

***msi***

# EdgeXpert 系列

小型服务器

MS-C931

用户指南

# 目录

快速入门.....	4
包装内容.....	4
安全与舒适提示.....	4
系统尺寸.....	5
系统概述.....	6
硬件安装.....	8
系统放置.....	9
系统堆叠方式.....	10
初始设置.....	11
什么是 NVIDIA DGX™ 操作系统.....	11
特色.....	11
首次启动设置.....	12
您要执行的操作流程.....	12
选择您的设置模式.....	12
准备事项.....	13
运行安装向导.....	13
快速入门.....	13
设置过程预期事项.....	14
系统堆叠.....	16
系统要求.....	16
系统间网络设置.....	16
运行系统发现脚本.....	17
安装必备软件并验证配置.....	18
两个系统 NCCL 配置指南.....	18
故障排除.....	21
升级 NVIDIA DGX™ 操作系统.....	22
重新安装 NVIDIA DGX™ 操作系统.....	22
创建可引导的 U 盘.....	22
启动 NVIDIA DGX™ OS ISO 映像.....	22

## 修订

V1.1, 2025/11

NVIDIA Sync .....	23
安装 .....	23
支持的应用程序 .....	23
其他连接方法 .....	23
DGX™ 动态面板 .....	24
集成 JupyterLab .....	24
访问动态面板 .....	25
NVIDIA Docker 容器运行时 .....	25
选择性配置:将用户添加至 Docker 群组 .....	26
使用指南 .....	26
验证 .....	27
故障排除 .....	27
NGC .....	28
快速入门 .....	29
基本用法 .....	30
常见工作流程 .....	30
最佳实践 .....	30
故障排除 .....	31
获取帮助 .....	31
从 NVIDIA 官方网站获取和激活 AI 模型 .....	32
固件更新 .....	32
建议方法 .....	32
手动更新方法 .....	33
故障排除 .....	33
其他资源 .....	33
安全注意事项 .....	34

# 快速入门

本章提供关于硬件安装程序的信息。在连接设备时,请小心固定设备并使用接地腕带以免产生静电。

## 包装内容物

小型服务器	MS-C931
文档	快速入门指南
配件	USB PD 适配器
	电源线



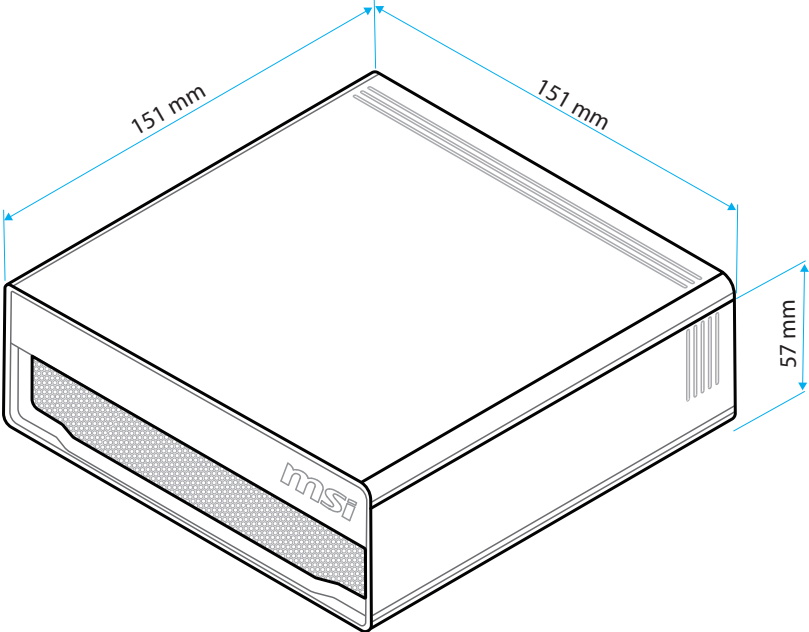
### 注意

- 如有任何物品损坏或缺失,请联系您的购买地或当地经销商。
- 包装内容物可能因国家/地区和型号而有所差异。
- 随附的电源线专用于此显示器,不应与其他产品一起使用。

