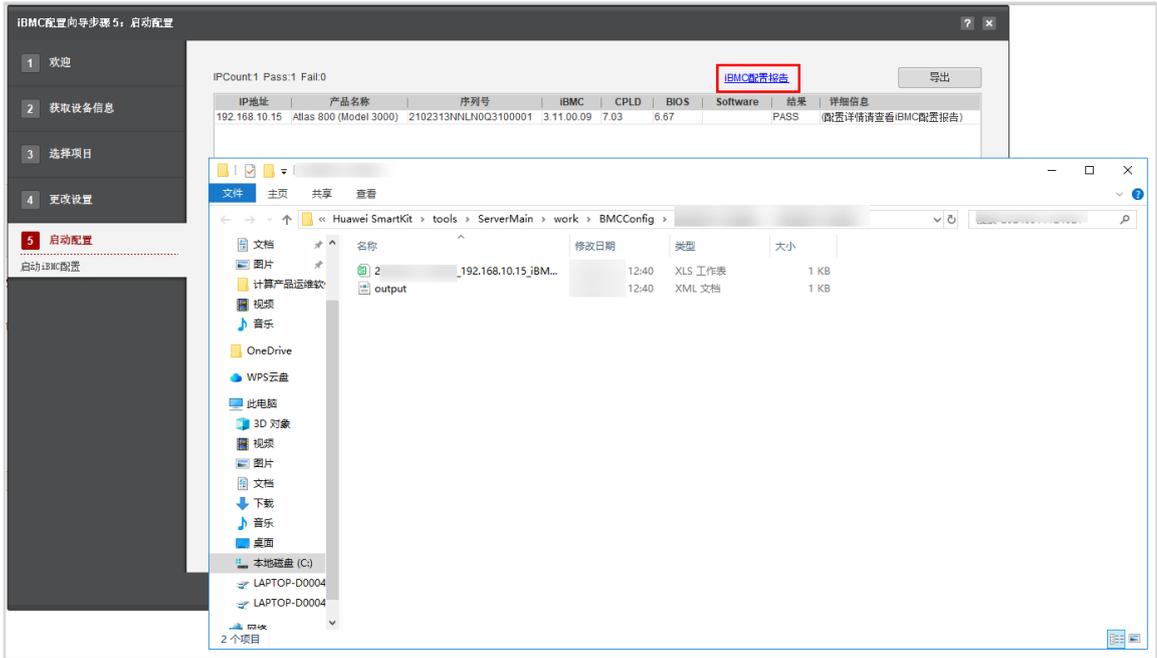




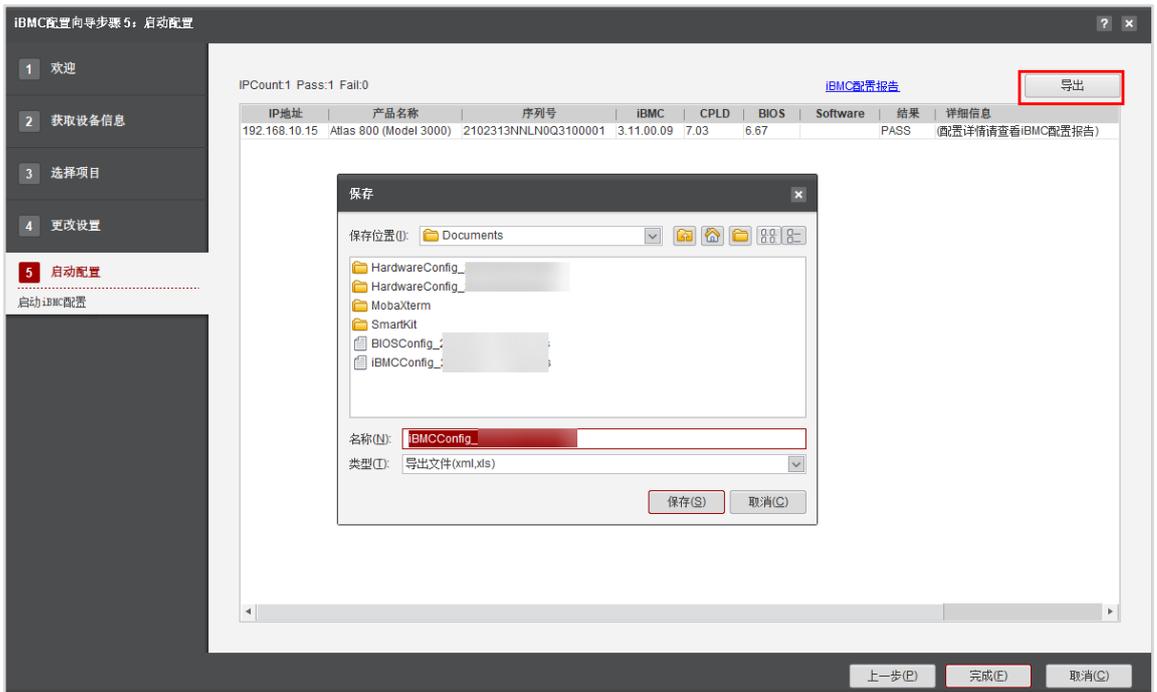
步骤 14 等待片刻，iBMC 配置任务为 PASS。

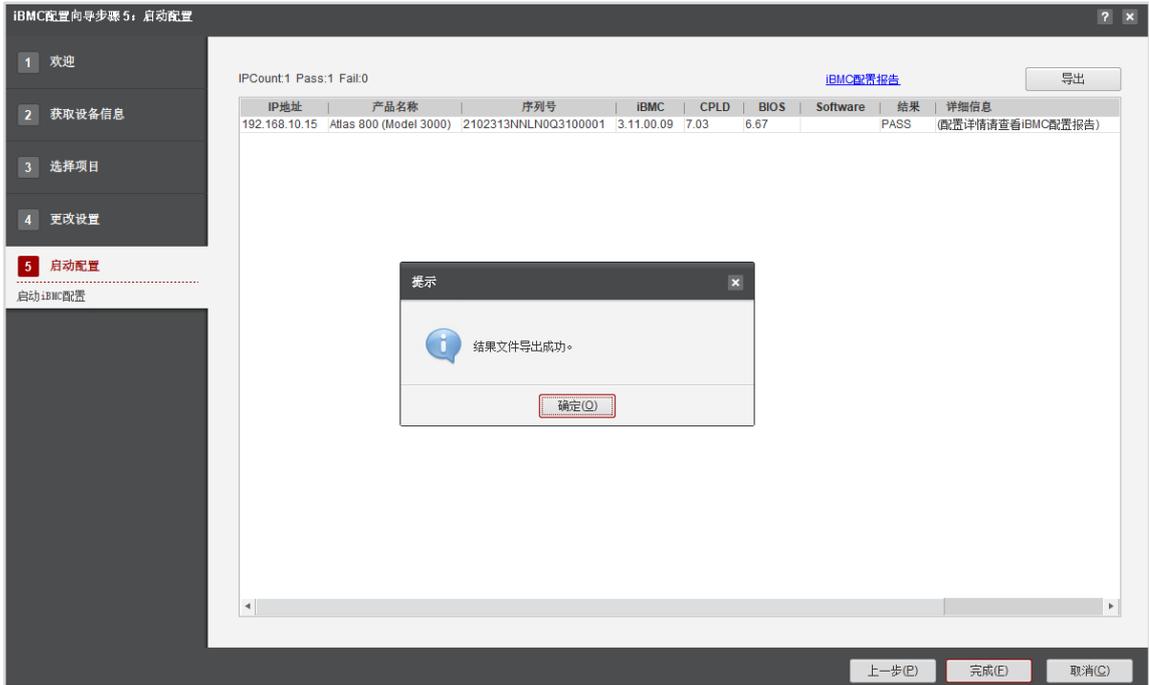


步骤 15 点击右上角“iBMC 配置报告”，可查看详细配置报告。

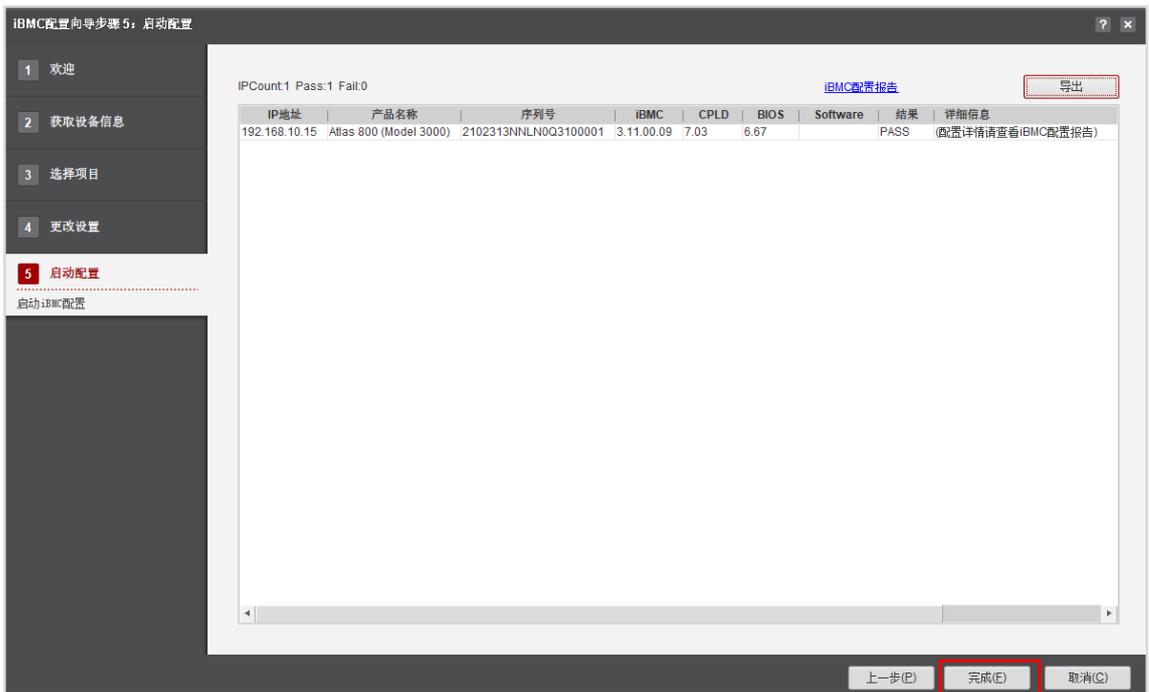


步骤 16 点击右上角“导出”，可把配置报告导出保存在本地。





步骤 17 点击右下角“完成”，结束 iBMC 配置任务，系统会自动回到“硬件配置”界面。





## 2.3.5 配置服务器 BIOS

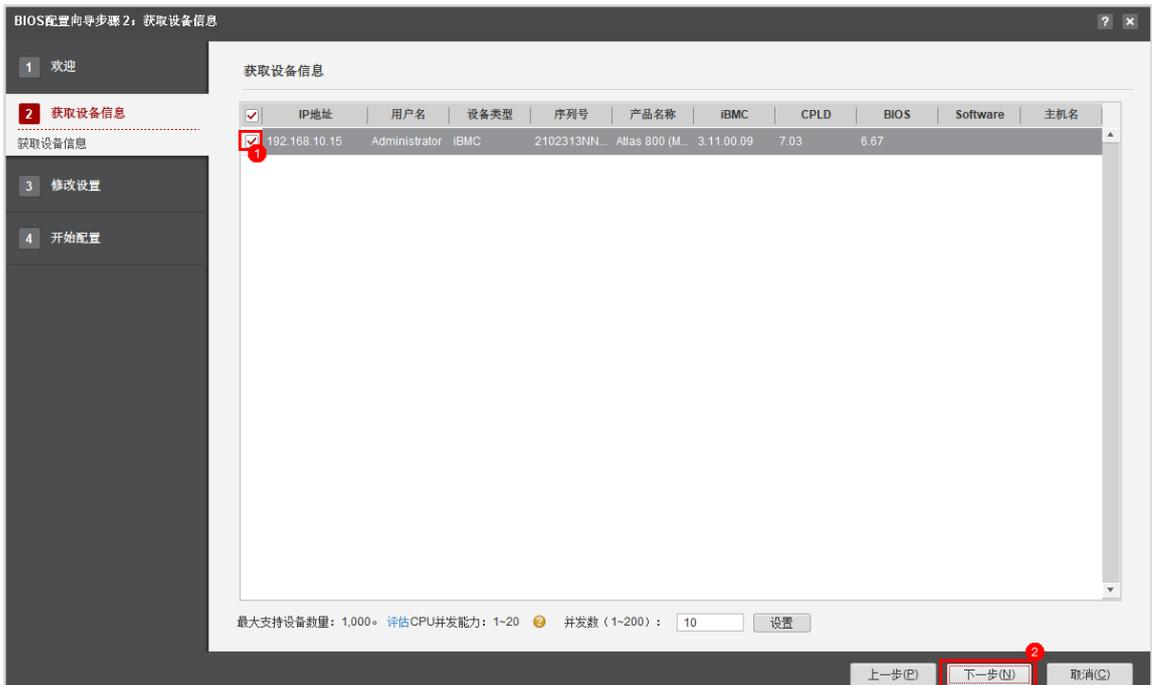
步骤 1 继续在“硬件配置”界面，点击“BIOS配置”。



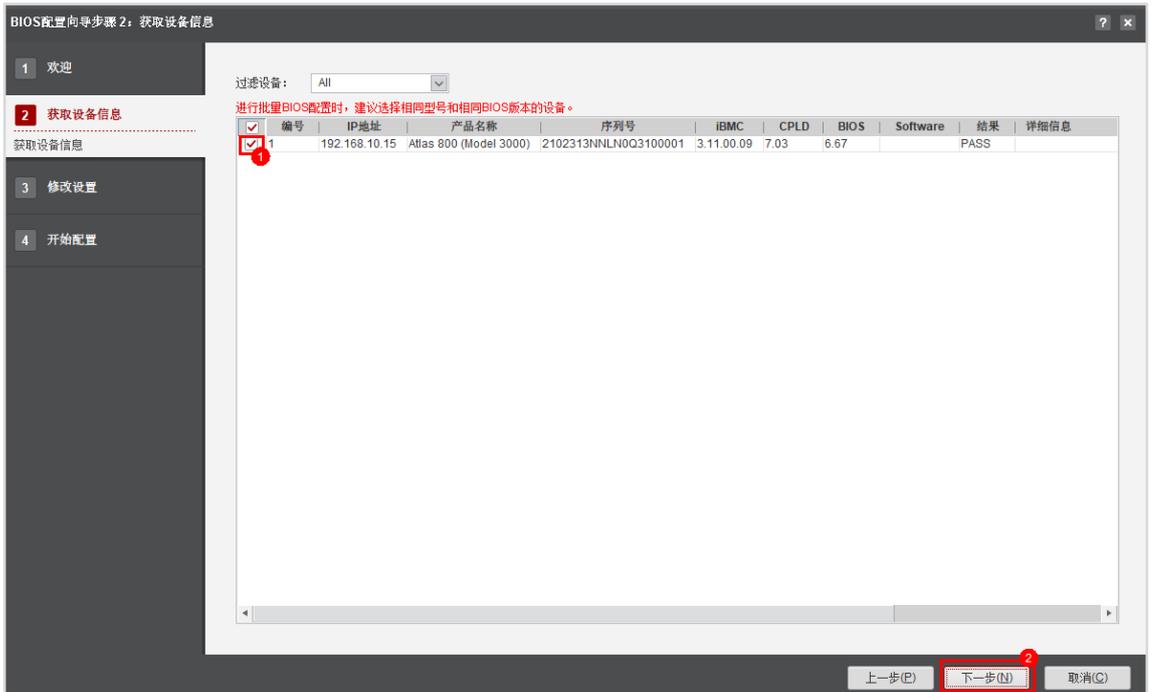
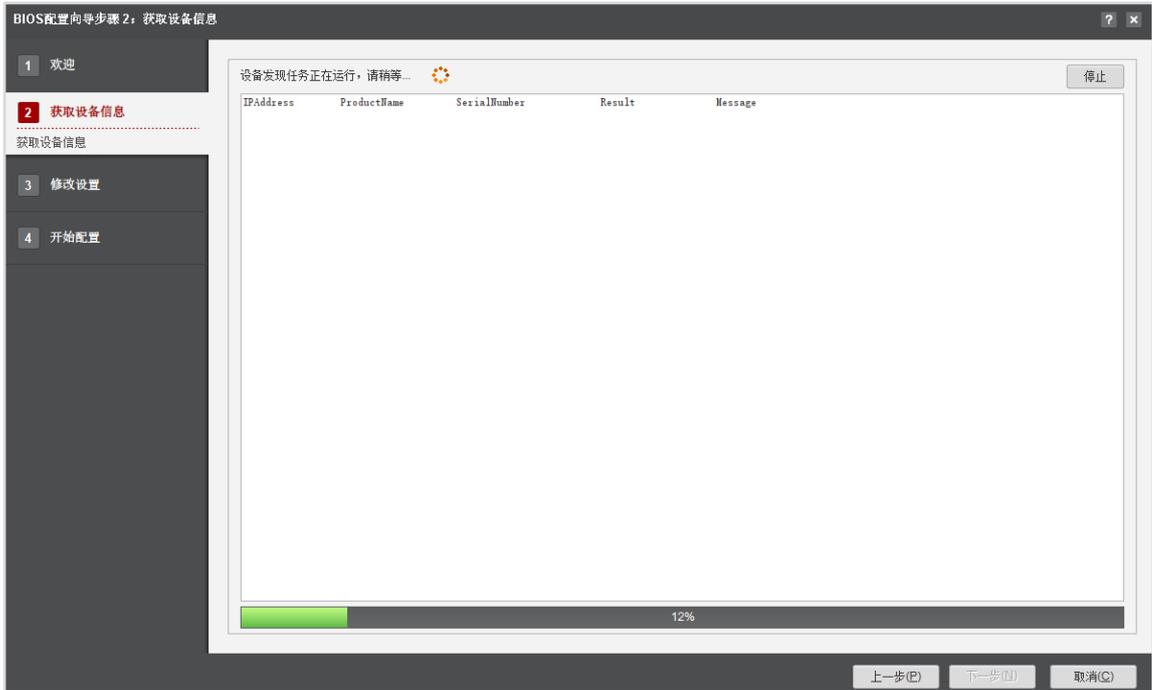
步骤 2 在“BIOS配置向导步骤 1: 欢迎”界面，直接点击下一步。



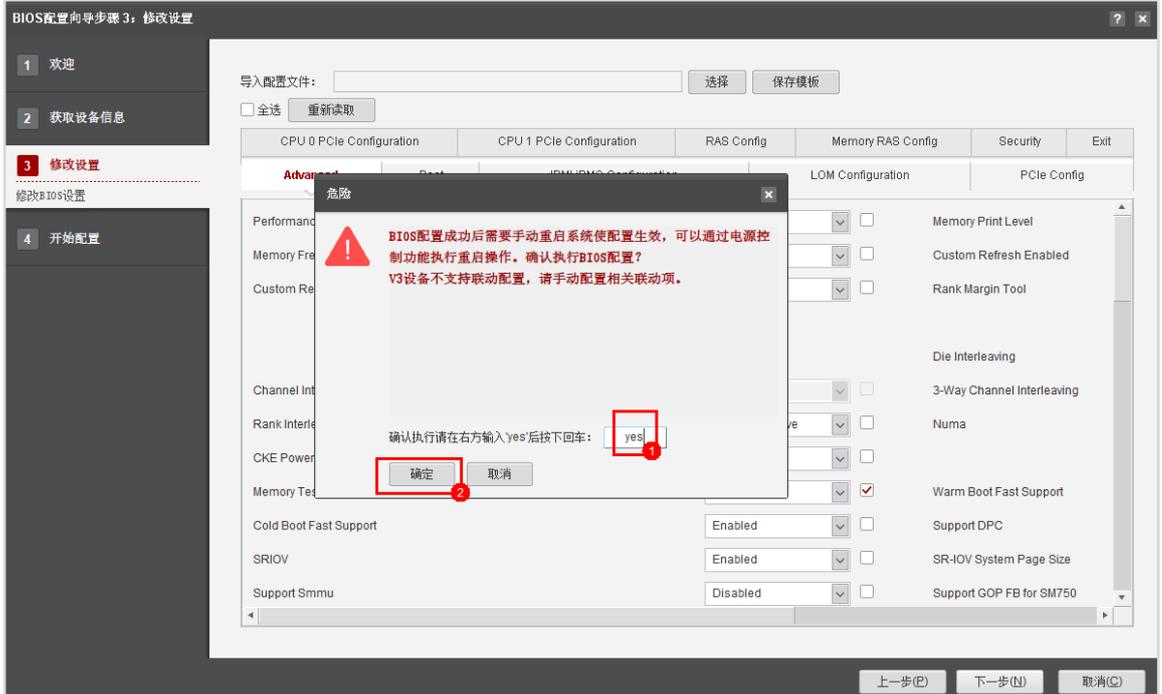
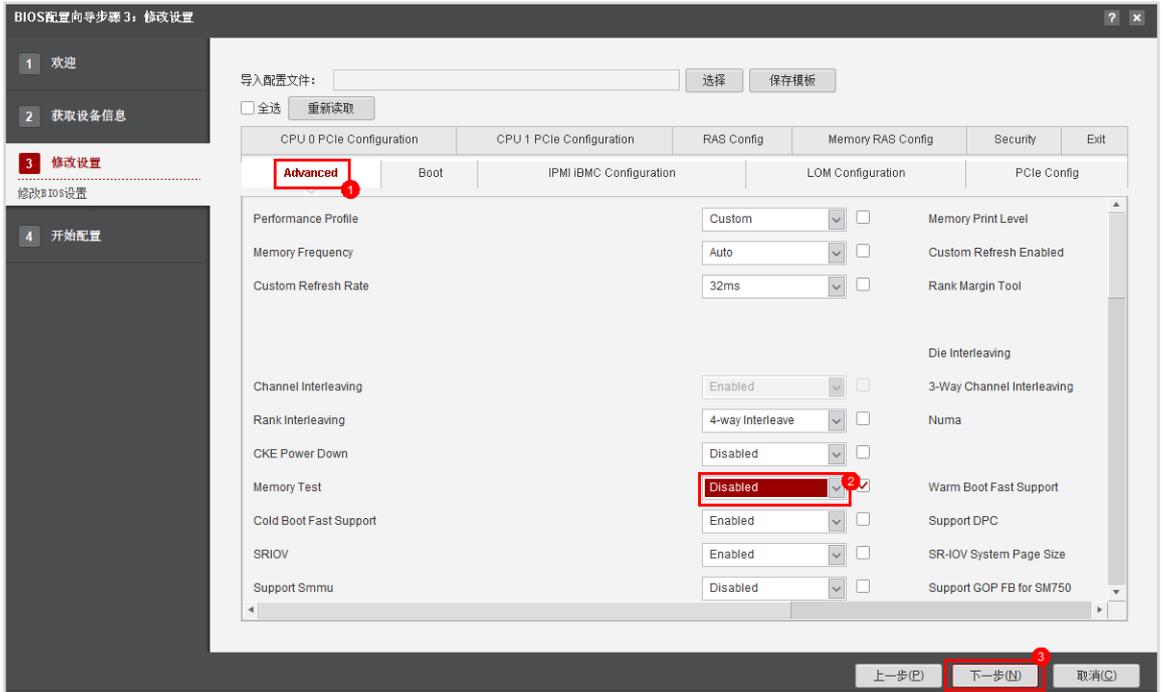
步骤 3 在“BIOS 配置向导步骤 2: 获取设备信息”界面，勾选服务器，点击“下一步”。



步骤 4 SmartKit 进行设备发现任务，等待片刻，看到结果是“PASS”，勾选服务器，点击“下一步”。

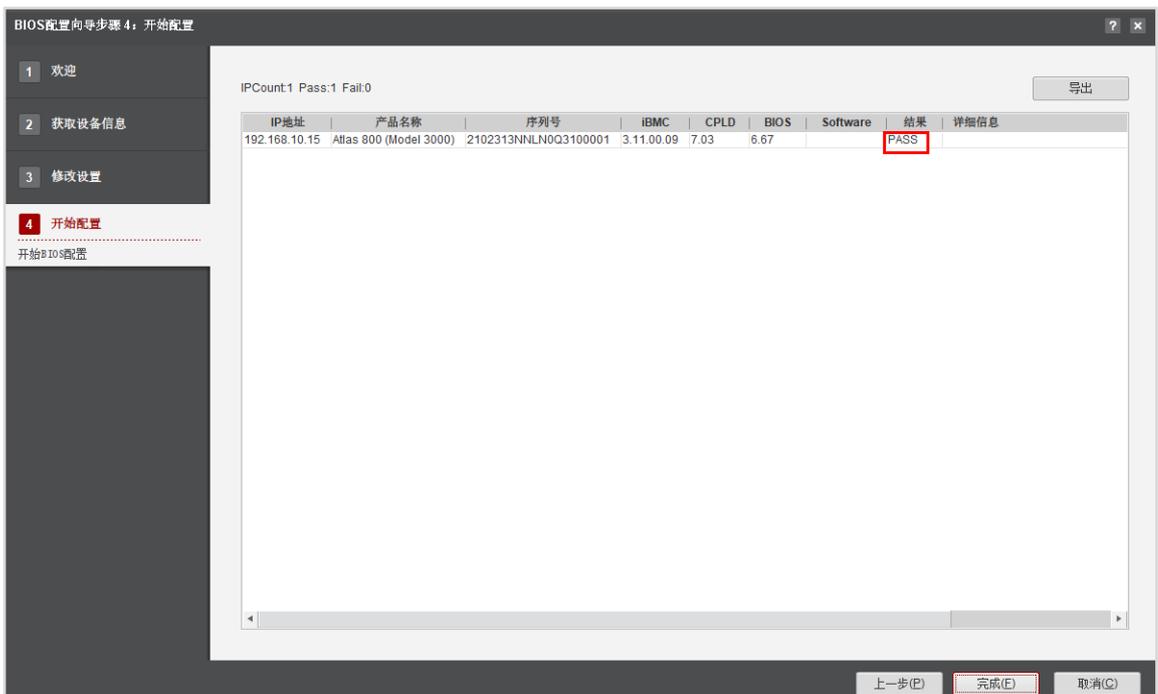
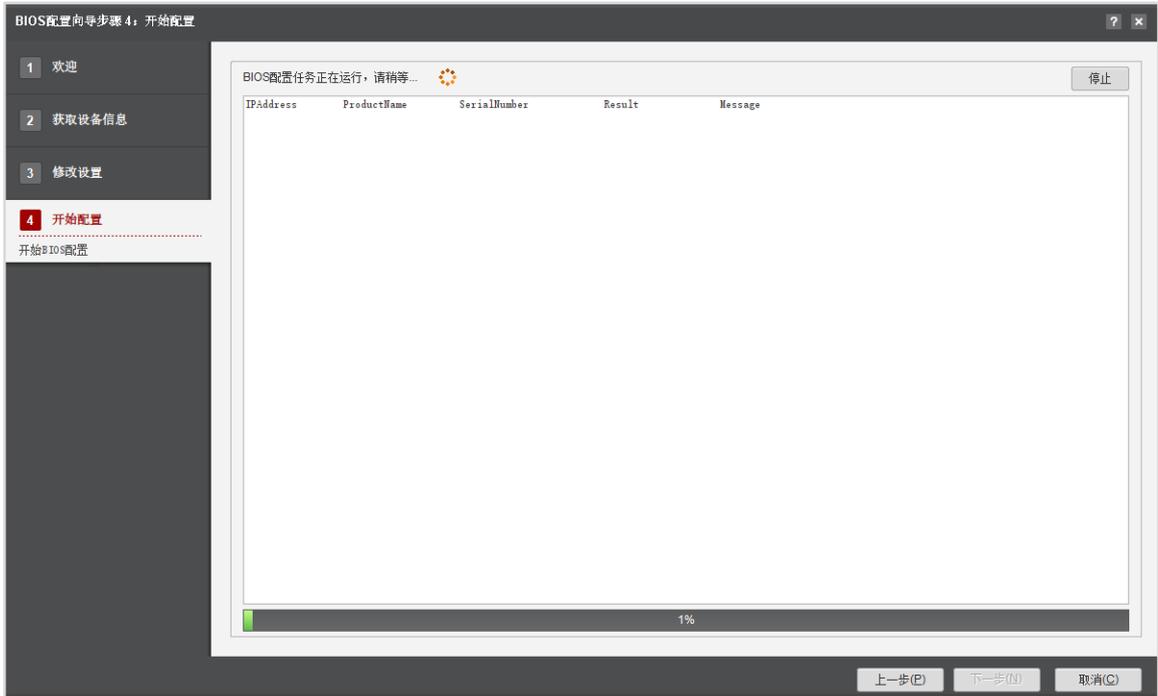


步骤 5 在“Advanced”的界面, 关闭“Memory Test”, 点击“下一步”, 弹出危险弹窗, 输入“yes”按下回车, 并点击“确定”。

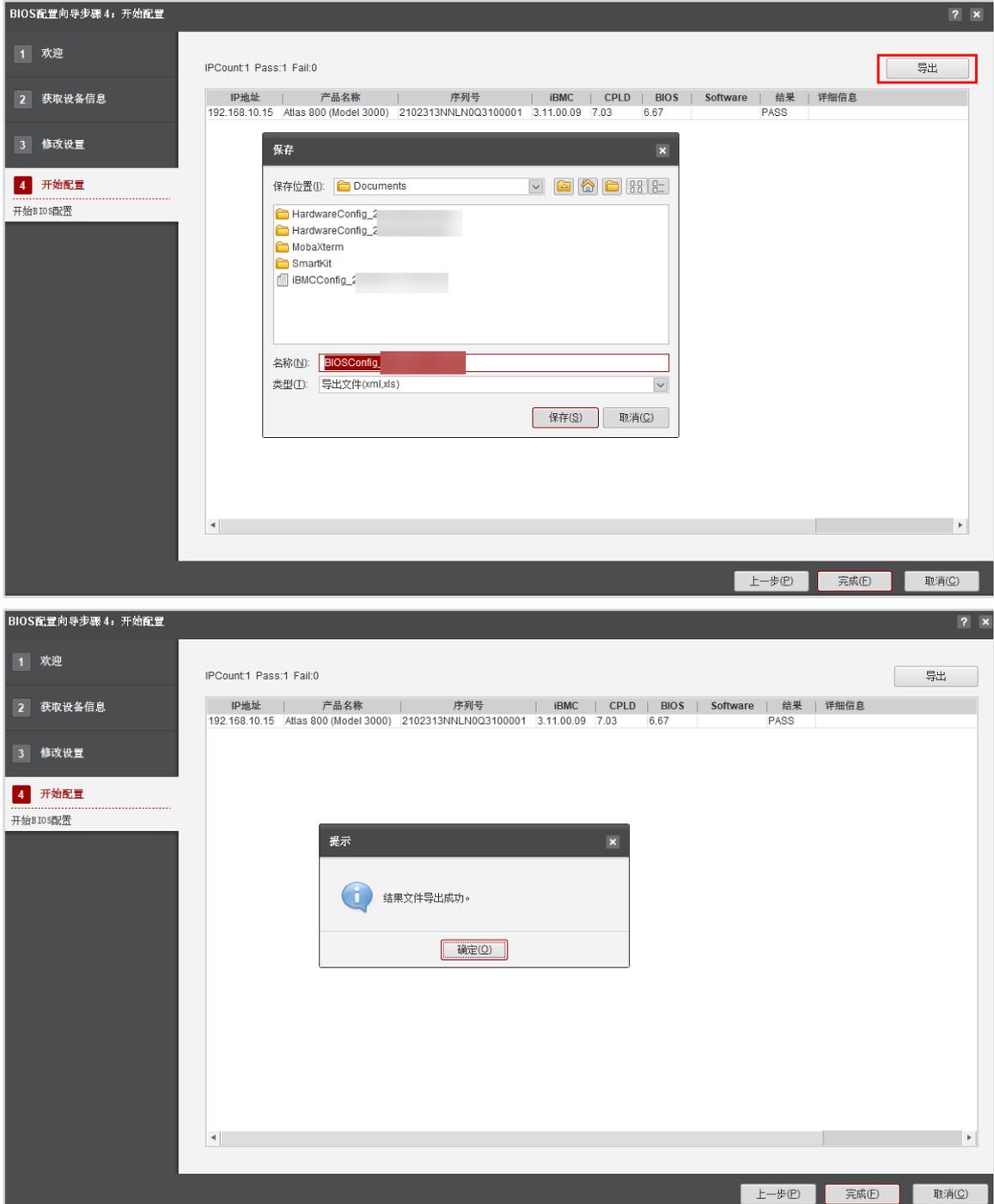


【注】BIOS 配置项非常多, 这里仅调节一个配置作为演示。

步骤 6 等待片刻, BIOS 配置任务为 PASS。



步骤 7 点击右上角“导出”，可把配置报告导出保存在本地。



步骤 8 点击右下角“完成”，结束 BIOS 配置任务，系统会自动回到“硬件配置”界面。

BIOS配置向导步骤 4: 开始配置

1 欢迎

2 获取设备信息

3 修改设置

4 开始配置

开始#ID配置

IPCount:1 Pass:1 Fail:0

导出

IP地址	产品名称	序列号	iBMC	CPLD	BIOS	Software	结果	详细信息
192.168.10.15	Atlas 800 (Model 3000)	2102313NNLN0Q3100001	3.11.00.09	7.03	6.67		PASS	

上一步(B) 完成(F) 取消(C)

SmartKit- 设备配置管理 首页 设备列表 Houp软件包

服务器 > 开局交付

< 返回首页

硬件配置

配置iBMC、BIOS、HMM、RAID。

配置iBMC、BIOS、HMM、RAID。

- iBMC配置  
成功: 1 台/套  
用时: 116时46分7秒
- BIOS配置  
成功: 1 台/套  
用时: 0时53分16秒
- HMM配置  
配置服务器的HMM。
- V3 RAID配置  
配置FusionServer V3及TaiSh...
- RAID配置  
配置FusionServer V5.  
FusionServe...

## 2.3.6 服务器电源控制

步骤 1 关闭“配置 iBMC、BIOS、HMM、RAID”弹出框，点击“电源控制”。

The image displays two screenshots of the SmartKit web interface, showing the progress of hardware configuration for a server.

**Top Screenshot:** The interface shows the "硬件配置" (Hardware Configuration) step. It lists five configuration items:

- iBMC配置:** 成功: 1台/套, 用时: 0时5分24秒. Status: Success (Green checkmark).
- BIOS配置:** 成功: 1台/套, 用时: 0时53分16秒. Status: Success (Green checkmark).
- HMM配置:** 配置服务器的HMM. Status: In Progress (Blue HMM icon).
- V3 RAID配置:** 配置FusionServer V3及TaiShan... Status: In Progress (Blue RAID icon).
- RAID配置:** 配置FusionServer V5, FusionServe... Status: In Progress (Blue RAID icon).

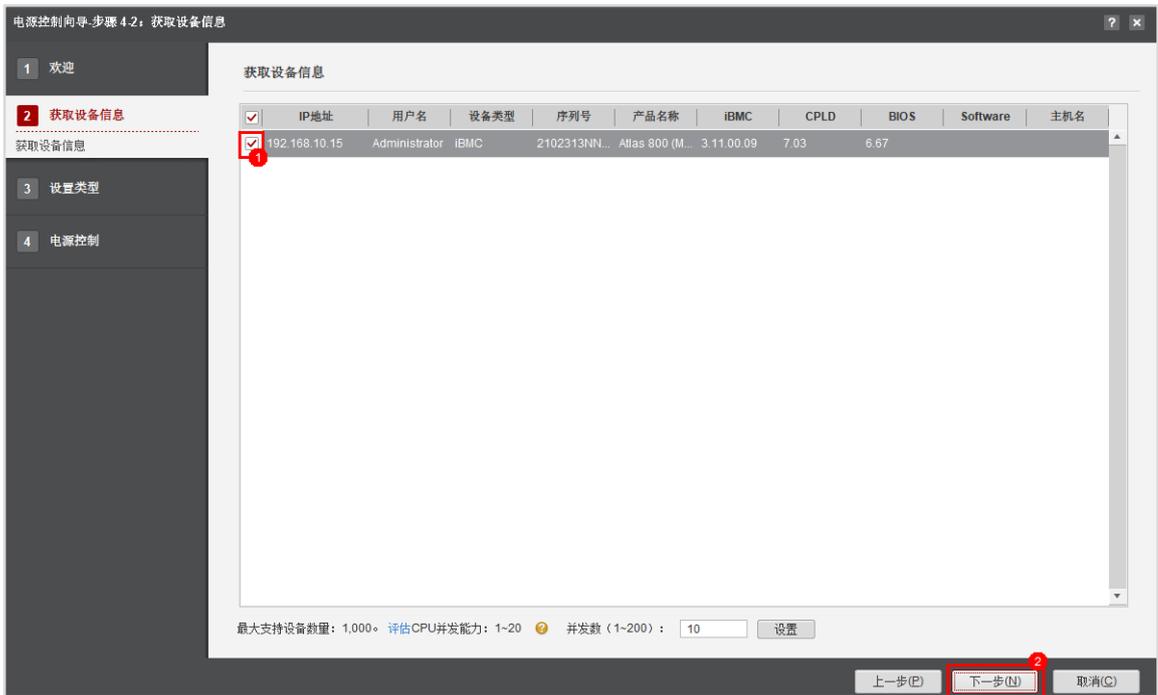
**Bottom Screenshot:** The interface shows the "电源控制" (Power Control) step, which is highlighted with a red box. It lists four steps in a sequence:

- 选择设备:** 选中设备: 1台/套, 用时: 0时2分11秒. Status: Success (Green checkmark).
- 硬件配置:** 配置iBMC、BIOS、HMM、RAID. Status: Success (Green checkmark).
- 电源控制:** 重启iBMC, 重启、上下电系统. Status: In Progress (Blue power icon).
- 配置检查:** 检查服务器的iBMC、BIOS、RAID配置. Status: In Progress (Blue magnifying glass icon).

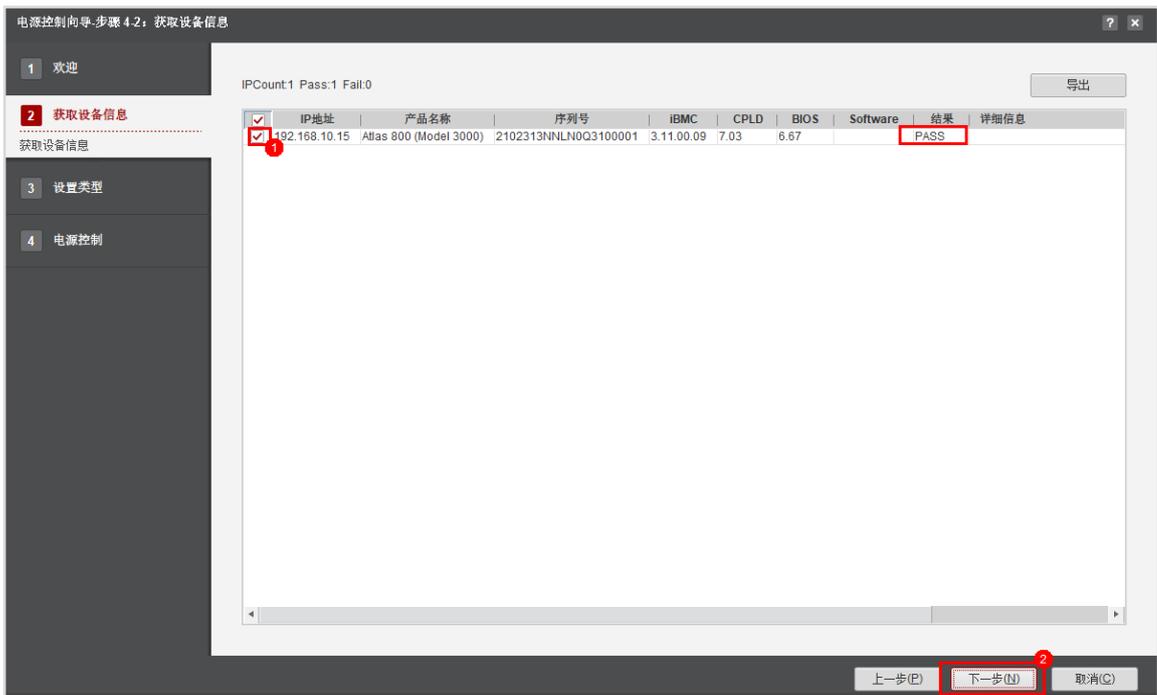
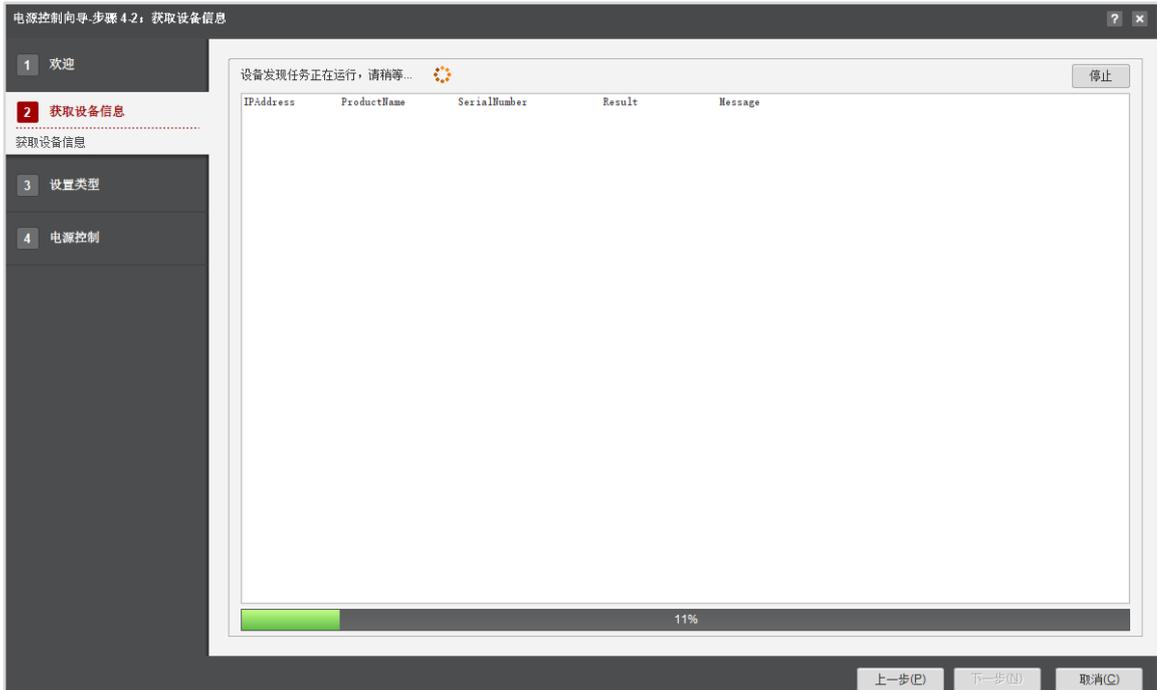
步骤 2 在“电源控制向导-步骤 4-1: 欢迎”界面, 点击“下一步”。



步骤 3 在“电源控制向导-步骤 4-2: 获取设备信息”界面，勾选服务器，点击“下一步”。



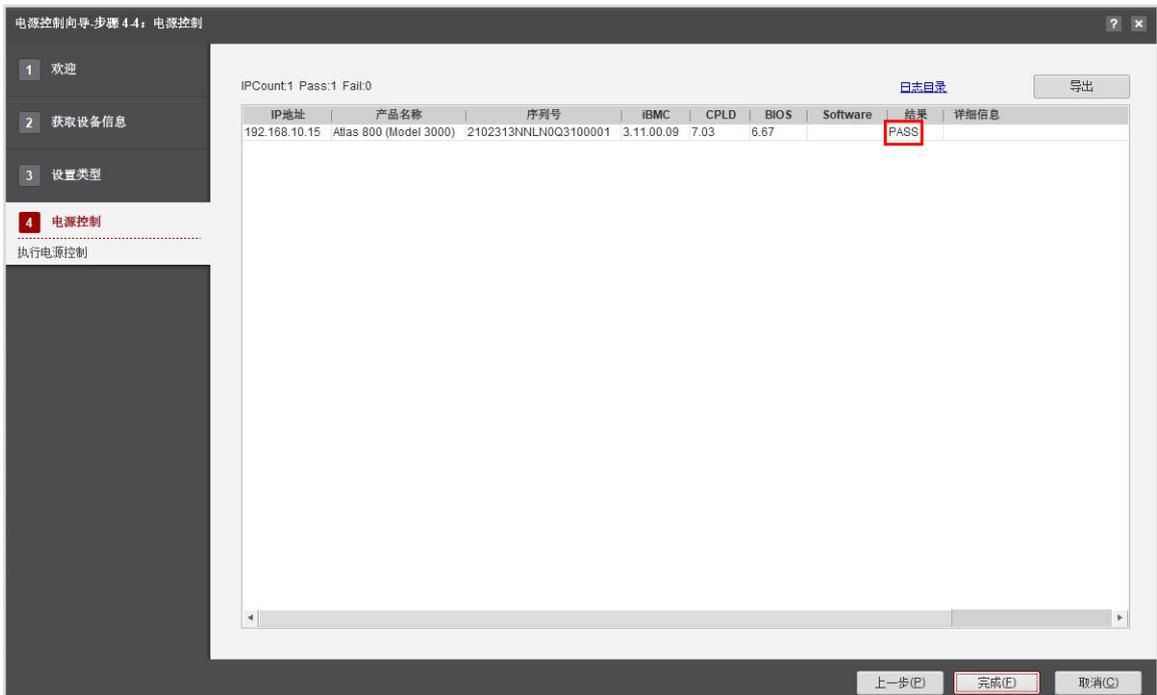
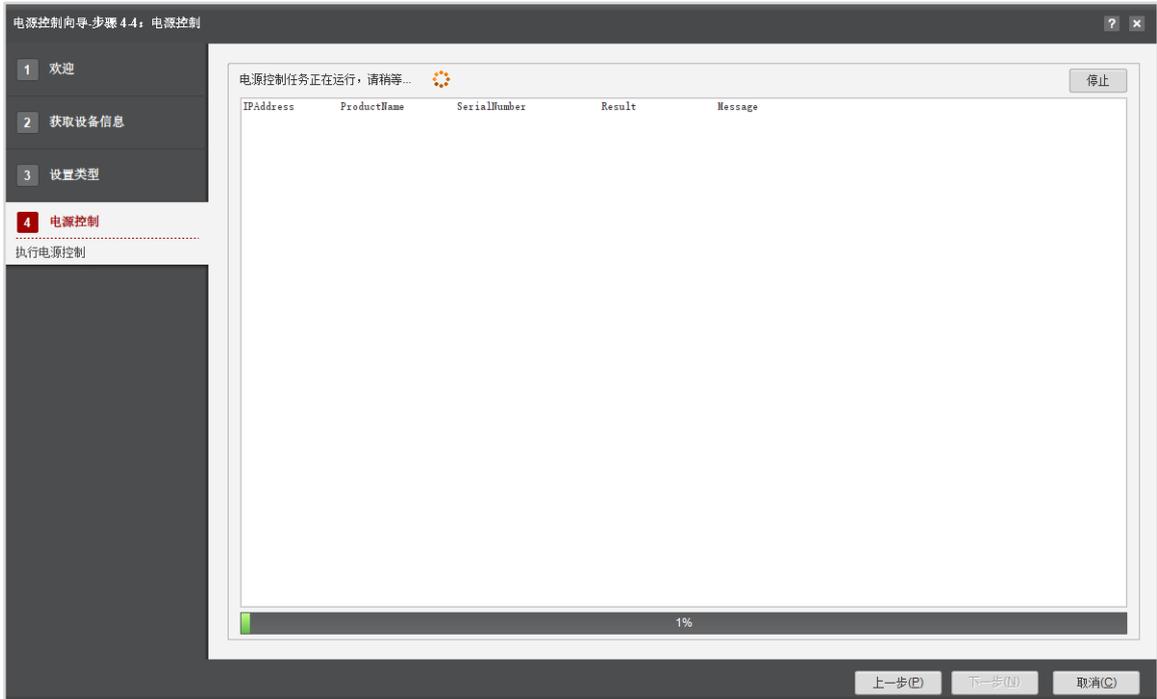
步骤 4 等待片刻，SmartKit 进行设备发现，当校验结果为“PASS”时，勾选服务器，点击“下一步”。