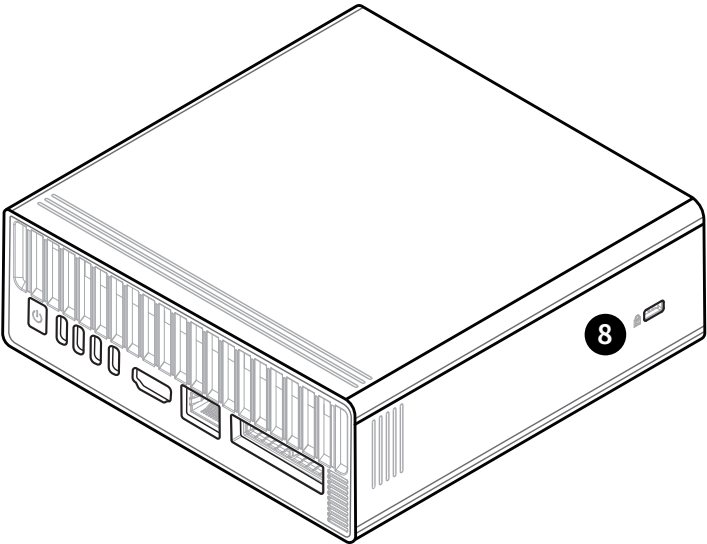
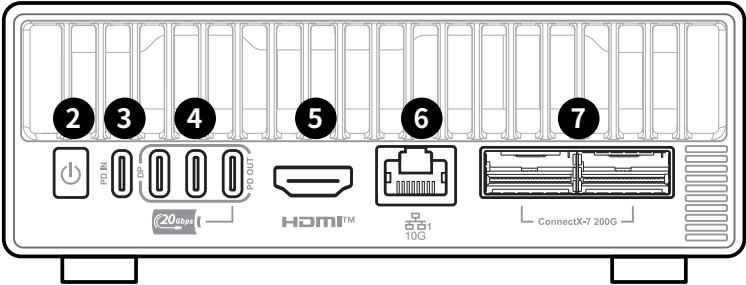
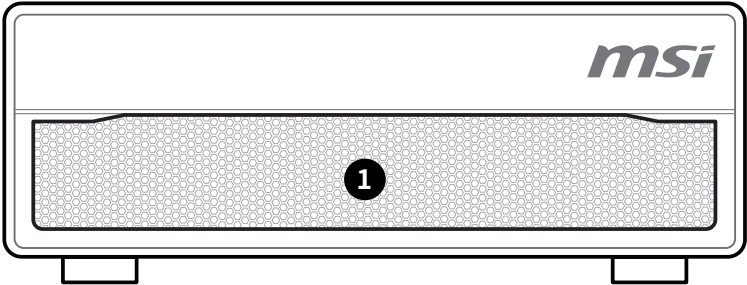
## 安全与舒适提示


- 如果您长期使用该设备,选择良好工作场所非常重要。
- 您的工作场所应具有充足的照明。
- 选择适当的工作台和座椅并调整高度,以符合您操作电脑时的坐姿。
- 当坐在椅子上时,请坐正并摆好姿势。调整椅背(若可能),让您的背部得到支撑并感觉舒适。
- 将双脚自然平放在地上,让膝盖和肘部在操作电脑时保持适当姿势(大约 90 度)。
- 将您的手自然放在桌面上以支撑您的手腕。
- 避免在可能造成身体不适的地方使用该设备(例如在床上)。
- 该设备属于电子设备。请小心使用,以免受伤。