步骤 5 在“电源控制向导-步骤 4-3：设置类型”界面，勾选“系统重启”，点击“下一步”，弹出危险弹窗，输入“yes”按下回车，并点击“确定”。

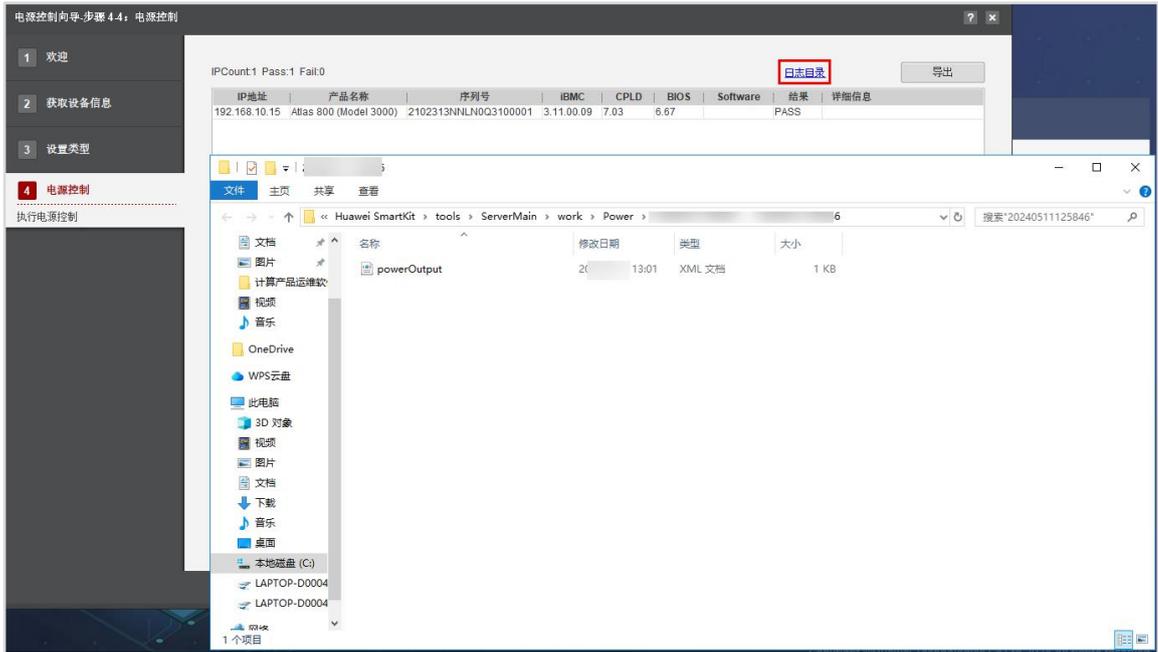


步骤 6 等待片刻, 电源控制任务为 PASS。

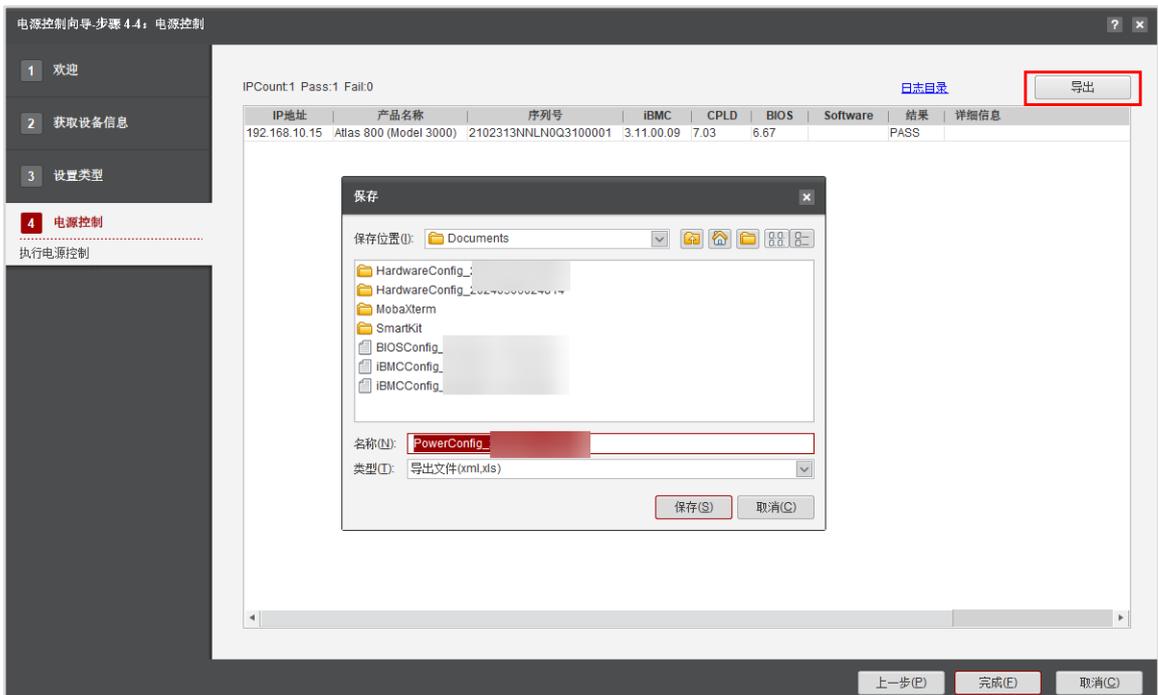


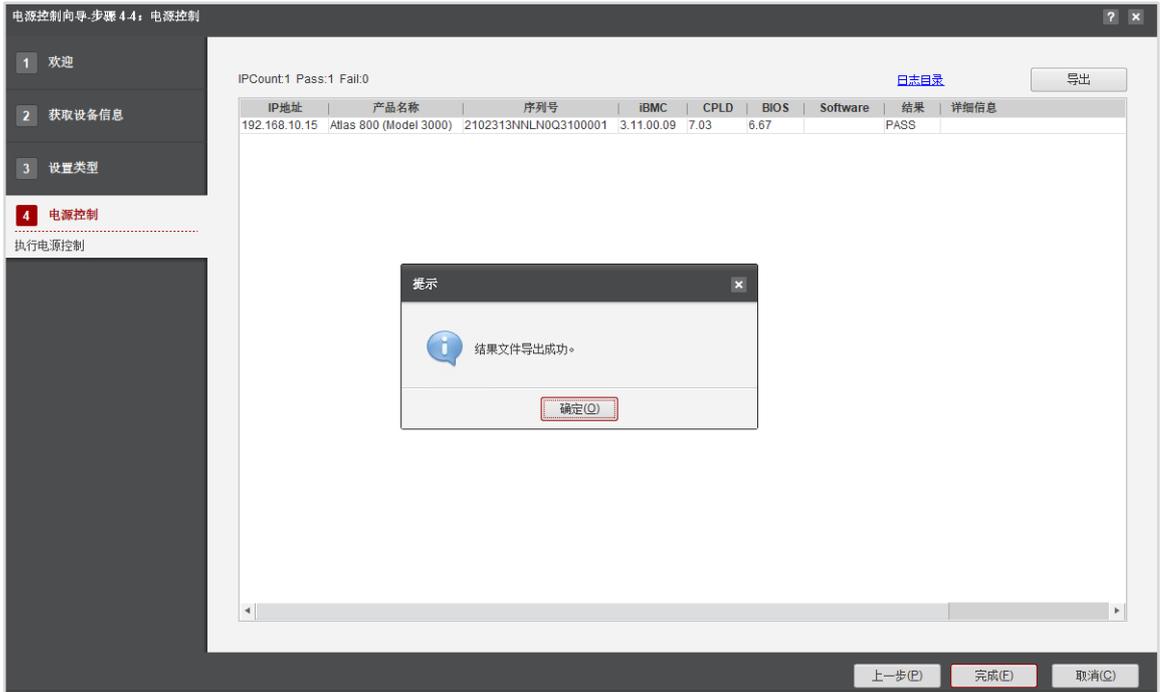
【注】服务器系统重启需要时间，此时电源控制任务消耗时间较长。

步骤 7 点击右上角“日志目录”，可查看详细电源控制日志。

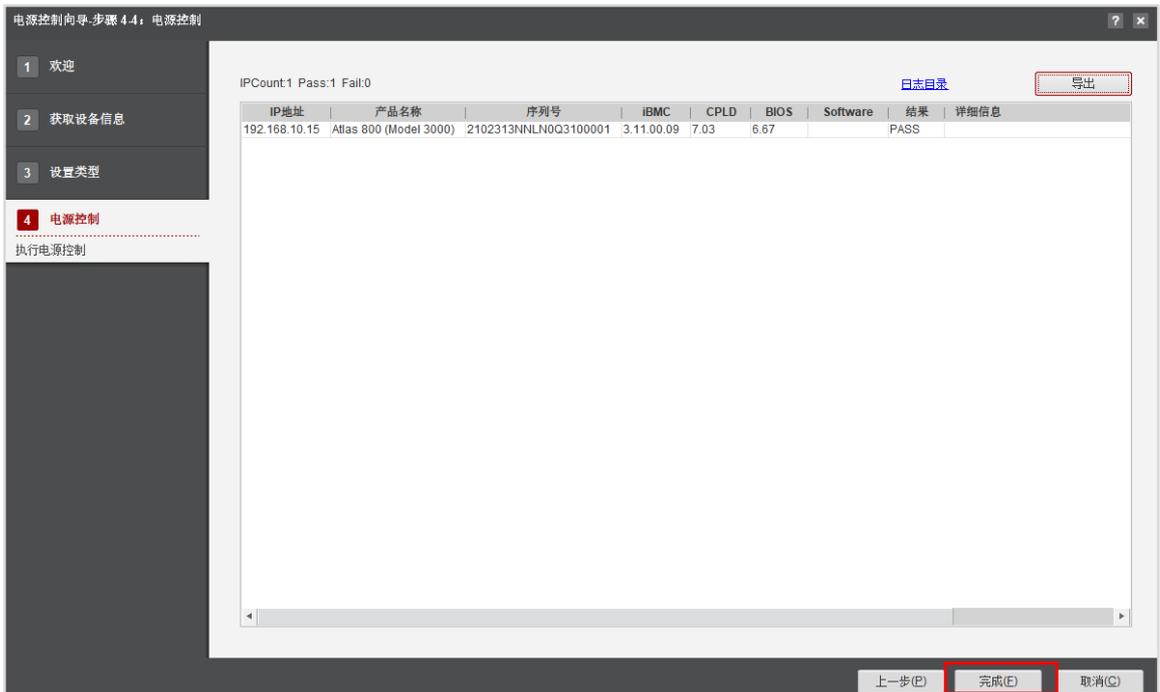


步骤 8 点击右上角“导出”，可把配置报告导出保存在本地。





步骤 9 点击右下角“完成”，结束电源控制任务，系统会自动回到“硬件配置”界面。





## 2.3.7 服务器配置检查

**步骤 1** 获取服务器的标准基线配置文件“Standard Configuration File.rar”，并进行解压，解压后的标准配置文件列表如下图所示，这些文件用于校验待检查的服务器的配置是否满足基线要求。

名称	修改日期	类型	大小
Standard Configuration File		压缩(zipped)文件...	40 KB

名称	修改日期	类型	大小
config		XML 文档	99 KB
currentRaidvalue.json		JSON 文件	25 KB
currentvalue.json		JSON 文件	13 KB
registry.json		JSON 文件	694 KB

【注】各模块标准配置文件说明如下：

- iBMC:  
标准配置文件：config.xml。
- BIOS:  
标准配置文件：currentvalue.json 或 options0.ini。  
导入模板文件：registry.json 或 dispaly0.ini。
- RAID:  
标准配置文件：currentRaidvalue.json。

**步骤 2** 在 SmartKit 的“硬件配置”界面，点击“配置检查”。



步骤 3 在“检查服务器的 iBMC、BIOS、RAID 配置”弹出框，点击“配置检查”。



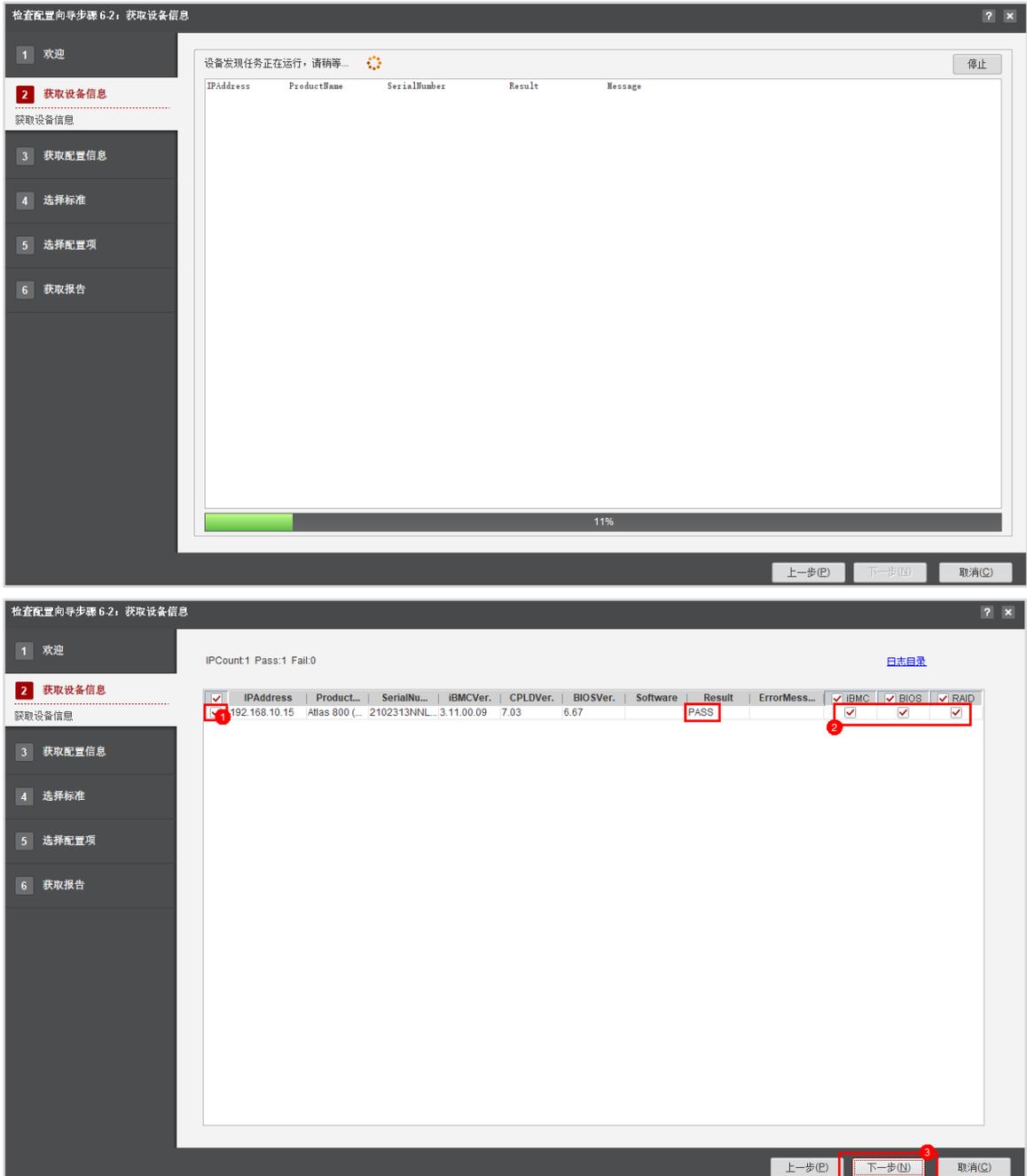
步骤 4 在“检查配置向导步骤 6-1: 欢迎”界面，点击“下一步”。



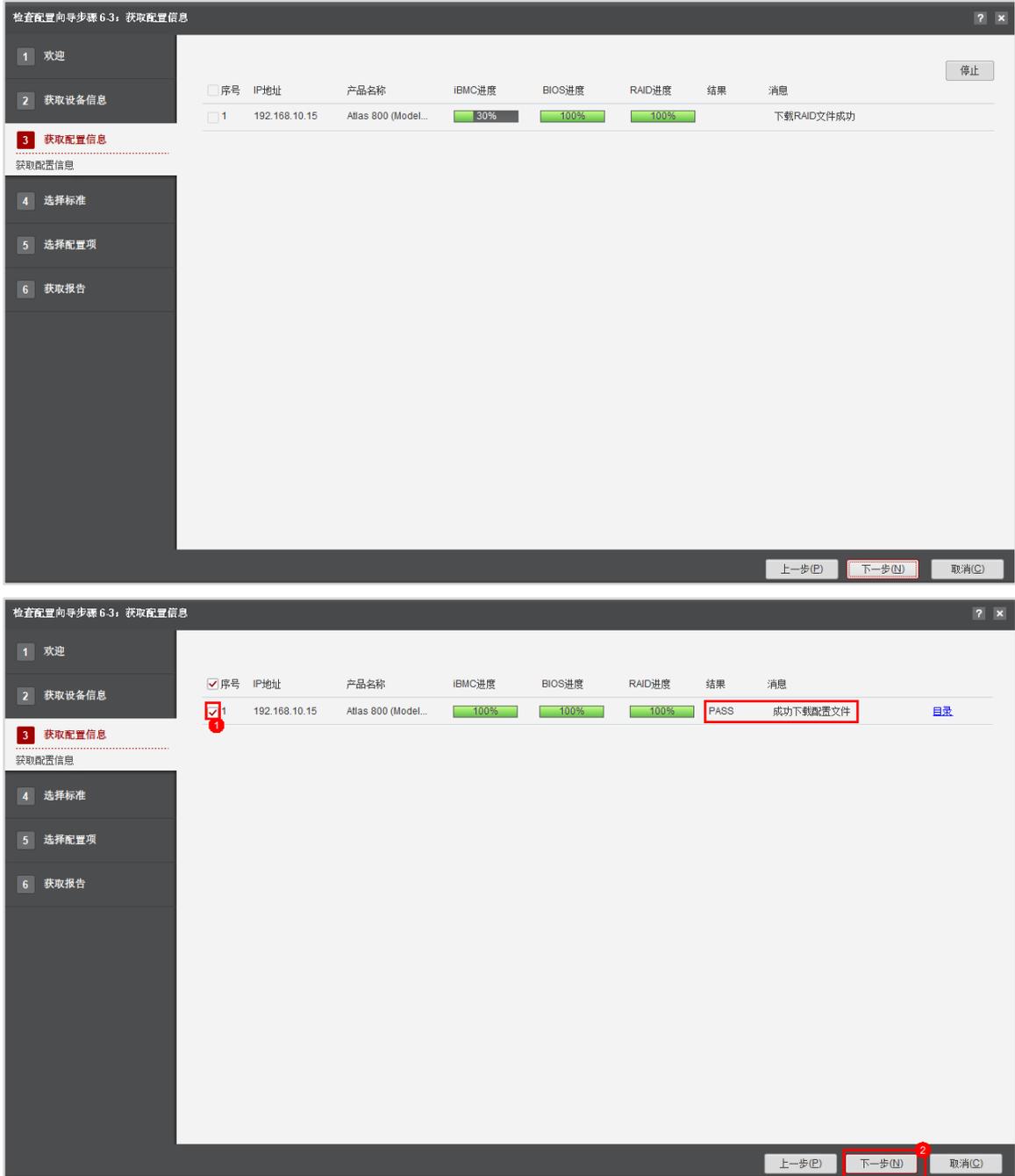
步骤 5 在“检查配置向导步骤 6-2: 获取设备信息”界面，勾选服务器，点击“下一步”



步骤 6 等待片刻，SmartKit 进行设备发现，当校验结果为“PASS”时，勾选服务器，并同时勾选 iBMC、BIOS 和 RAID，点击“下一步”



步骤 7 在“检查配置向导步骤 6-3: 获取配置信息”界面, 等待下载 iBMC、BIOS 和 RAID 文件成功, 勾选服务器, 点击“下一步”。

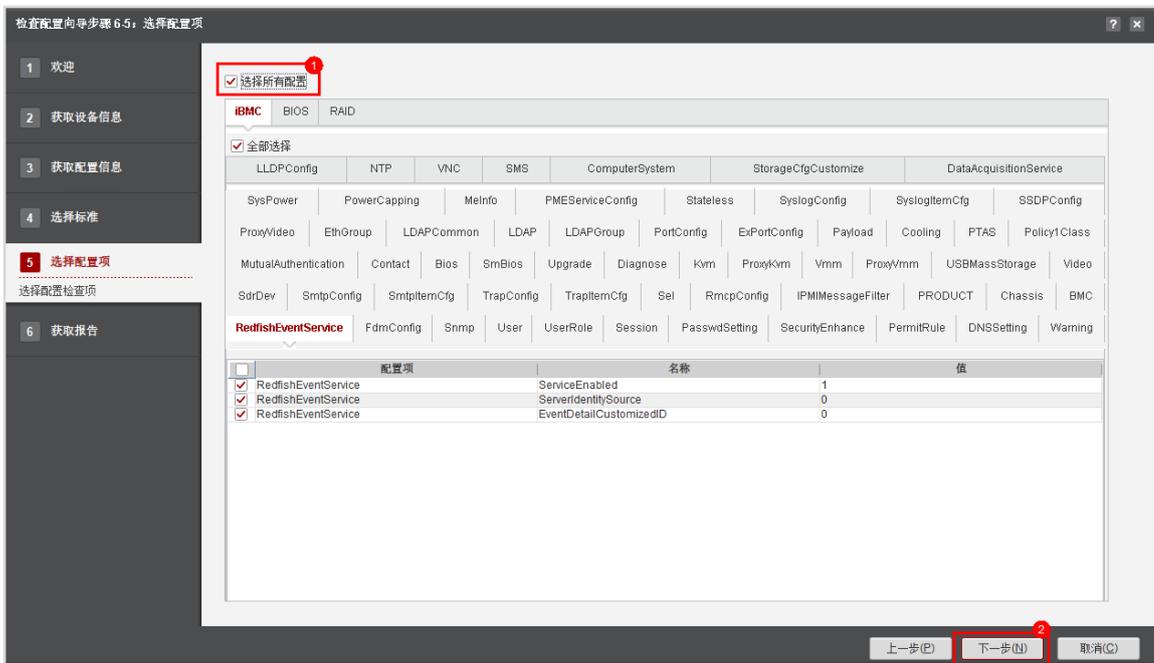


步骤 8 按照下图，分别选择 2.3.7 小节步骤 1 里准备好的 iBMC、BIOS 和 RAID 的标准配置文件和模板文件，点击“下一步”。

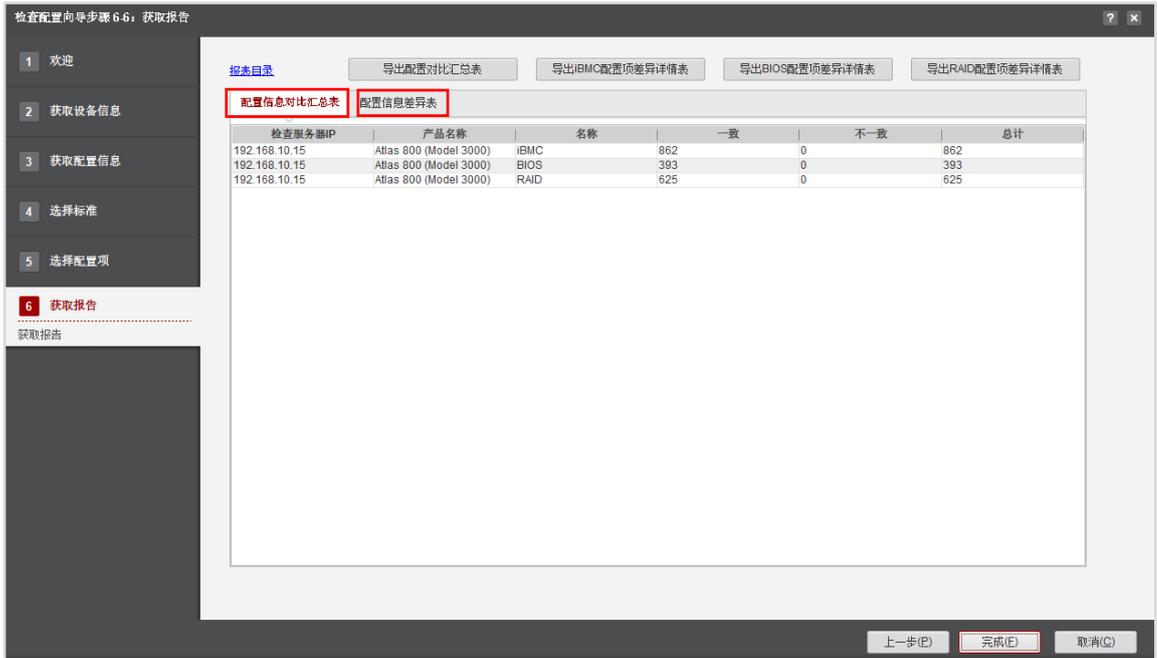


【注】标准选择里有两种方式，第一种以某一台服务器的配置作为标准，校验其他服务器的配置是否满足标准，本实验由于只涉及一台服务器，无法使用这种方式；第二种，获取标准配置文件，导入的方式进行校验其他服务器的配置是否满足标准，本实验采用这种方式，提前准备好了标准配置文件。

步骤 9 在“检查配置向导步骤 6-5：选择配置项”界面，勾选“选择所有配置”，点击“下一步”。

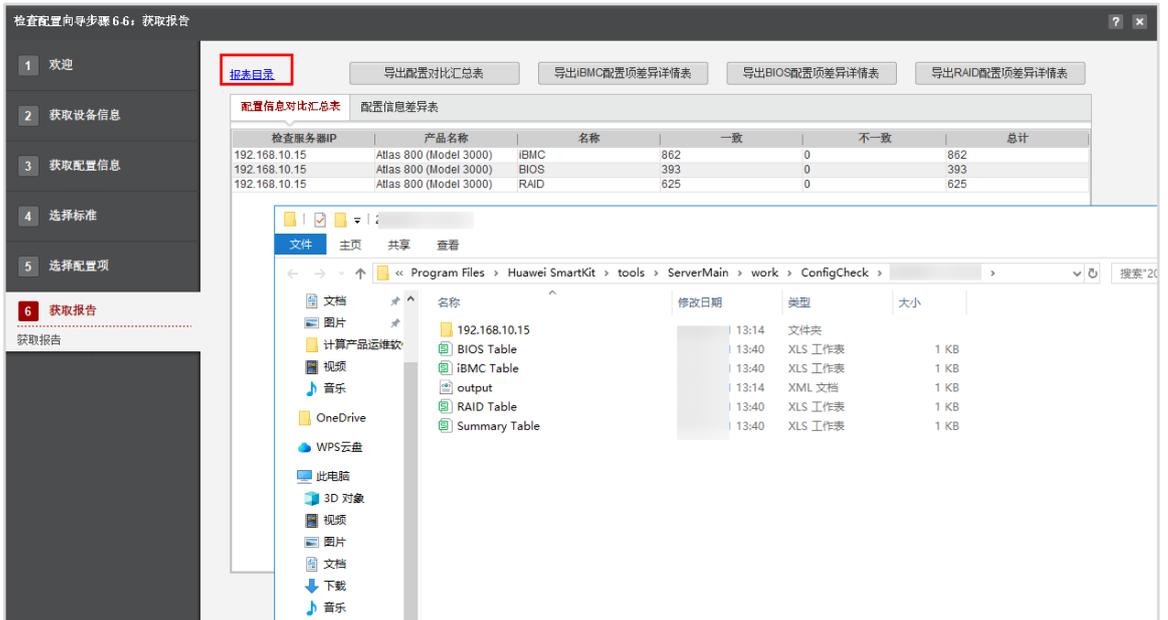


步骤 10 在“检查配置向导步骤 6-6：获取报告”界面，可以查看“配置信息对比汇总表”和“配置信息差异表”。

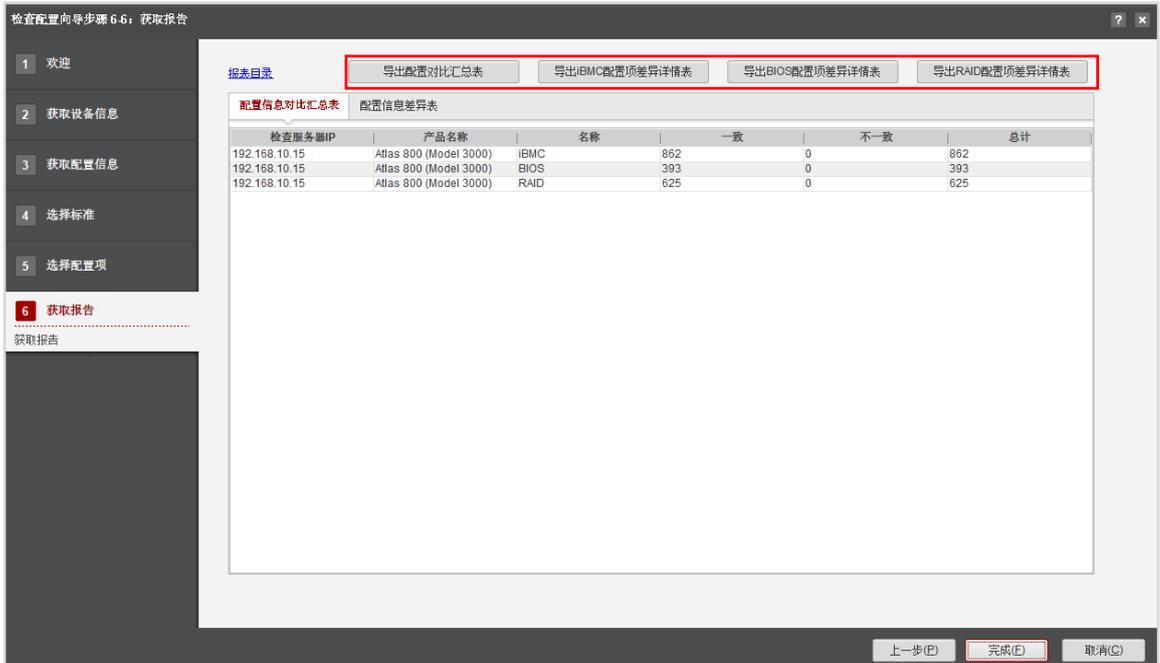


【注】从对比结果发现，待校验服务器的配置与标准配置完全一致。

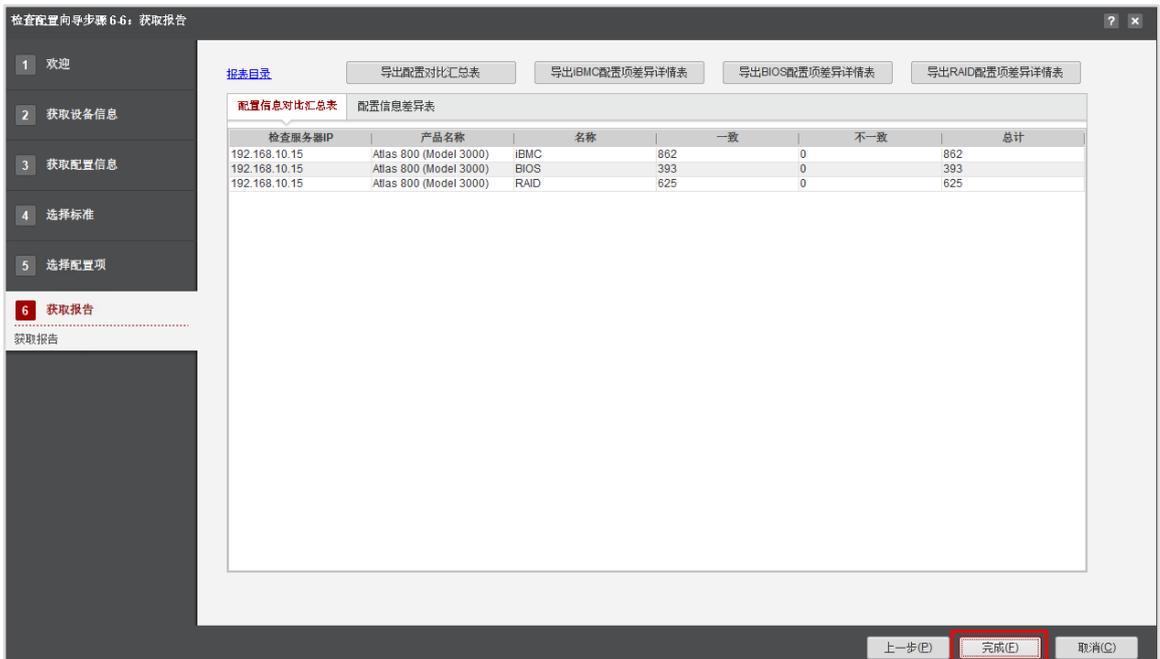
步骤 11 点击左上角“报表目录”，可查看详细报表本地文件。



步骤 12 点击右上角导出功能键，可分别导出“配置对比汇总表”、“iBMC 配置项差异详情表”、“BIOS 配置项差异详情表”和“RAID 配置项差异详情表”。



步骤 13 点击右下角“完成”，结束检查配置任务，系统会自动回到“硬件配置”界面。





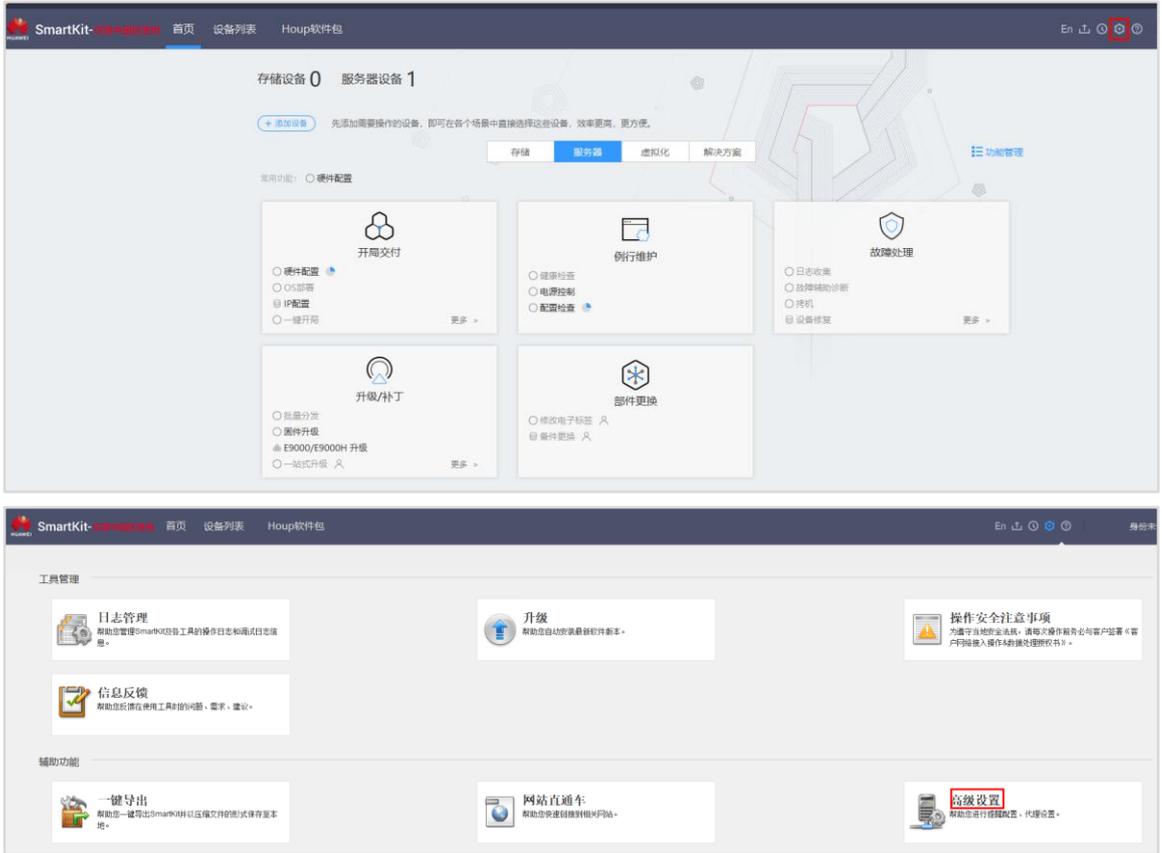
## 2.3.8 服务器健康检查

步骤 1 关闭“检查服务器的 iBMC、BIOS、RAID”配置弹窗，点击“返回首页”，系统会弹出提示框，点击“确定”。





步骤 2 在“首页”，点击右上角“设置”按钮，在“高级设置 > 安全设置”里，勾选“阅读并同意华为隐私声明”，点击“确定”。

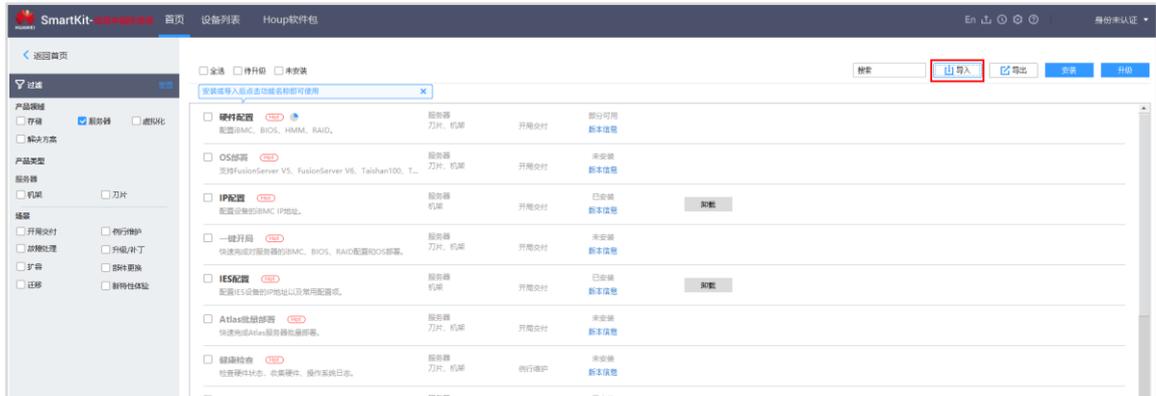




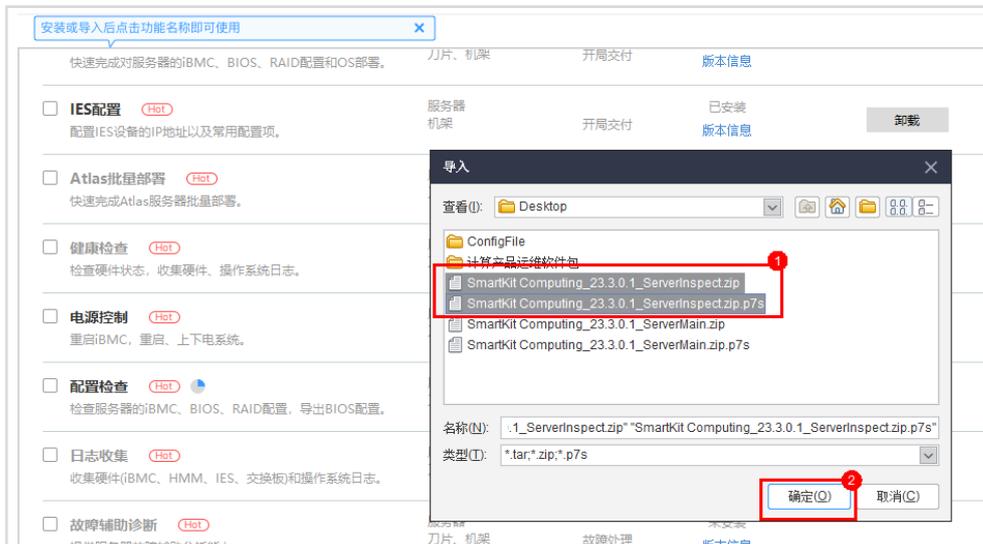
步骤 3 在“首页”，点击“功能管理”。



步骤 4 在功能管理界面，点击右上角“导入”。

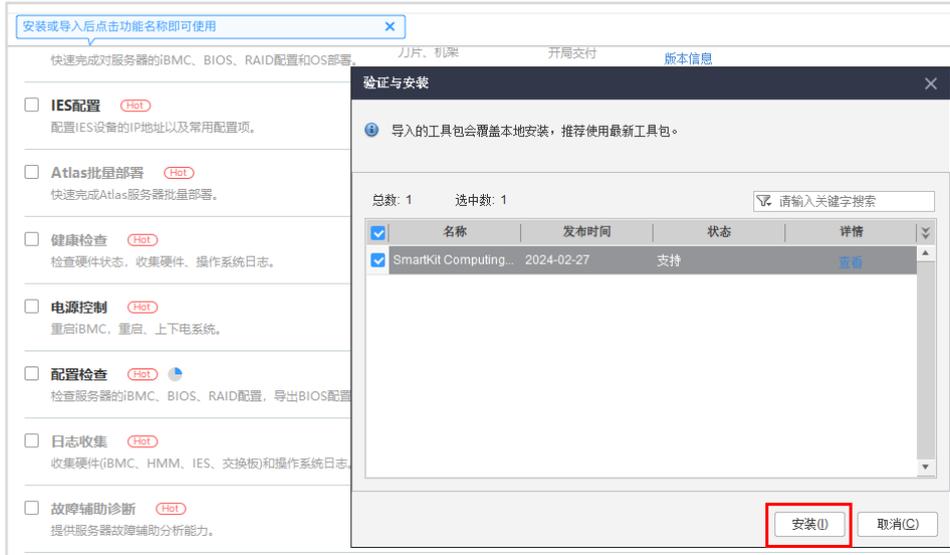


步骤 5 选择“SmartKit Computing\_23.3.0.1\_ServerInspect.zip”功能包及 p7s 校验文件，点击“确定”。

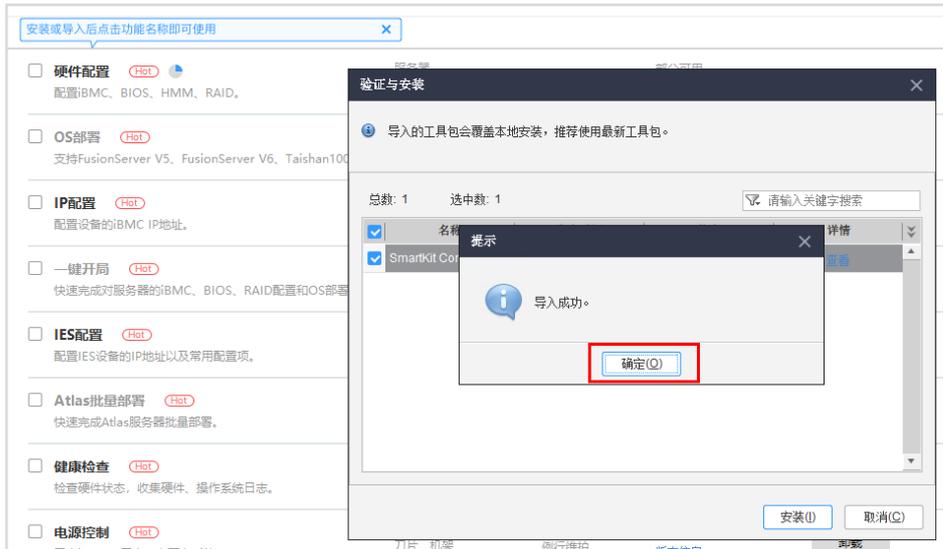


步骤 6 待系统验证通过后，点击“安装”。





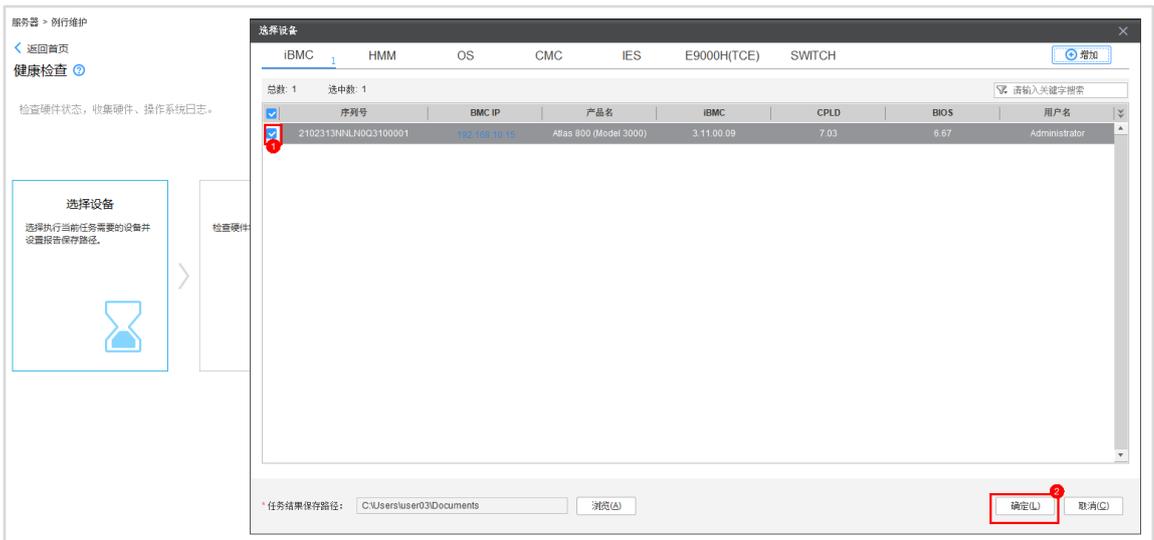
步骤 7 弹出“导入成功”提示框，点击“确定”。



步骤 8 返回“首页”，点击“例行维护 > 健康检查”。



步骤 9 在“健康检查”界面，点击“选择设备”，勾选服务器，并点击“确定”。



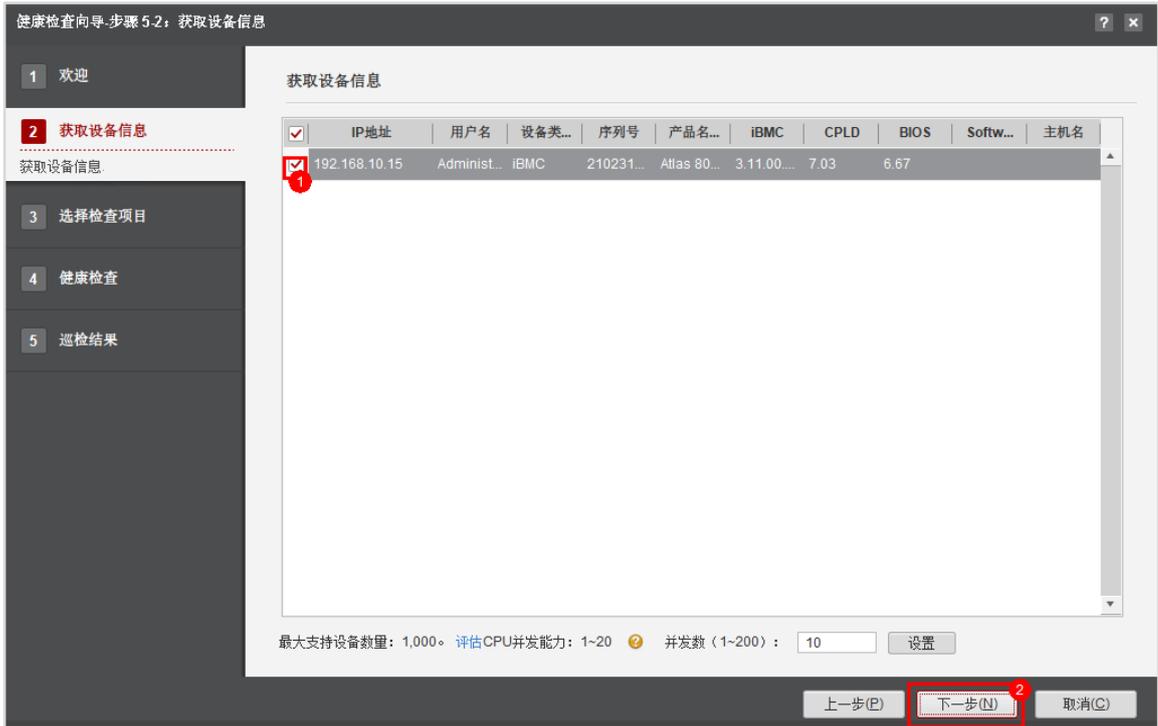
步骤 10 点击“健康检查”。



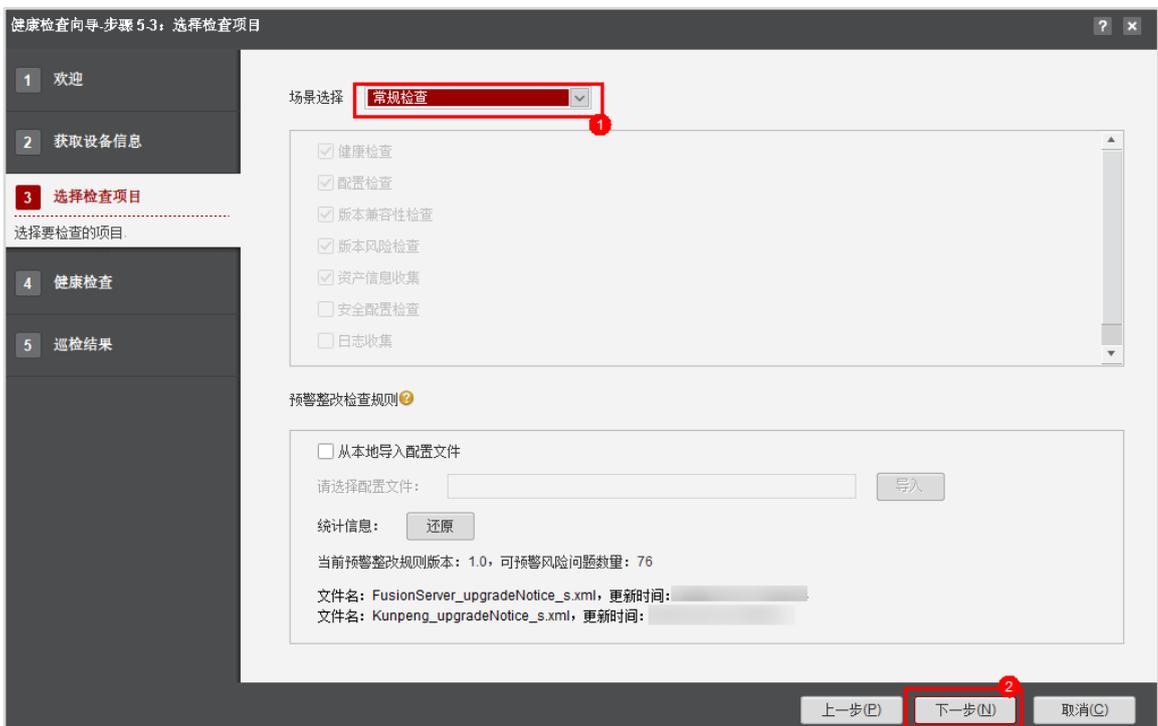
步骤 11 在“健康检查向导-步骤 5-1: 欢迎”界面, 点击“下一步”。



步骤 12 在“健康检查向导-步骤 5-2: 获取设备信息”界面, 勾选服务器, 点击“下一步”。



步骤 13 在“健康检查向导-步骤 5-3：选择检查项目”界面，选择“常规检查”，点击“下一步”。



步骤 14 在“健康检查向导-步骤 5-4：健康检查”界面，等待巡检完毕，点击“下一步”。

健康检查向导-步骤 5.4：健康检查

1 欢迎

2 获取设备信息

3 选择检查项目

**4 健康检查**

健康检查

5 巡检结果

全部成功  全部失败 再次巡检 停止

<input checked="" type="checkbox"/> 序号	IP地址	产品名称	进度	开始时间	耗时	结果	详情
<input checked="" type="checkbox"/> 1	192.168.10.15		0%	14:43:54	00:00:02	巡检开始...	

上一步(B) 下一步(N) 取消(C)

健康检查向导-步骤 5.4：健康检查

1 欢迎

2 获取设备信息

3 选择检查项目

**4 健康检查**

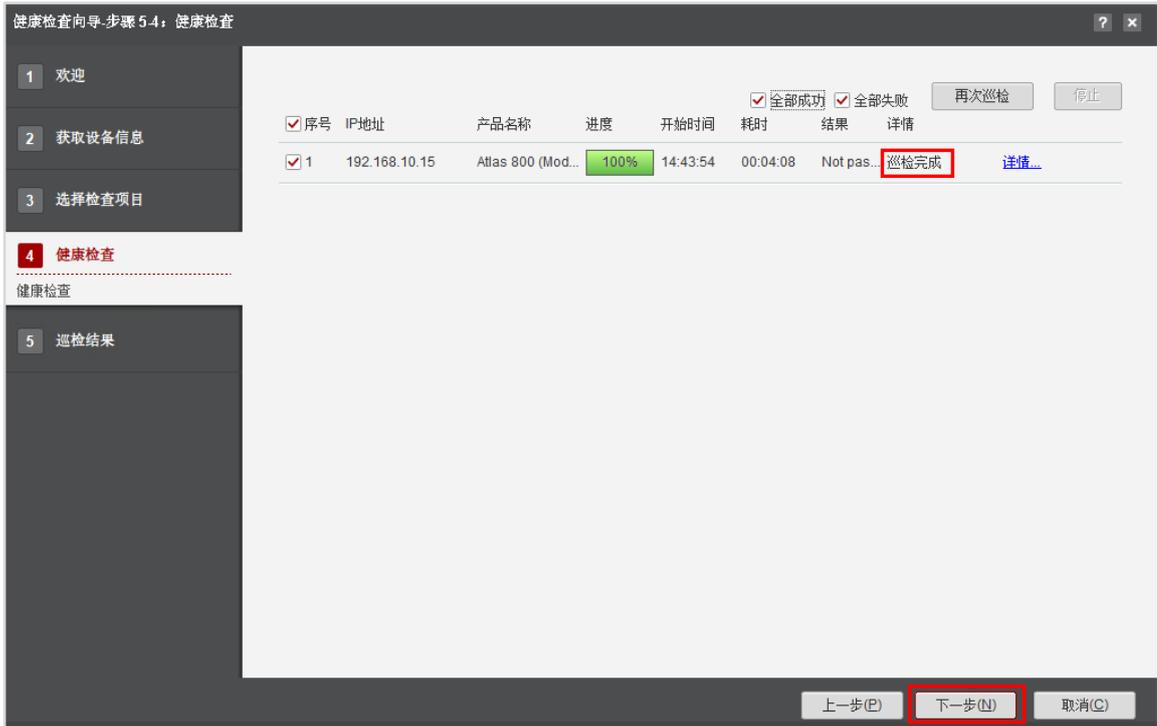
健康检查

5 巡检结果

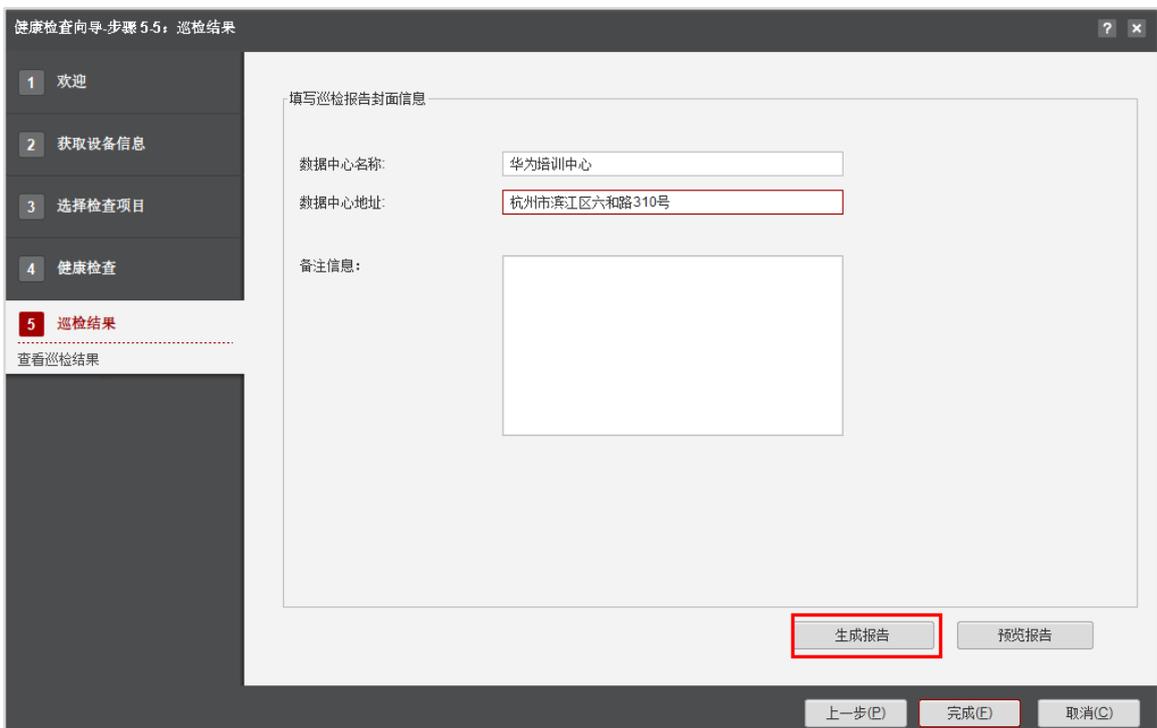
全部成功  全部失败 再次巡检 停止

<input checked="" type="checkbox"/> 序号	IP地址	产品名称	进度	开始时间	耗时	结果	详情
<input checked="" type="checkbox"/> 1	192.168.10.15	Atlas 800 (Mod...	4%	14:43:54	00:01:13	Hardware Infor...	

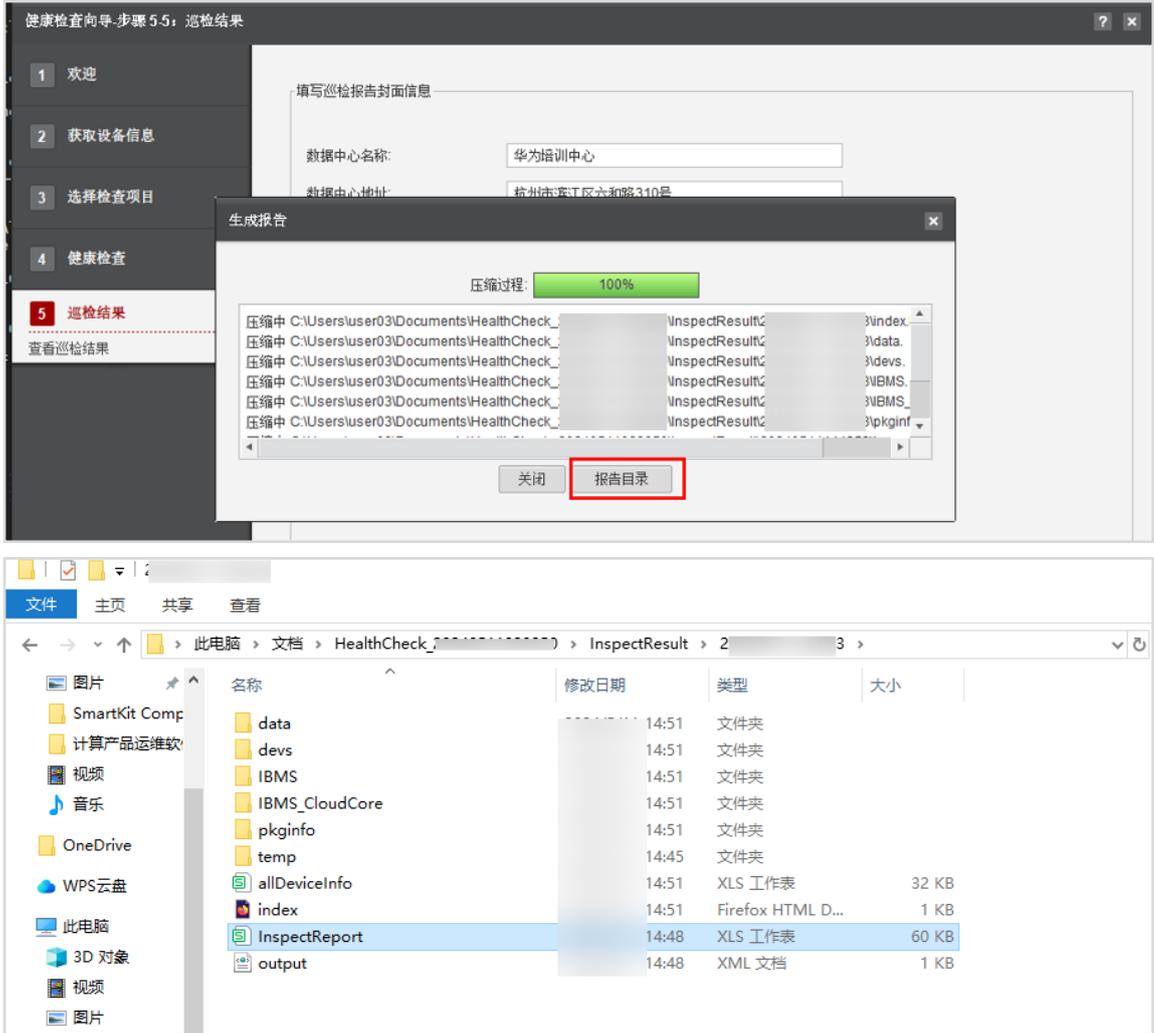
上一步(B) 下一步(N) 取消(C)



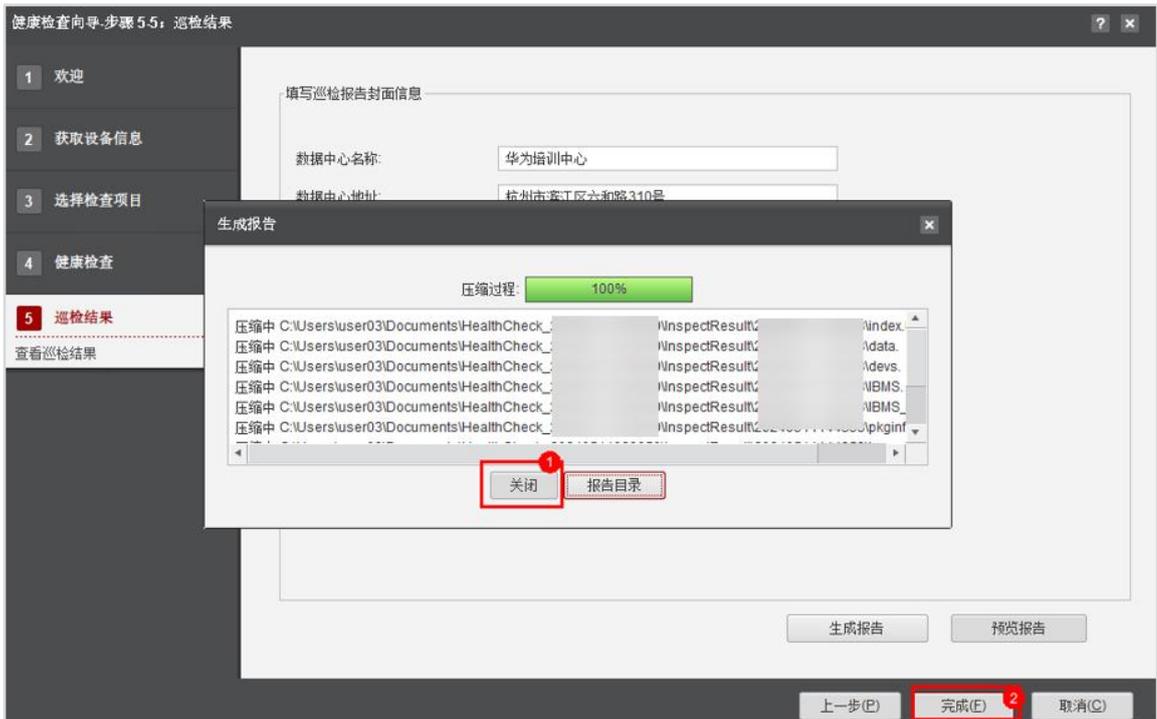
步骤 15 在“健康检查向导-步骤 5-5: 巡检结果”界面，自定义填写相关信息，点击“生成报告”。



步骤 16 点击“报告目录”，可在本地查看巡检报告。



步骤 17 关闭“生成报告”窗口，并点击右下角“完成”，关闭“健康检查”任务，系统会自动会到“健康检查”界面。



【注】这里出现感叹号是由于服务器健康检查部分检查项未通过，通过报告可以详细查看未通过的检查项，并根据报告里的处理建议进行解决。

## 2.3.9 服务器日志收集

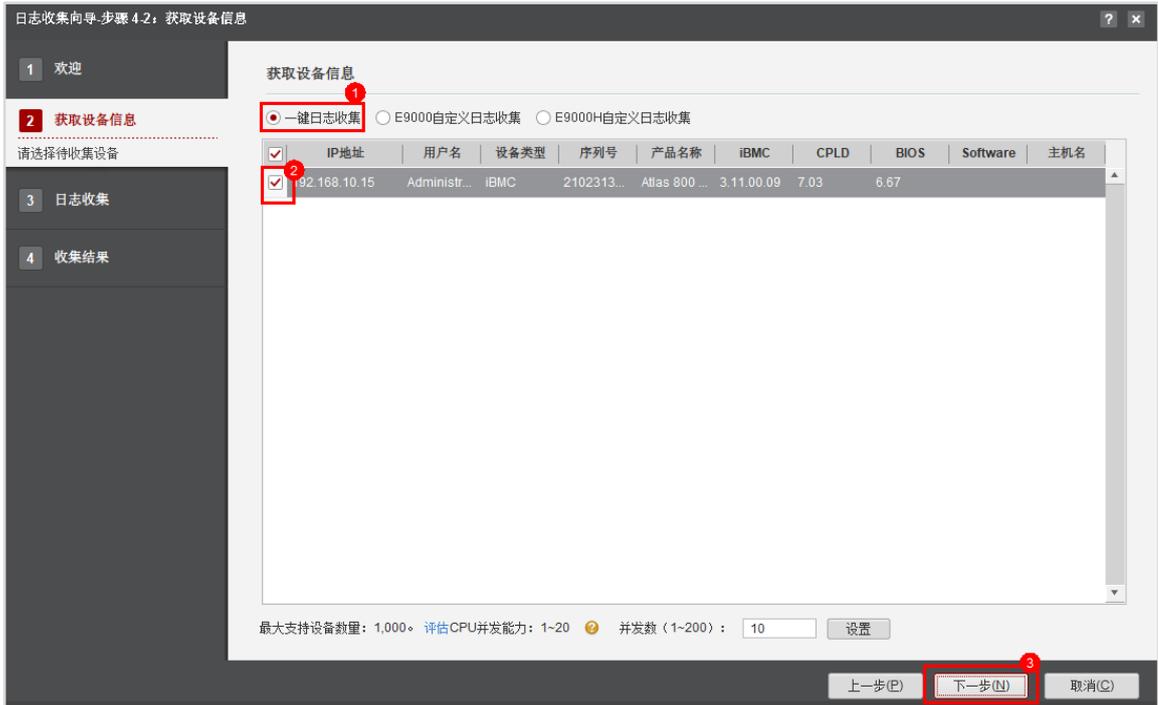
步骤 1 在“健康检查”界面，点击“日志收集”。



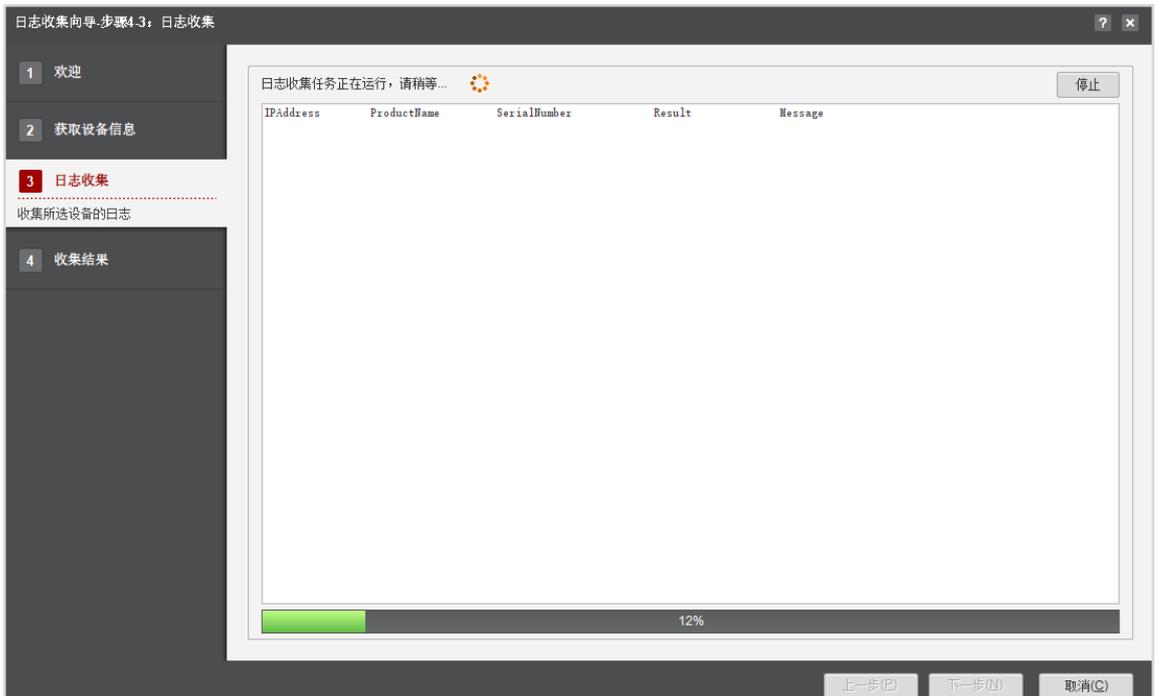
步骤 2 在“日志收集向导-步骤 4-1: 欢迎”界面，点击“下一步”。

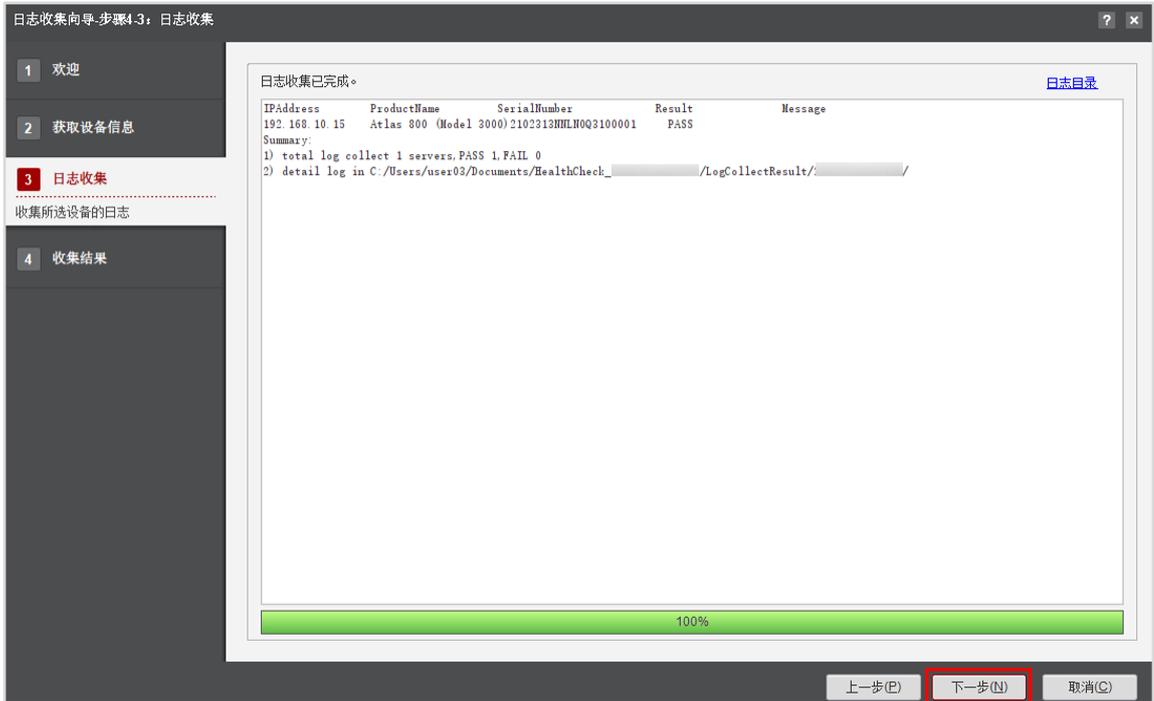


步骤 3 在“日志收集向导-步骤 4-2: 获取设备信息”界面，点击“一键日志收集”，勾选服务器，然后点击“下一步”。

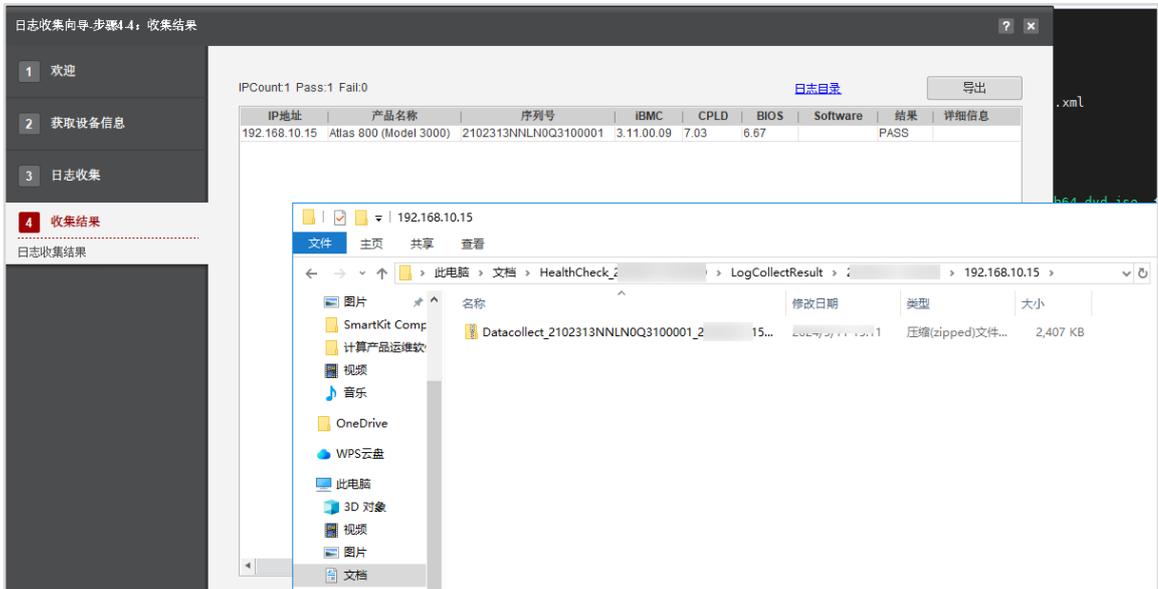


步骤 4 在“日志收集向导-步骤 4-3: 日志收集”界面等待片刻，系统完成日志收集任务，点击“下一步”。

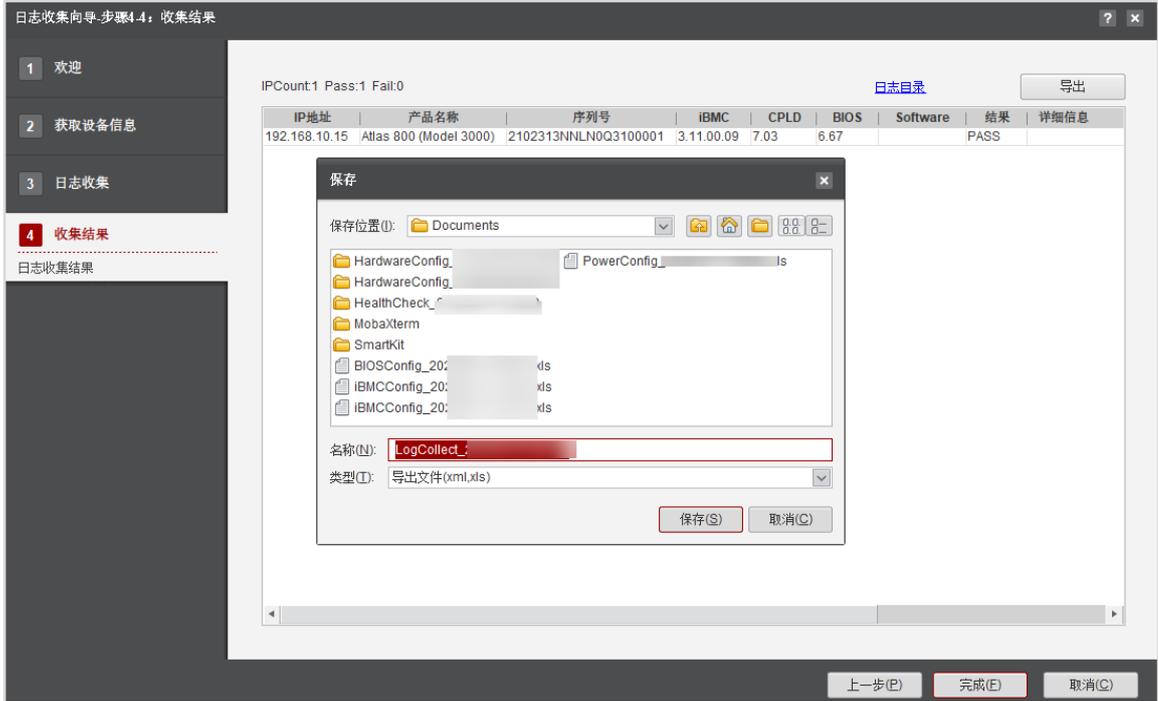




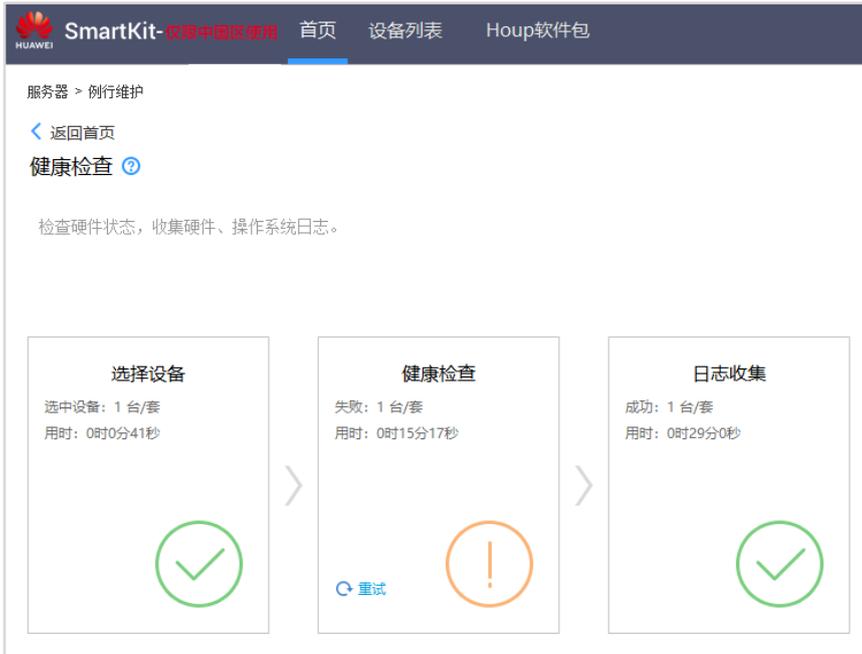
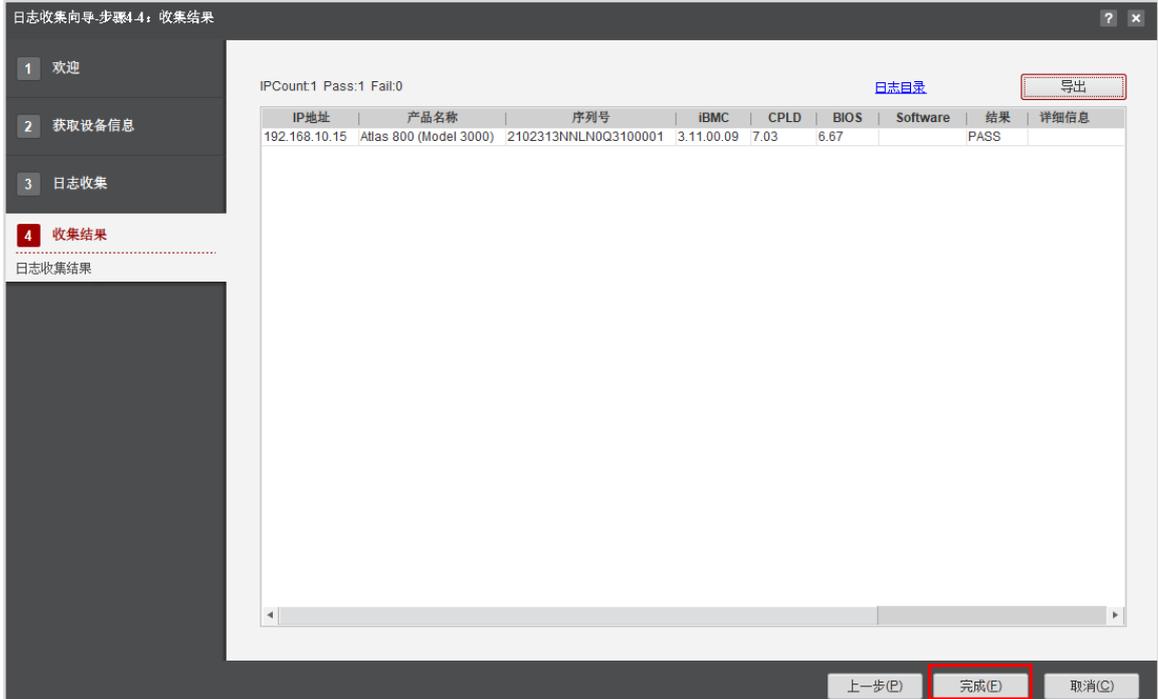
步骤 5 在“日志收集向导-步骤 4-4: 收集结果”界面，点击右上角“日志目录”，可查看采集到的日志压缩包。



步骤 6 点击右上角“导出”，可把日志收集任务导出到本地。



步骤 7 点击右下角“完成”，结束日志收集任务，系统会自动返回“健康检查”界面。



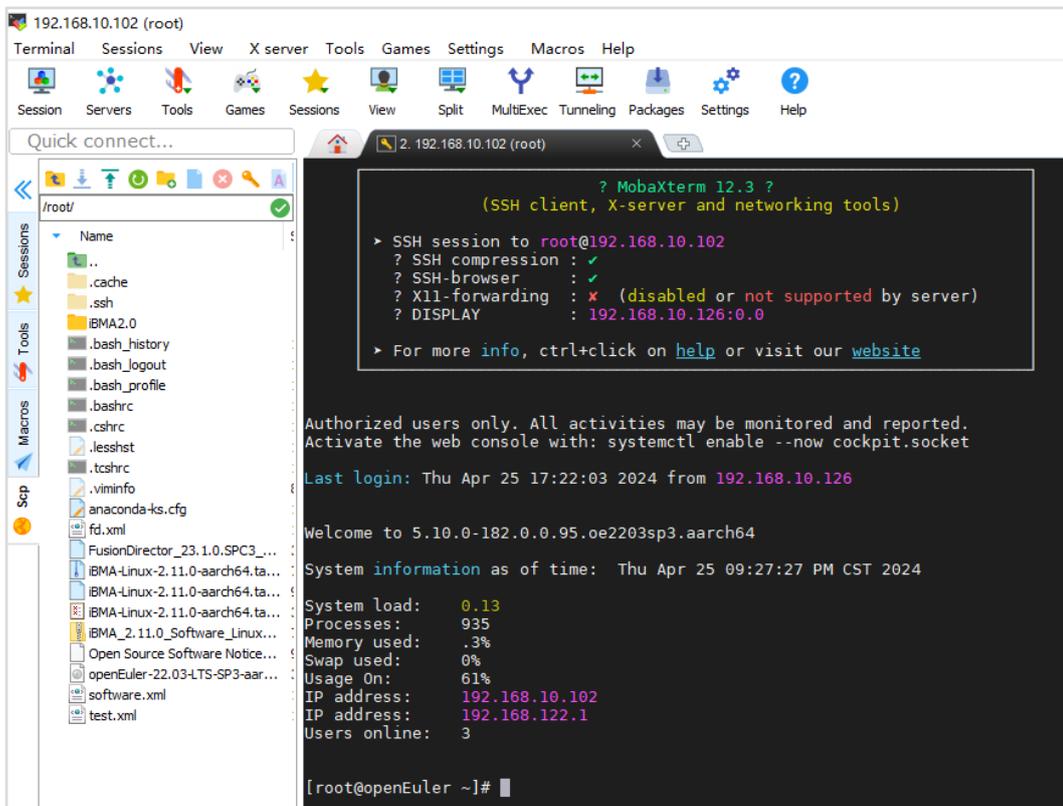
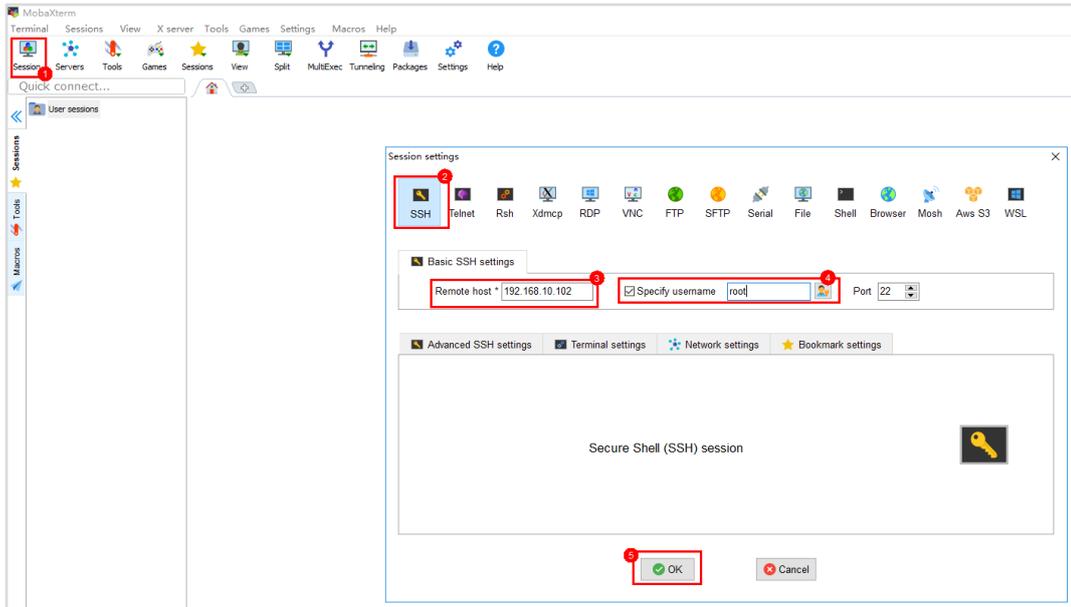
## 2.4 FusionDirector 安装与使用

### 2.4.1 安装 FusionDirector

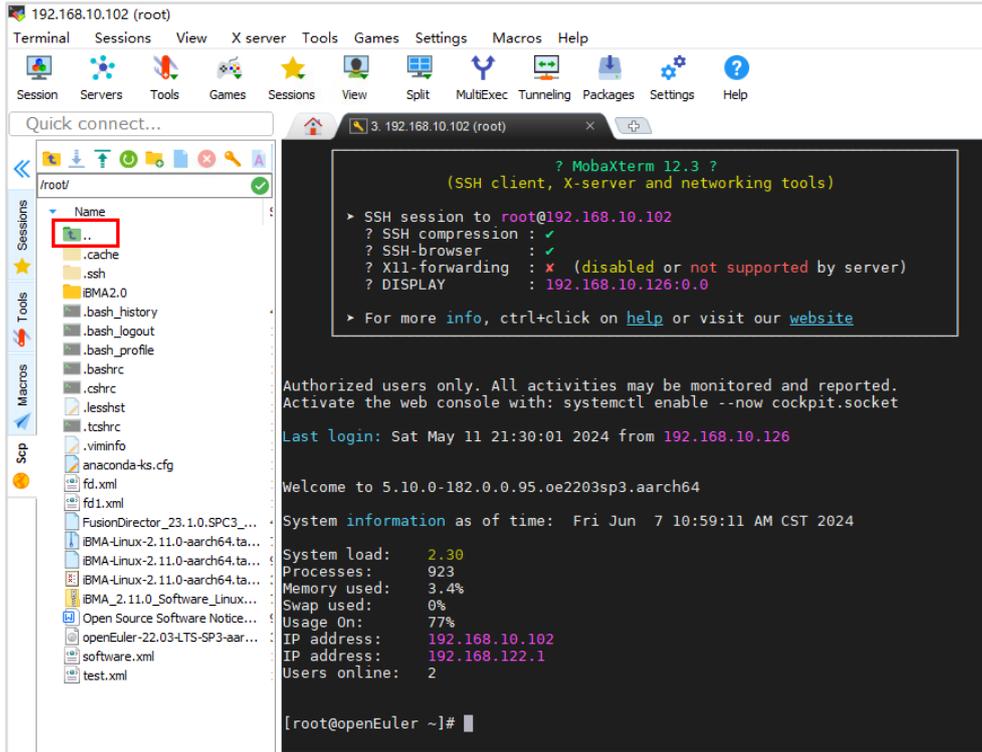
步骤 1 获取 FusionDirector 镜像包。



步骤 2 使用 MobaXterm 工具 SSH 登录到服务器操作系统后台。



步骤 3 使用 MobaXterm 工具 FTP 上传 FusionDirector 镜像包到 “/home” 目录下。



```
192.168.10.102 (root)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
3. 192.168.10.102 (root)
Name
.cache
.ssh
BMA2.0
.bash_history
.bash_logout
.bash_profile
.bashrc
.cshrc
.flesshst
.tcshrc
.viminfo
anaconda-ks.cfg
fd.xml
fd1.xml
FusionDirector_23.1.0.SPC3...
BMA-Linux-2.11.0-aarch64.ta...
BMA-Linux-2.11.0-aarch64.ta...
BMA-Linux-2.11.0-aarch64.ta...
BMA_2.11.0_Software_Linux...
Open Source Software Notice...
openEuler-22.03-LTS-SP3-aar...
software.xml
test.xml

? MobaXterm 12.3 ?
(SSh client, X-server and networking tools)
> SSH session to root@192.168.10.102
? SSH compression : ✓
? SSH-browser : ✓
? X11-forwarding : ✗ (disabled or not supported by server)
? DISPLAY : 192.168.10.126:0.0
> For more info, ctrl+click on help or visit our website

Authorized users only. All activities may be monitored and reported.
Activate the web console with: systemctl enable --now cockpit.socket

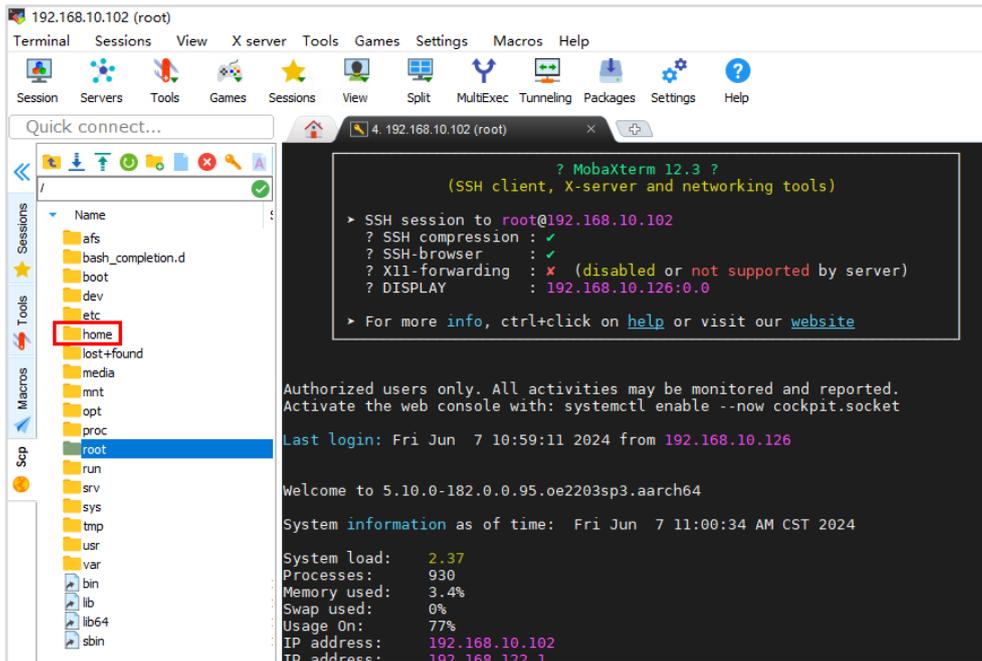
Last login: Sat May 11 21:30:01 2024 from 192.168.10.126

Welcome to 5.10.0-182.0.0.95.oe2203sp3.aarch64

System information as of time: Fri Jun 7 10:59:11 AM CST 2024

System load: 2.30
Processes: 923
Memory used: 3.4%
Swap used: 0%
Usage On: 77%
IP address: 192.168.10.102
IP address: 192.168.122.1
Users online: 2

[root@openEuler ~]#
```



```
192.168.10.102 (root)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
4. 192.168.10.102 (root)
Name
afs
bash_completion.d
boot
dev
etc
home
lost+found
media
mnt
opt
proc
root
run
srv
sys
tmp
usr
var
bin
lib
lib64
sbin

? MobaXterm 12.3 ?
(SSh client, X-server and networking tools)
> SSH session to root@192.168.10.102
? SSH compression : ✓
? SSH-browser : ✓
? X11-forwarding : ✗ (disabled or not supported by server)
? DISPLAY : 192.168.10.126:0.0
> For more info, ctrl+click on help or visit our website

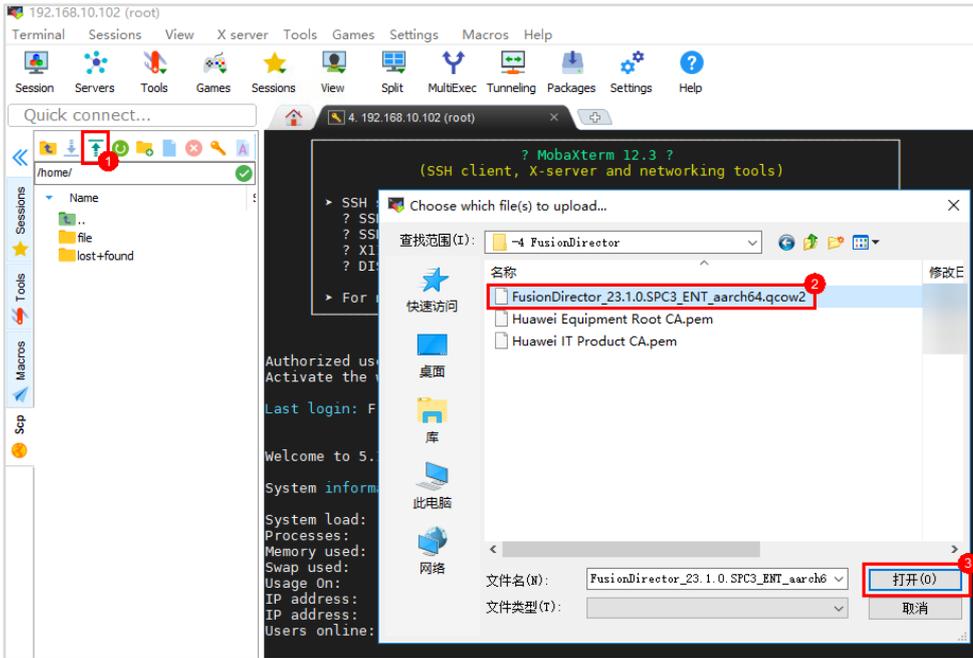
Authorized users only. All activities may be monitored and reported.
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Jun 7 10:59:11 2024 from 192.168.10.126

Welcome to 5.10.0-182.0.0.95.oe2203sp3.aarch64

System information as of time: Fri Jun 7 11:00:34 AM CST 2024

System load: 2.37
Processes: 930
Memory used: 3.4%
Swap used: 0%
Usage On: 77%
IP address: 192.168.10.102
IP address: 192.168.122.1
```



步骤 4 等待镜像包上传完毕，查看该镜像。

```
cd /home
ll
```

```
[root@openEuler ~]# cd /home
[root@openEuler home]# ll
total 2002200
drwxr-xr-x. 2 root root      4096  FusionDirector_23.1.0.SPC3_ENT_aarch64.qcow2 file
-rw-r--r--. 1 root root 2050228224  lost+found
drwx-----. 2 root root     16384
```

步骤 5 记录 FusionDirector 镜像的路径。

```
pwd
```

```
[root@openEuler home]# ll FusionDirector_23.1.0.SPC3_ENT_aarch64.qcow2
-rw-r--r--. 1 root root 2050228224  FusionDirector_23.1.0.SPC3_ENT_aarch64.qcow2
[root@openEuler home]# pwd
/home
[root@openEuler home]#
```

步骤 6 记录服务器未承载 IP 及状态 UP 的网卡名称。

```
ip a
```

```
[root@openEuler ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp125s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether b0:4f:a6:5e:f8:ce brd ff:ff:ff:ff:ff:ff
   inet 192.168.10.102/24 brd 192.168.10.255 scope global noprefixroute enp125s0f0
       valid_lft forever preferred_lft forever
   inet6 fe80::b24f:a6ff:fe5e:f8ce/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp125s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether b0:4f:a6:5e:f8:cf brd ff:ff:ff:ff:ff:ff
4: enp125s0f2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
   link/ether b0:4f:a6:5e:f8:d0 brd ff:ff:ff:ff:ff:ff
5: enp125s0f3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
   link/ether b0:4f:a6:5e:f8:d1 brd ff:ff:ff:ff:ff:ff
6: enp189s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether b0:4f:a6:5e:f8:d2 brd ff:ff:ff:ff:ff:ff
7: enp189s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether b0:4f:a6:5e:f8:d3 brd ff:ff:ff:ff:ff:ff
8: enp189s0f2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether b0:4f:a6:5e:f8:d4 brd ff:ff:ff:ff:ff:ff
9: enp189s0f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether b0:4f:a6:5e:f8:d5 brd ff:ff:ff:ff:ff:ff
10: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
   link/ether 52:54:00:9a:01:4f brd ff:ff:ff:ff:ff:ff
   inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
11: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
   link/ether 52:54:00:9a:01:4f brd ff:ff:ff:ff:ff:ff
12: veth: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 128
   link/ether 9c:7d:a3:28:6f:f9 brd ff:ff:ff:ff:ff:ff
   inet6 fe80::9e7d:a3ff:fe28:6ff9/64 scope link
       valid_lft forever preferred_lft forever
24: macvtap0@enp125s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 500
   link/ether 52:54:00:c3:23:28 brd ff:ff:ff:ff:ff:ff
   inet6 fe80::5054:ff:fec3:2328/64 scope link
       valid_lft forever preferred_lft forever
[root@openEuler ~]#
```

【注】这里选择 enp125s0f1，也可以挑选其他网卡。

步骤 7 新建 fd.xml 文件，并把如下内容填写到该文件中。

```
vim fd.xml
```

【注】使用 vim 命令可以直接创建新文件并进入编辑模式。

```
<domain type='kvm'>
  <name>FusionDirector</name>
  <memory unit='KiB'>16777216</memory>
  <currentMemory unit='KiB'>16777216</currentMemory>
  <vcpu placement='static'>8</vcpu>
  <os>
    <type arch='aarch64' machine='virt'>hvm</type>
    <loader readonly='yes' type='pflash'>/usr/share/edk2/aarch64/QEMU_EFI-pflash.raw</loader>
    <nvram>/var/lib/libvirt/qemu/nvram/FusionDirector.fd</nvram>
    <boot dev='hd'>
  </os>
  <features>
    <acpi/>
    <gic version='3'>
  </features>
  <cpu mode='host-passthrough' check='none'>
  <clock offset='utc'>
  <on_poweroff>restart</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
```

```

<devices>
  <emulator>/usr/libexec/qemu-kvm</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2'/>
    <source file='/home/FusionDirector_23.1.0.SPC3_ENT_aarch64.qcow2'/>
    <target dev='sda' bus='virtio'/>
  </disk>
  <controller type='usb' index='0' model='qemu-xhci' ports='8'>
    <address type='pci' domain='0x0000' bus='0x02' slot='0x00' function='0x00'/>
  </controller>
  <interface type='direct' trustGuestRxFilters='yes'>
    <source dev='enp125s0f1' mode='bridge'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x00'/>
  </interface>
  <console type='pty'>
    <target type='serial' port='0'/>
  </console>
</devices>
</domain>
    
```

【注】加粗的 2 处，根据步骤 5 和步骤 6 查询到的实际情况进行更换。

步骤 8 执行以下命令，定义虚拟机安装规格和镜像路径。

```
virsh define fd.xml
```

```

[root@openEuler ~]# virsh define fd.xml
Domain FusionDirector defined from fd.xml

[root@openEuler ~]# virsh list
 Id   Name      State
-----
-

[root@openEuler ~]# virsh list --all
 Id   Name      State
-----
-    FusionDirector  shut off

[root@openEuler ~]# █
    
```

执行以下命令，打开虚拟机。

```
virsh start FusionDirector
```

```

[root@openEuler ~]# virsh start FusionDirector
Domain FusionDirector started

[root@openEuler ~]# virsh list
 Id   Name      State
-----
 1    FusionDirector  running

[root@openEuler ~]# █
    
```

步骤 9 执行以下命令，打开虚拟机控制台，查看安装过程和安装状态。

```
virsh console FusionDirector
```

```

GNU GRUB  version 2.06

-----
*EulerOS_Server_V200R012C00SPC303B680
-----

Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
The highlighted entry will be executed automatically in 4s.
    
```

```

Booting `EulerOS_Server_V200R012C00SPC303B680'

Loading Linux 5.10.0-136.12.0.86.h1526.eulerosv2r12.aarch64 ...
Loading initial ramdisk ...
[ 0.001398][ T1] CrossPartition: IPI: swapper/0[1]: manage section has not been initialized!
[ 0.693853][ T1] SDEI NMI watchdog: Disable SDEI NMI Watchdog in VM

Authorized users only. All activities may be monitored and reported.
FusionDirector login: █
    
```

【注】

- 如果 virsh start 命令之后，没有立即运行 virsh console 命令，不一定看到安装过程。
- 如果 virsh console 后，没有出现 login 的回显，需要键盘按一下回车。

步骤 10 使用 root 用户登录 FusionDirector 后台，第一次登录需要修改 root 密码。

```

Booting `EulerOS_Server_V200R012C00SPC303B680'

Loading Linux 5.10.0-136.12.0.86.h1526.eulerosv2r12.aarch64 ...
Loading initial ramdisk ...
[ 0.001414][ T1] CrossPartition: IPI: swapper/0[1]: manage section has not been initialized!
[ 0.707627][ T1] SDEI NMI watchdog: Disable SDEI NMI Watchdog in VM

Authorized users only. All activities may be monitored and reported.
FusionDirector login: root
Password:
You are required to change your password immediately (administrator enforced).
Changing password for root.
Current password: login: timed out after 60 seconds

Authorized users only. All activities may be monitored and reported.
FusionDirector login: root
Password:
You are required to change your password immediately (administrator enforced).
Changing password for root.
Current password:
New password:
Retype new password:

Authorized users only. All activities may be monitored and reported.
FusionDirector:~ # █
    
```

【说明】以“root”用户登录后台的默认密码为“manager0(FD)”。

步骤 11 执行以下命令，查看 FusionDirector 虚拟机自动获取的 IP 地址。

```
ip a | grep eth0
```

```
FusionDirector:~ # ip a | grep eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc htb state UP group default qlen 1000
   inet 192.168.10.121/24 brd 192.168.10.255 scope global dynamic eth0
153: veth@eaa351@if152: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker_gwbridge state UP group default
201: veth@e3f41b@if200: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker_gwbridge state UP group default
```

【注】如果虚拟机未自动获取 IP 地址，或者自己规划了 FusionDirector 的 IP 地址，可以执行以下命令手动添加 IP 地址。

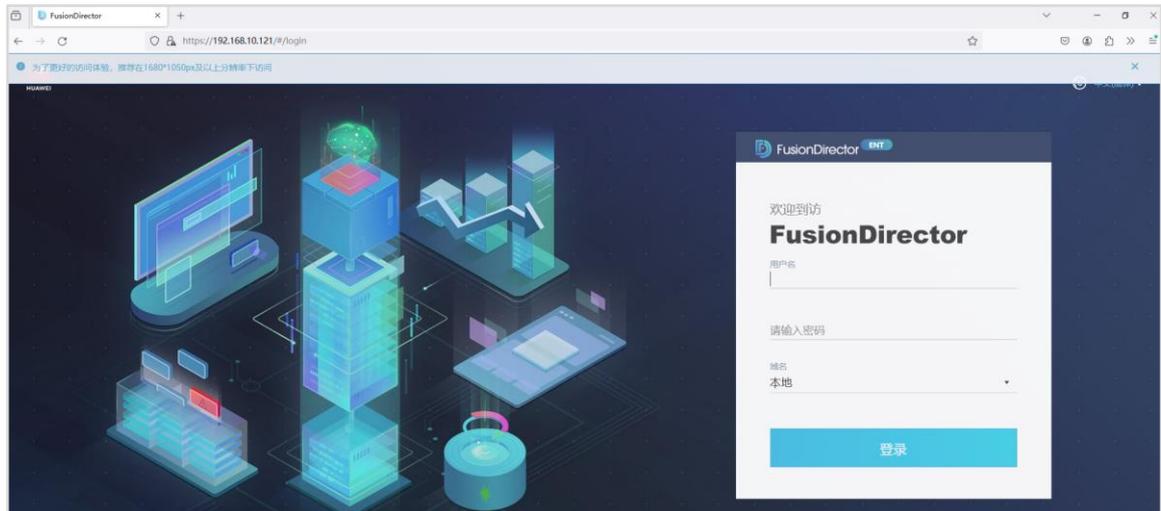
```
ip a add #ipaddress/24 dev eth0
```

```
ip r add default via #Gateway
```

- ipaddress: 表示 FusionDirector 的 IP 地址，如：192.168.10.121。
- Gateway: 表示 FusionDirector 所在网关地址，如：192.168.10.1。

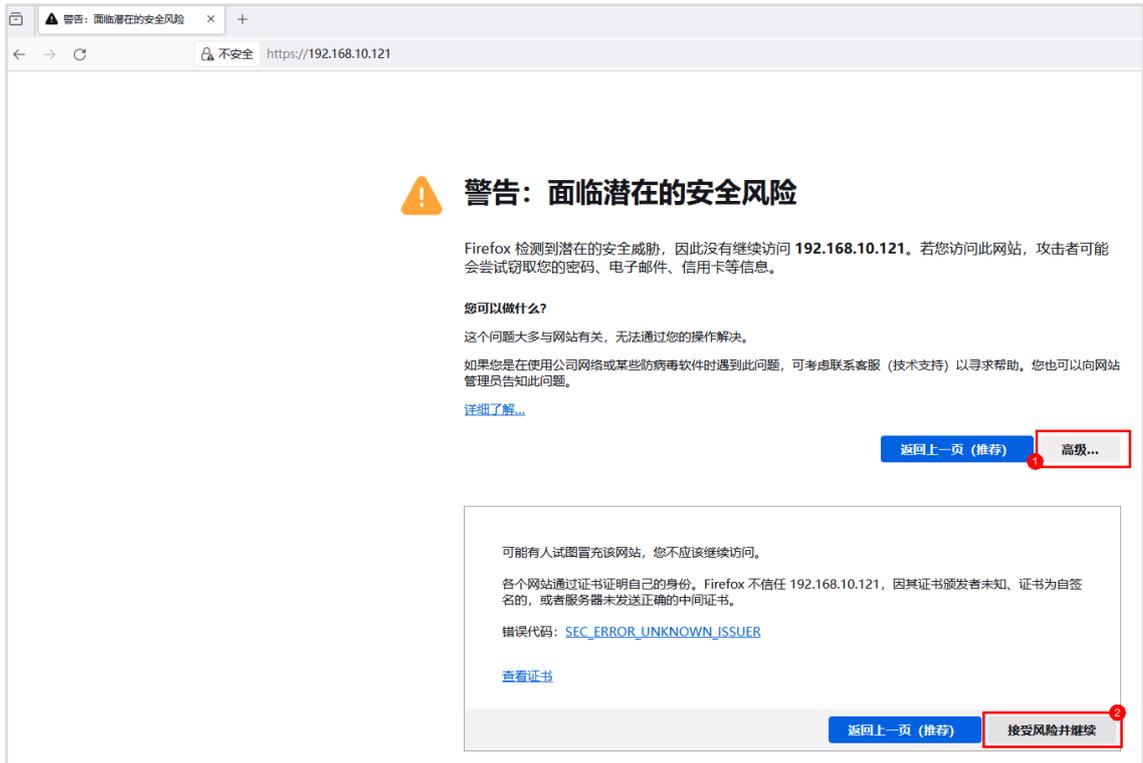
## 2.4.2 初始化 FusionDirector

步骤 1 安装完成后，等待 10 分钟左右的时间，后台自动启动相关 FusionDirector 服务，然后在浏览器中输入“https://FusionDirector 的 IP 地址:443”并按“Enter”。进入 FusionDirector 登录界面。

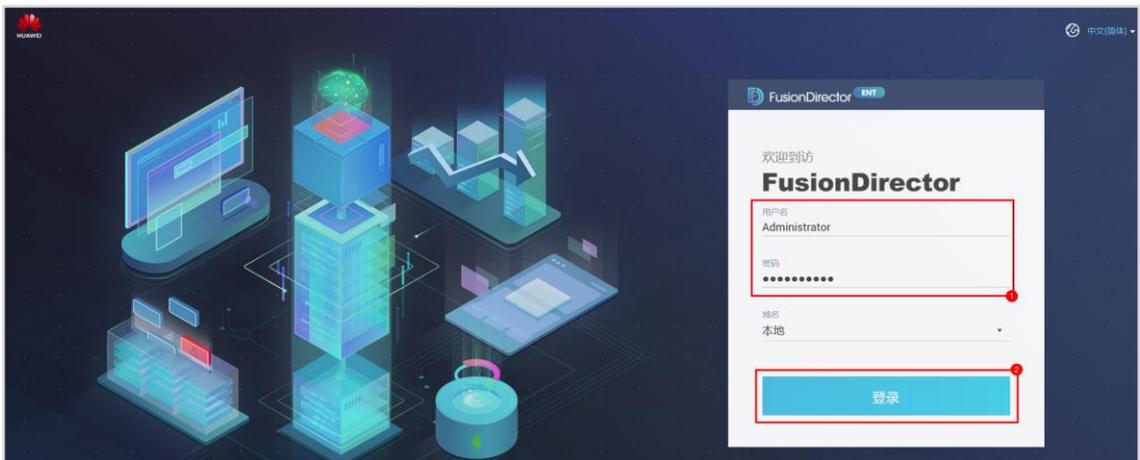


【注】

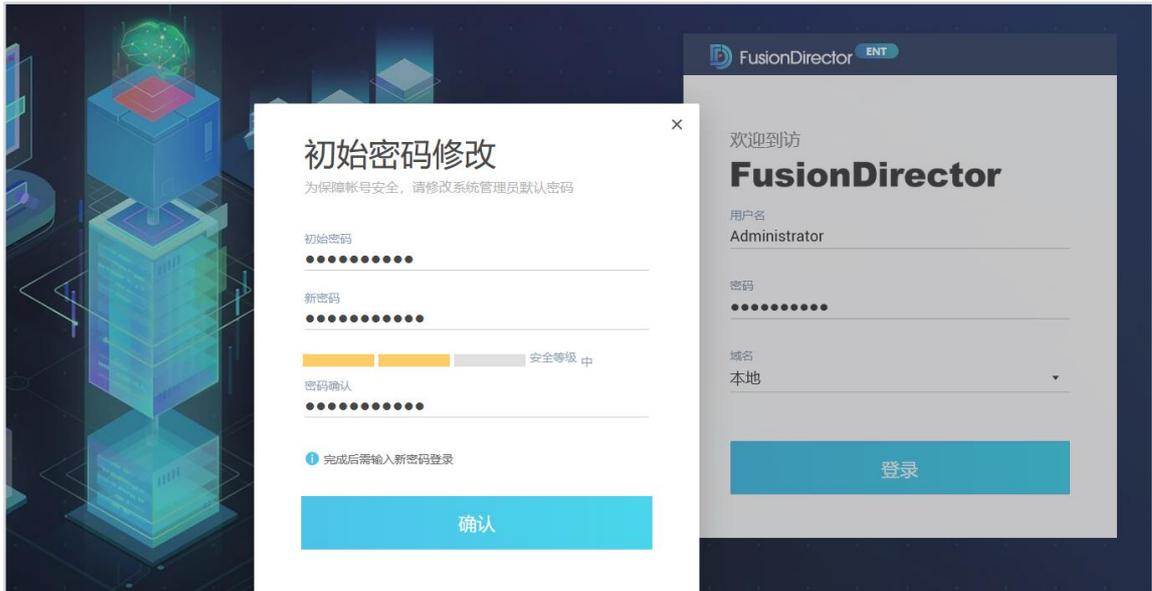
- 安装完成后，立马浏览器登录 FusionDirector 界面，网页会打不开，后台服务启动需要时间。
- 端口号默认为“443”，可以不用输入；首次在浏览器登录时，浏览器会有安全警告，按下图处理即可。



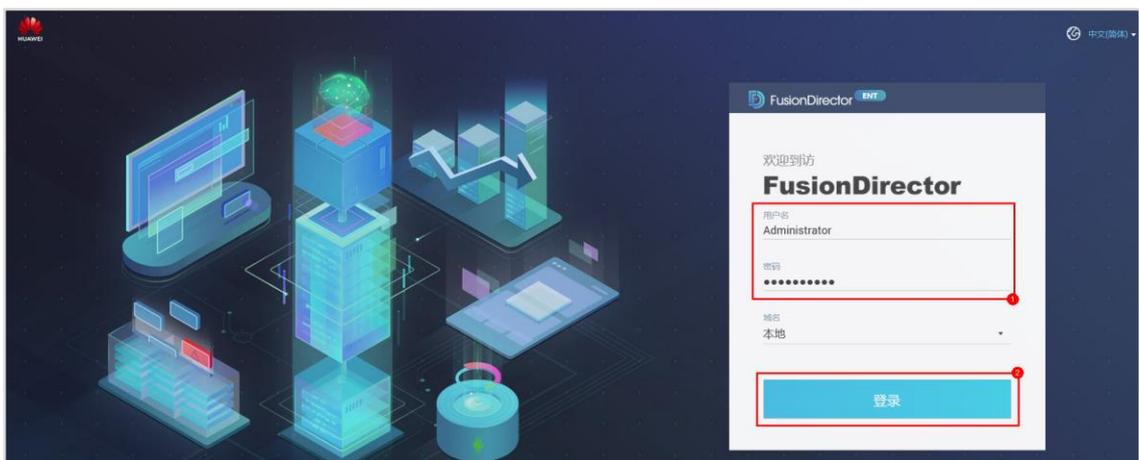
步骤 2 输入默认用户名“Administrator”和默认密码“Admin@9000”，单击“登录”。



步骤 3 根据界面提示输入“初始密码”、“新密码”以及“密码确认”，单击“确认”，修改 Administrator 的默认密码。

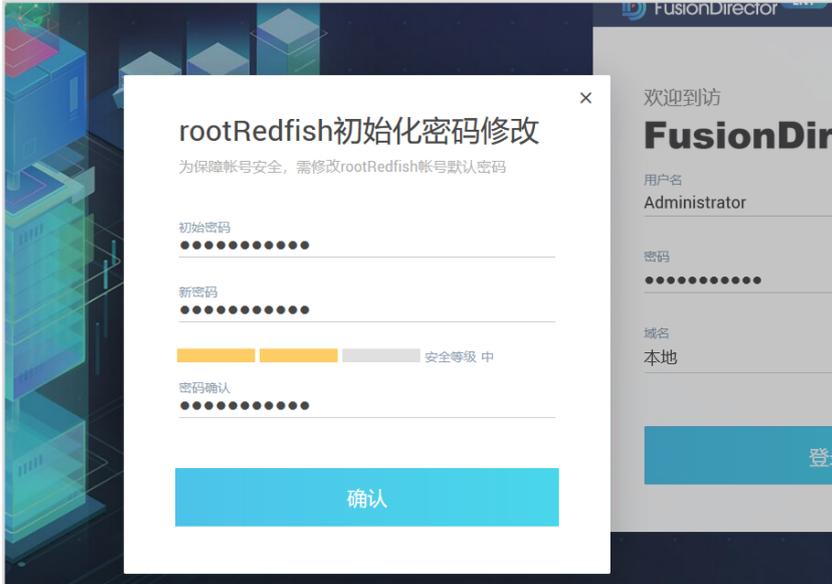


步骤 4 重新使用 Administrator 用户进行登录。



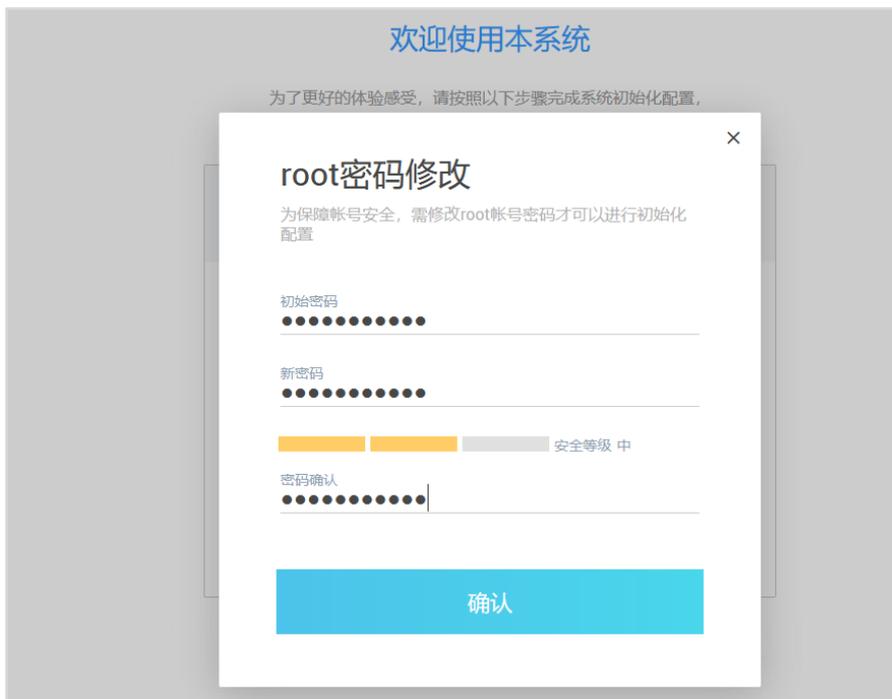
步骤 5 根据界面提示输入“初始密码”、“新密码”以及“密码确认”，单击“确认”，修改 rootRedfish 用户的默认密码。

【注】调用 REST API 的默认用户名为“rootRedfish”，默认密码为“Machine@123”。



步骤 6 根据界面提示输入“初始密码”、“新密码”以及“密码确认”，单击“确认”，修改 root 用户的密码。

【注】前面步骤在登录 FusionDirector 后台查看 IP 地址时，已经修改了 root 密码，这里的初始密码请填写修改后的密码。



步骤 7 在“配置网关类型”界面，选择“下级网管”，点击“下一步”。



步骤 8 在“配置主机名”界面，按下图进行配置，点击“下一步”。



步骤 9 在“配置网络”界面，按下图进行配置，点击“下一步”。

## 欢迎使用本系统

为了更好的体验感受，请按照以下步骤完成系统初始化配置，  
配置完成后可以开始体验本系统

### 3 配置网络

请设置系统外部访问IP地址信息

eth0 Up

\* IP地址:

192	.	168	.	10	.	121
-----	---	-----	---	----	---	-----

\* 子网掩码:

255	.	255	.	255	.	0
-----	---	-----	---	-----	---	---

\* 网关:

192	.	168	.	10	.	1
-----	---	-----	---	----	---	---

上一步
下一步

【注】这里也可以为 FusionDirector 虚拟机替换 eth0 的 IP，为了方便，本次实验不进行更换，还是填写前面步骤查询到的 IP 地址。

步骤 10 在“配置系统时间”界面，按下图进行配置，勾选许可协议，点击“立即体验”。

## 欢迎使用本系统

为了更好的体验感受，请按照以下步骤完成系统初始化配置，  
配置完成后可以开始体验本系统

### 4 配置系统时间

请设置系统时区和系统时间

\* 选择时区:

亚洲

(UTC+08:00) 上海市

\* 设置时间:

自

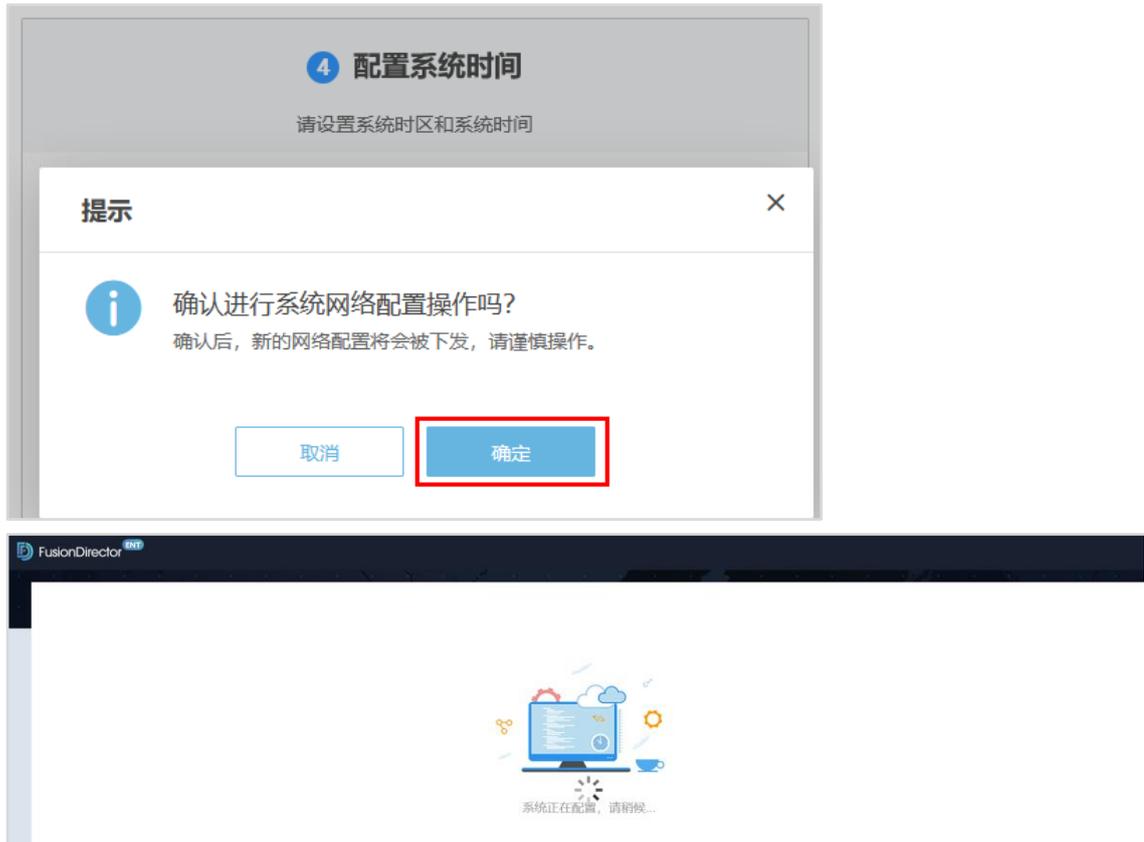
当前时区: (UTC+08:00) 亚洲/上海市

当前时间:

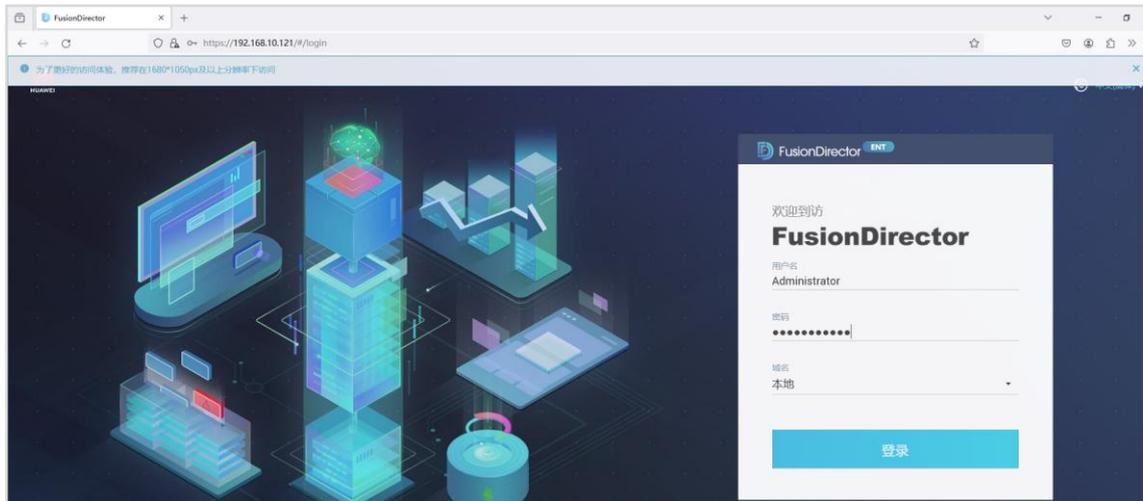
已阅读并接受 [《华为企业业务最终用户许可协议》](#)

上一步
立即体验

步骤 11 系统会弹出提示框，点击“确认”，系统会根据用户的设置进行配置下发。

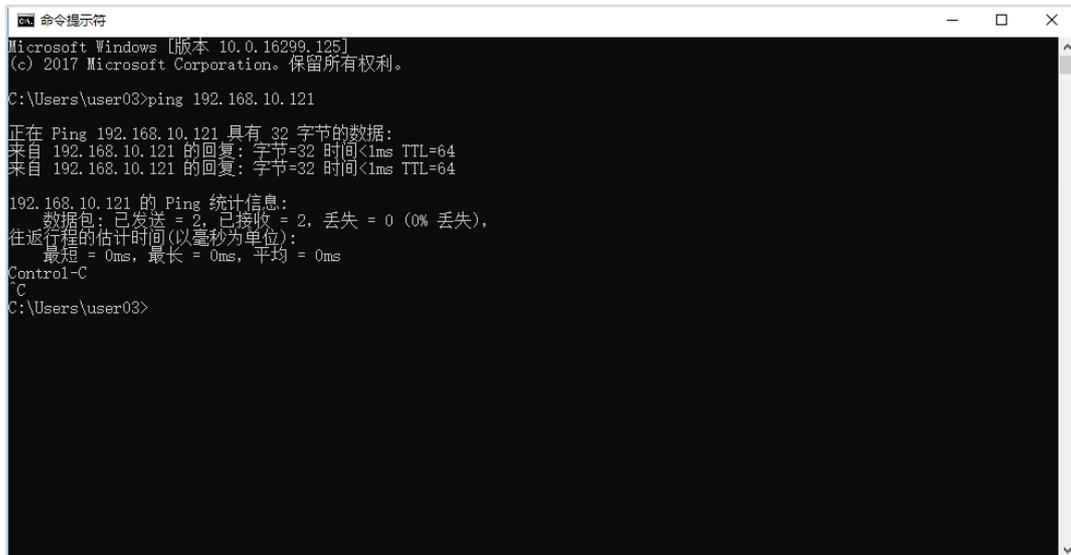


步骤 12 配置完毕后，刷新网页，重新登录 FusionDirector 的 webUI，进入 FusionDirector 主页。



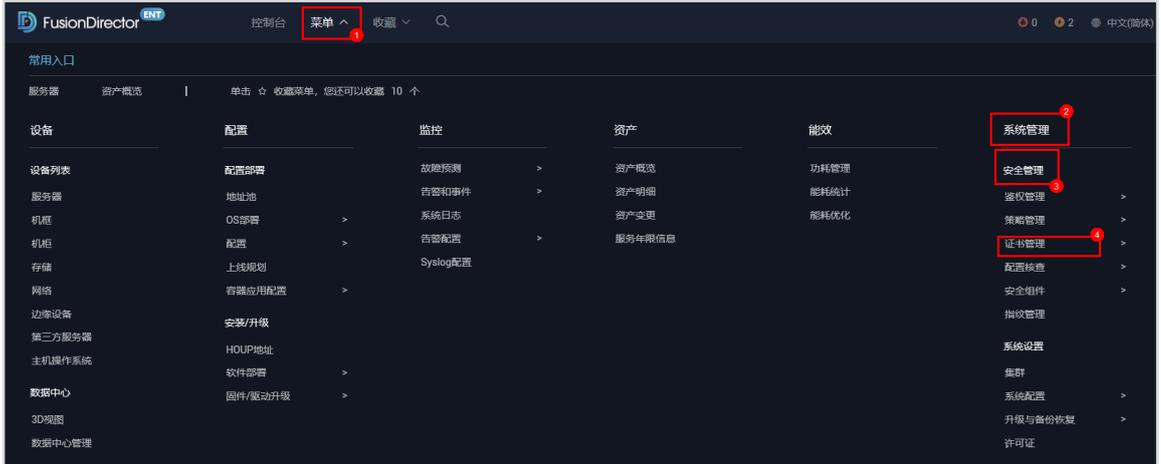


【注】如果刷新页面出现网络问题无法进入登录界面，请打开本地 PC 的 CMD，ping 一下 FusionDirector 的 IP 地址后，再重新刷新页面。



### 2.4.3 上传根证书及二级 CA 证书

步骤 1 在 FusionDirector 主页，点击“菜单 > 系统管理 > 安全管理 > 证书管理”。

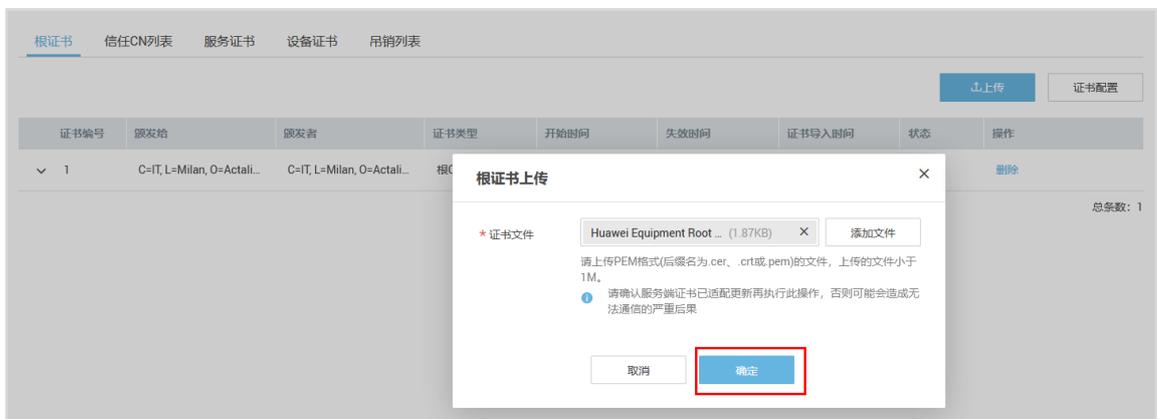
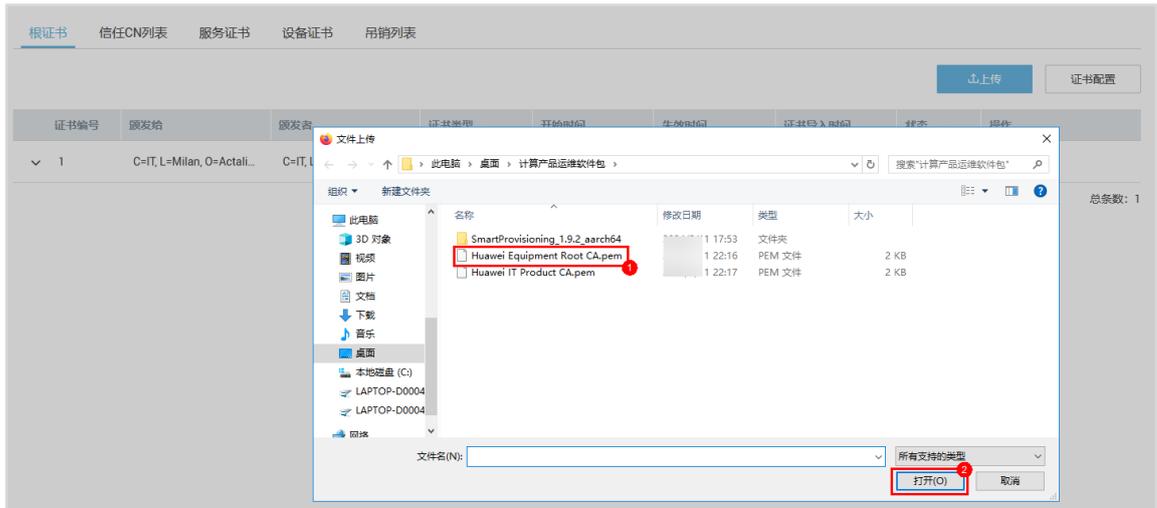


步骤 2 在根证书管理界面，点击“上传”。

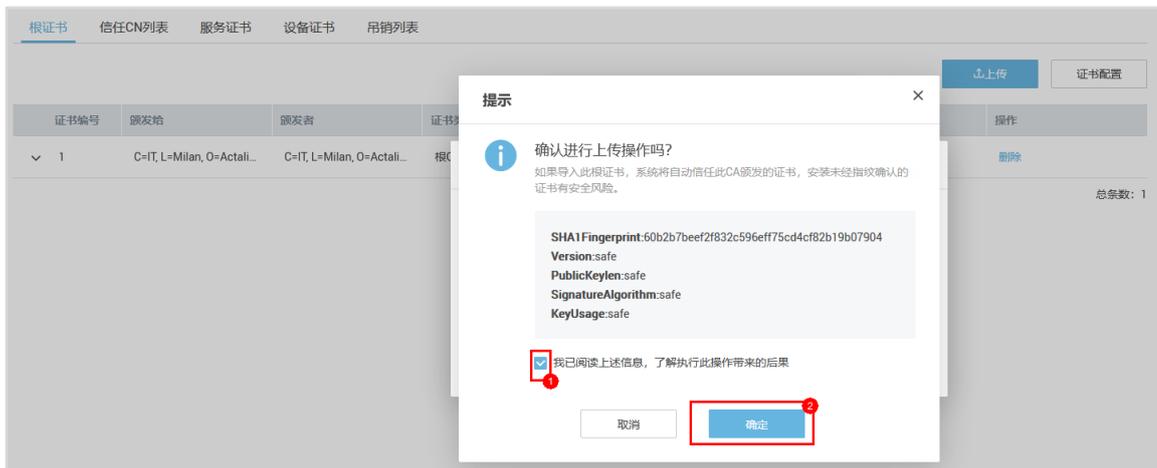


步骤 3 在“根证书上传”界面，点击“添加文件”，选择本地的“Huawei Equipment Root CA.pem”，点击“确定”。





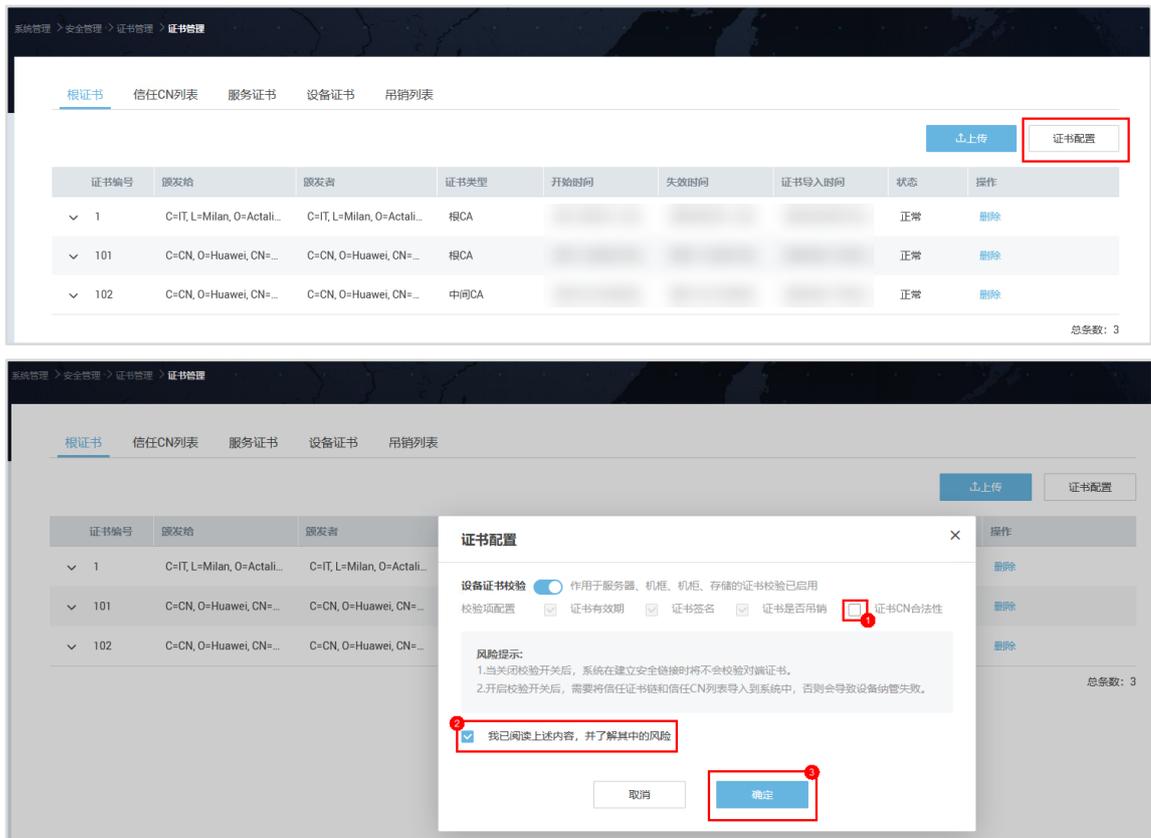
步骤 4 弹出提示窗，勾选协议，点击“确定”，华为设备的根证书上传完毕。



步骤 5 同样的步骤，上传“Huawei IT Product CA.pem”二级 CA 证书。



步骤 6 点击“证书配置”，取消勾选“证书 CN 合法性”，勾选协议，点击“确定”。



步骤 7 点击界面右上角头像图标，点击“退出”，并重新使用 Administrator 账号登录，使证书配置生效。

## 2.4.4 添加服务器

步骤 1 点击“菜单 > 设备 > 服务器”，进入服务器列表界面。



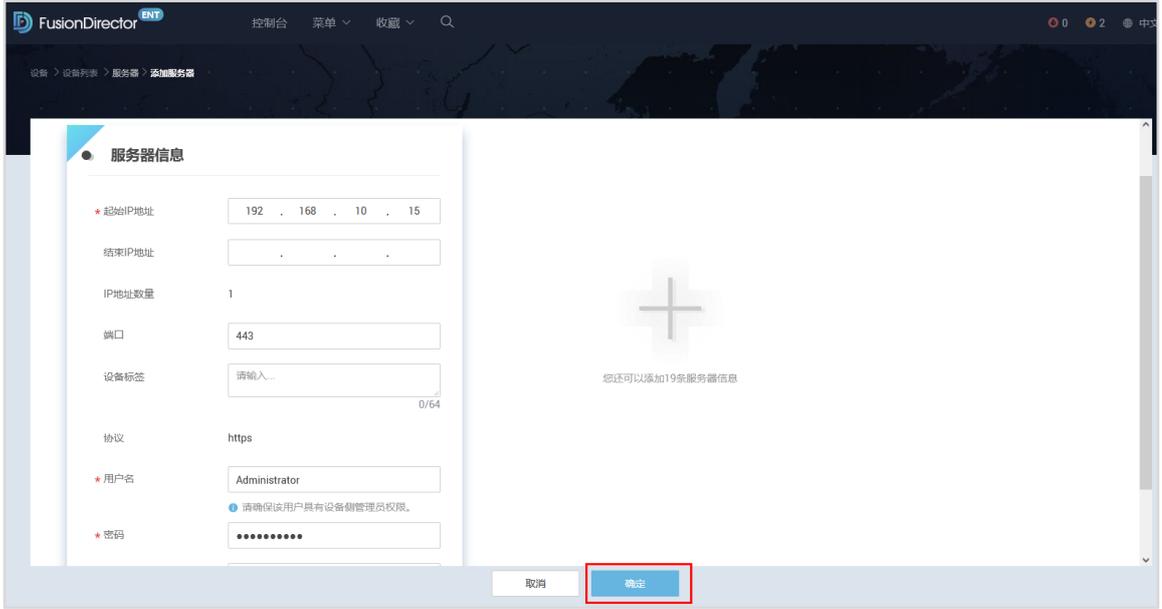
步骤 2 点击“添加服务器”，选择“手动添加”。



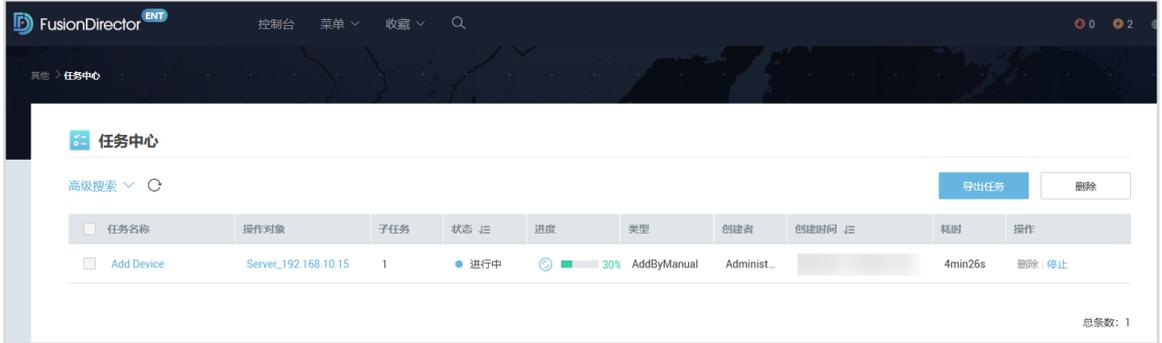
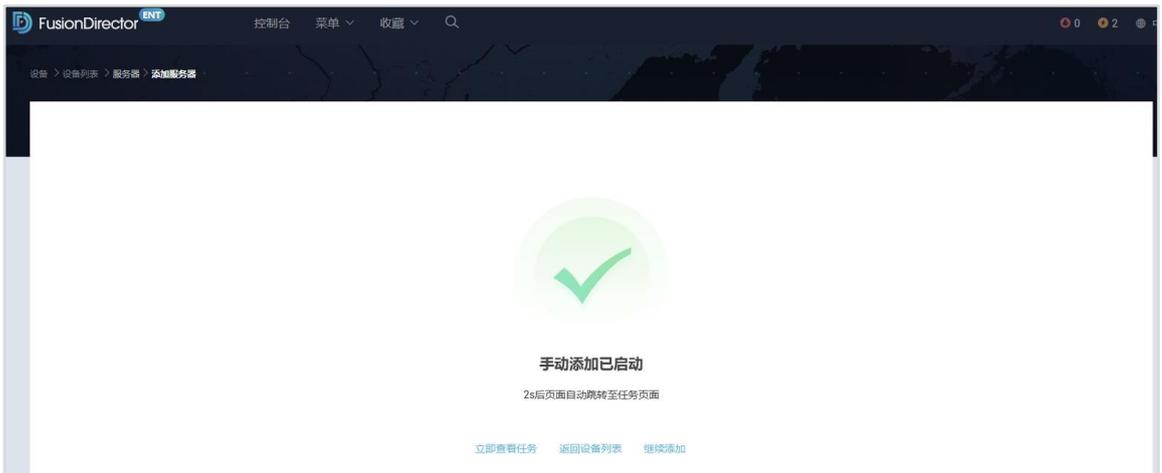
步骤 3 填写服务器信息，其他参数保持默认，点击“确定”。

【注】服务器示例信息如下，请根据实际环境信息进行填写。

- 起始 IP 地址：192.168.10.15
- 用户名：Administrator
- 密码：Admin@9000

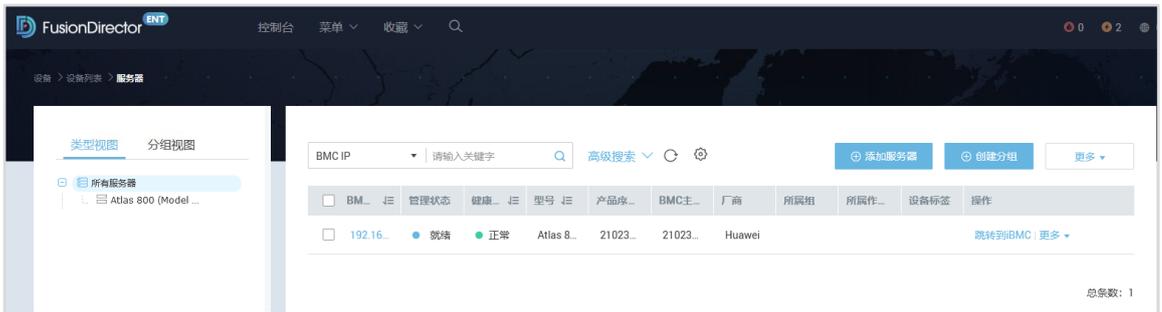


步骤 4 系统会自动跳转到任务中心，等待服务器添加进度为 100%。



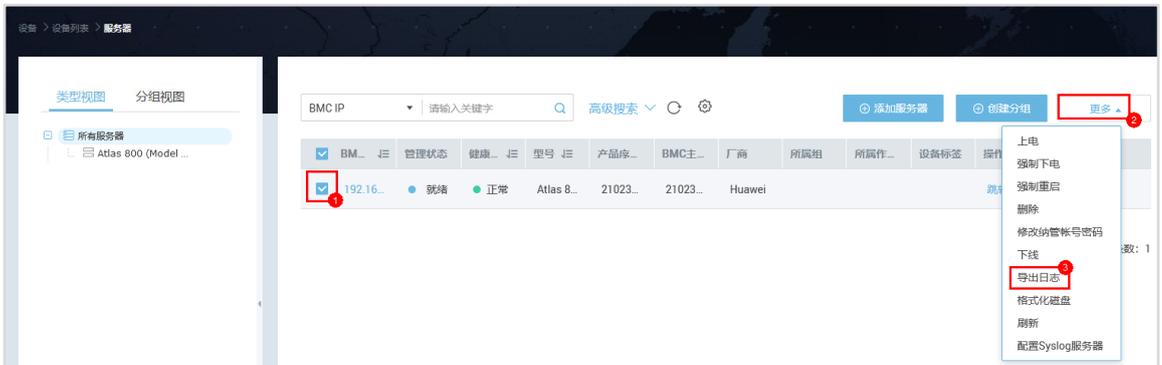


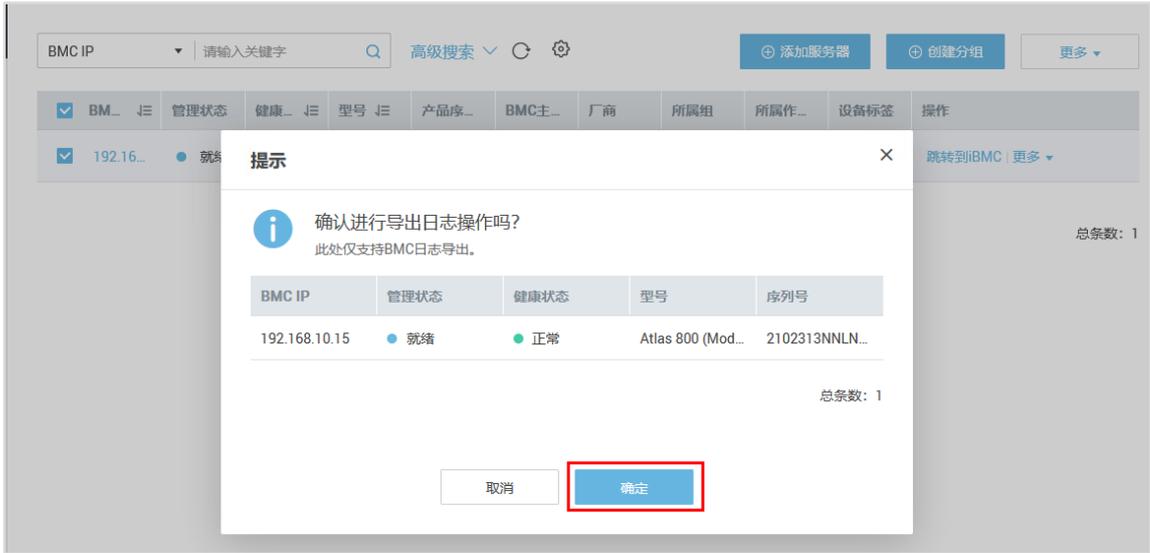
步骤 5 返回服务器列表，可以看到服务器添加已经完成。



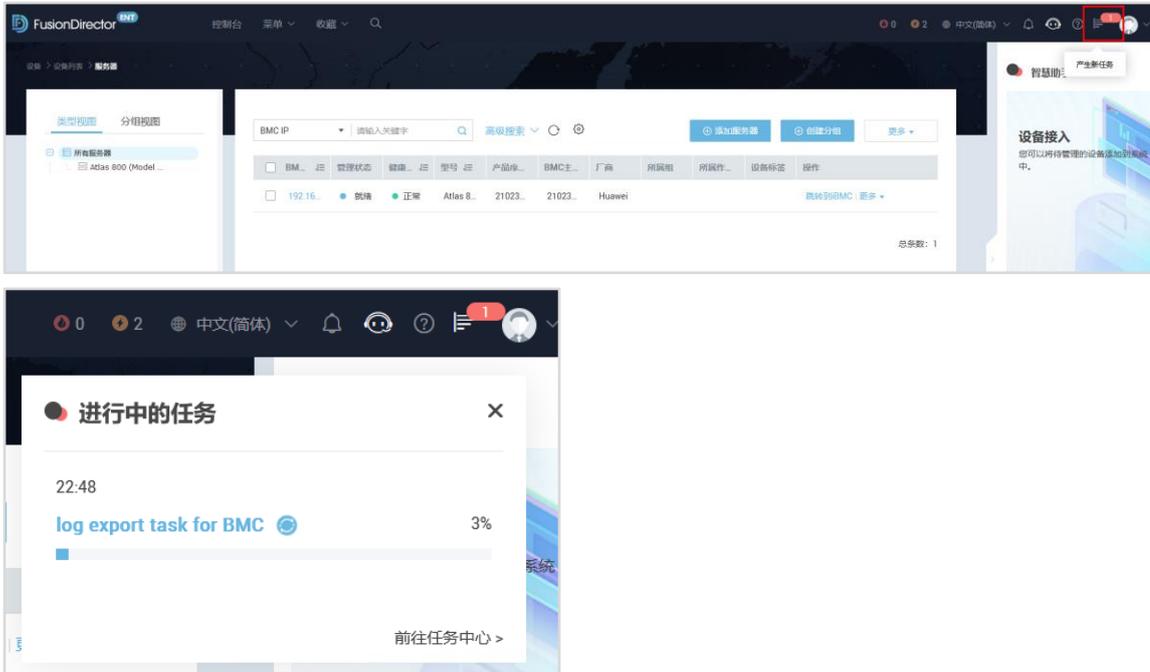
## 2.4.5 服务器基本操作

步骤 1 勾选服务器，点击“更多 > 导出日志”，弹出提示框，点击“确认”，可导出 BMC 日志。

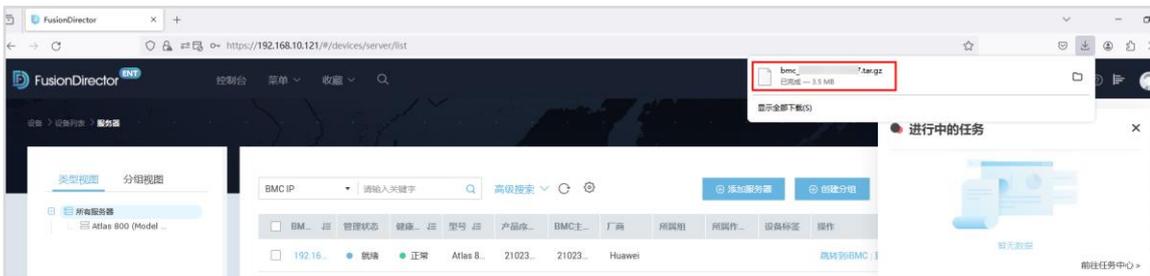




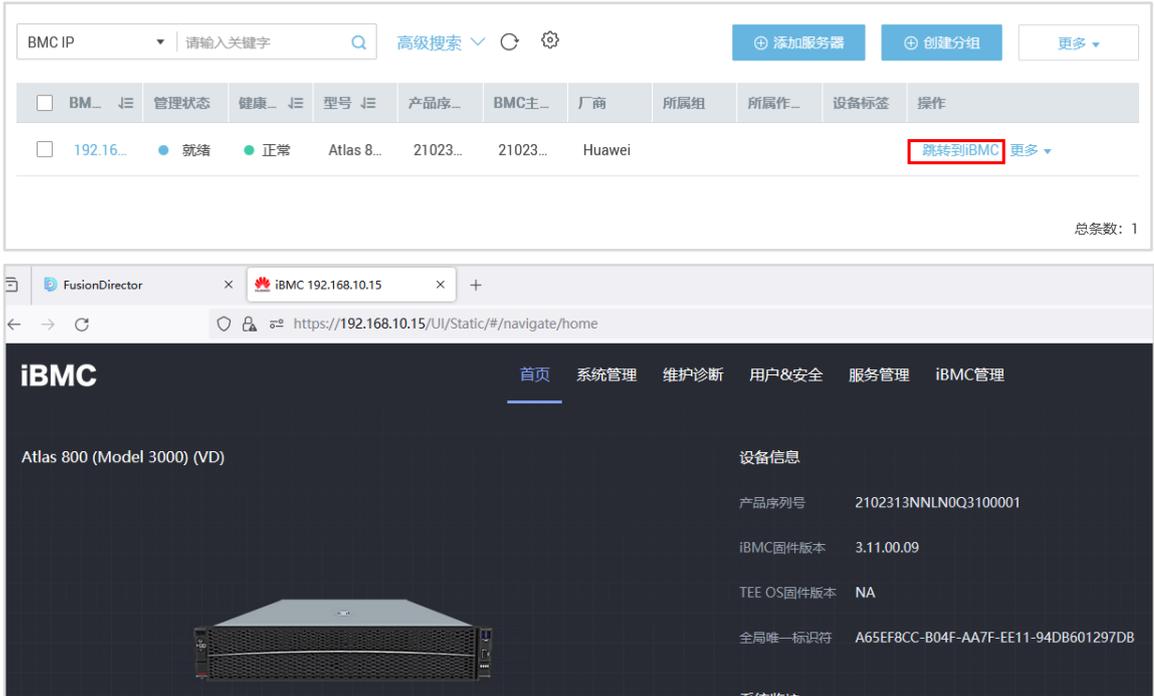
步骤 2 点击右上角任务中心图标，可查看 BMC 日志导出进度。



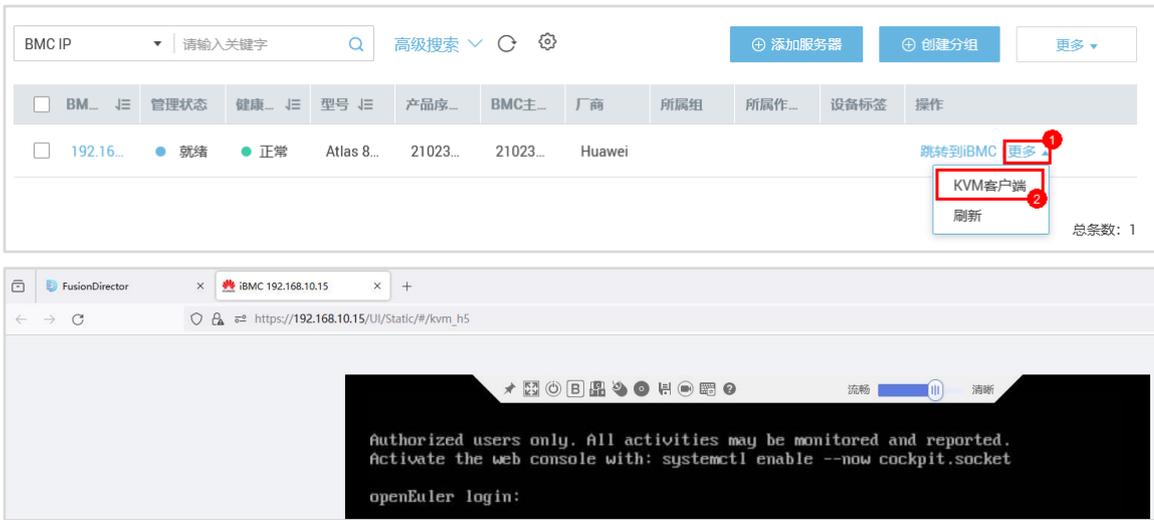
步骤 3 等待导出日志任务执行完毕，自动下载日志压缩包到本地。



步骤 4 点击服务器的“跳转到 iBMC”，可一键登录服务器的 iBMC WebUI。

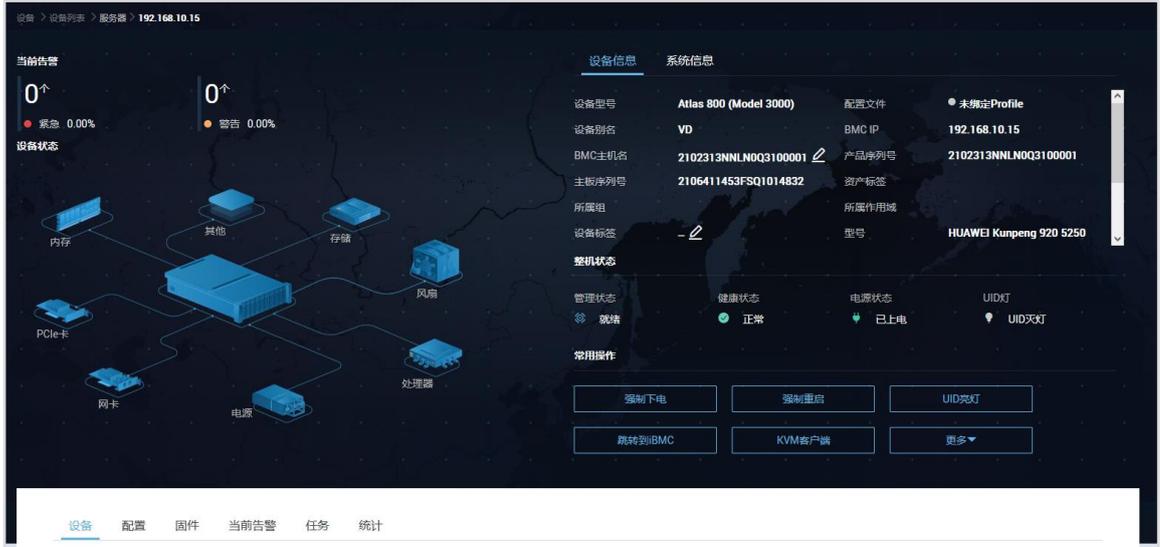


步骤 5 点击服务器的“更多 > KVM 客户端”，可一键进入虚拟控制台。



步骤 6 点击服务器 BMC IP，可查看服务器的相关信息。





## 2.4.6 服务器监控信息

步骤 1 点击“菜单 > 监控 > 故障预测 > 硬盘”，进入硬盘故障预测界面。



步骤 2 点击“开启”，弹出提示框，点击“确定”。





步骤 3 可查看硬盘的故障预测信息。



【注】刚开启没有相关数据，需要一定时间收集硬盘信息方可给出预测结果。

步骤 4 同样也可以进行内存故障预测。

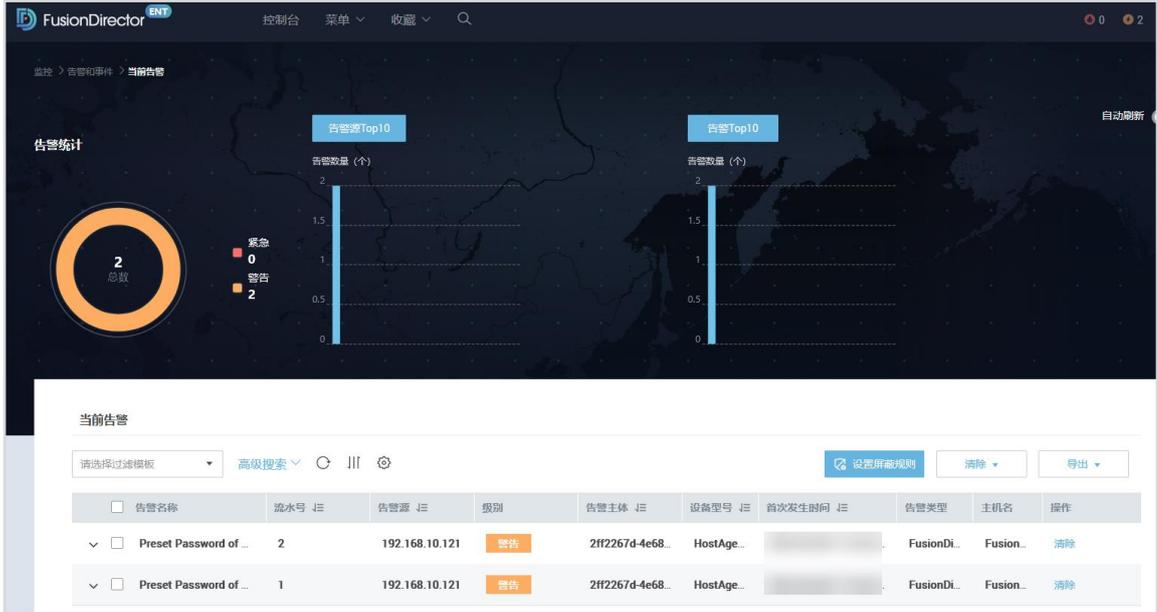




步骤 5 点击“菜单 > 告警和事件 > 当前告警”，进入告警界面。



步骤 6 可查看目前接入 FusionDirector 平台所有设备的告警。



步骤 7 点击“菜单 > 监控 > 系统日志”，进入系统日志界面。



步骤 8 可通过条件进行筛选日志，也可导出日志。

监控 > 系统日志

### 日志

高级搜索

日志类型:

日志级别:

操作结果:

时间筛选:

操作名称	操作对象	操作用户	日志级别	操作用户IP	操作结果	发生时间	详情
Login System	Appliance	Administrator	提示	192.168.10.126	成功		The Administrator user logs in to the syste...
Logout System	Appliance	System	提示	192.168.10.121	成功		The Administrator(192.168.10.126) user is ...
Set PFAService Switch	FaultPrediction	Administrator	严重	192.168.10.126	成功		Update PFAService switch success. descri...
KVM Login	Device	Administrator	提示	192.168.10.126	成功		The Administrator user logs in to 192.168.1...

# 3 思考题

---

1、随着大模型的火热应用，某企业项目采购了一批 Atlas 服务器作为大模型的训练集群，您是该项目的工程师，现在需要进行 Atlas 服务器操作系统的安装，您能想到几种安装 OS 的方式？

答案：

方式 1：登录 iBMC 界面，通过 VNC 方式挂载本地镜像文件进行 OS 安装。

方式 2：在 Smart Provisioning 里，通过 OS 部署功能进行 OS 安装。

方式 3：在 SmartKit 工具里，通过 OS 部署功能进行 OS 安装。

2、该项目已经顺利交付并进行转维，后续您在运维该批服务器时，遇到了一些问题需要进行故障处理，您能使用哪些工具辅助工具定位与处理？

答案：

iBMC、iBMA、Smart Provisioning、SmartKit、eService、FusionDirector 这些工具都具备辅助运维工程师进行故障处理。

---

# 4 缩略语

缩略语	英文全称	解释
iBMC	Intelligent Baseboard Management Controller	实现服务器管理的控制单元。
iBMA	Intelligent Baseboard Management Agent	服务器智能板级管理代理，华为 iBMA2.0 作为带内管理代理软件，对上层管理软件 iBMC 开放了标准的 Redfish 管理接口，且可以通过 BoB 映射支持与带外系统的融合，以提供更完整的服务器设备管理能力。
BIOS	Basic Input Output System	基本输入输出系统。一种相对较小的程序，驻留在服务器或类服务器设备上的可编程非易失性存储器中，负责引导该设备并执行某些独立于操作系统的 I/O 操作。
HMM	Hyper Management Module	超级管理模块。HMM 负责电源和风扇的集中管理，是服务器节点 iBMC 的管理延伸，主要负责电源和风扇健康状态监控、资产信息和实时数据获取、散热调速、网络汇聚。
RAID	Redundant Array of Independent Disks	独立磁盘冗余数组。一种把多块独立的硬盘（物理硬盘）按不同方式组合起来形成一个硬盘组（逻辑硬盘），从而提供比单个硬盘更高的存储性能和提供数据冗余的技术。
CA	Certificate Authority	证书颁发中心。
CN	Change Notice	变更通知。以 CN 为变更的发布单元，最小验证单元，可以使用 CN 进行配套发布。
KVM	Keyboard, Video, and Mouse	键盘/显示器/鼠标。KVM 是一款远程协助软件，通过 KVM 远程查看终端主机的屏幕。