# 系统尺寸



# 系统概述

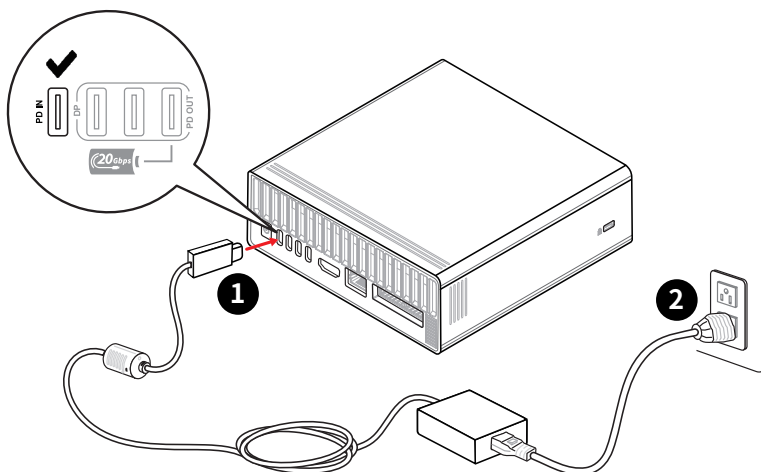


<p><b>1</b></p>	<p><b>通风孔</b> 通风孔用于空气流通,以防设备过热。请勿盖住通风孔。</p>
<p><b>2</b></p>	<p><b>电源按钮</b> 按下电源按钮打开或关闭系统。</p>
<p><b>3</b></p>	<p><b>电源插孔</b> 此插孔为系统提供电力供应。</p>
<p><b>4</b></p>	<p><b>USB 20Gbps Type-C 端口</b> 每个接口可提供高达 5V/3A 电源,三个接口同时使用时,总输出功率最高为 30W。</p>
<p><b>5</b></p>	<p><b>HDMI™ 接口</b>  支持 HDMI™ 2.1。</p>
<p><b>6</b></p>	<p><b>10 Gbps LAN 插孔</b> 此标准的 RJ-45 LAN 插孔接口,可以连接到局域网 (LAN) 上。您可以将网线连接其上。</p>
<p><b>7</b></p>	<p><b>200 Gbps QSFP LAN 端口</b> 请使用 DAC/AOC 电缆连接兼容系统。</p>
<p><b>8</b></p>	<p><b>Kensington 锁孔</b> 本设备配置 Kensington 锁孔,可用钥匙或机械插销连接的橡胶钢缆装置将设备固定锁住。钢缆末端有一个小圈用来绕一件牢固的物体 (例如沉重的桌子等),而将设备锁住。</p>

## 硬件安装

### 连接电源

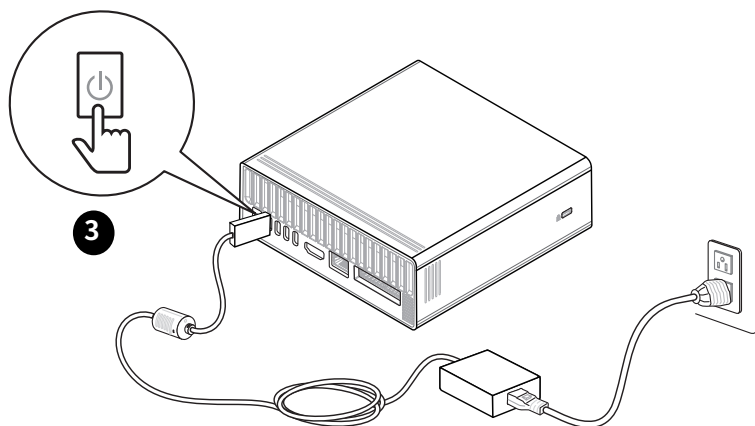
- 外置电源: 240W, 48.0V
  - 输入: 110~120Vac, 50/60Hz, 3.5A / 200~240Vac, 50/60Hz, 2.5A
  - 输出: 48.0V  $\equiv$  5.0A



### 注意

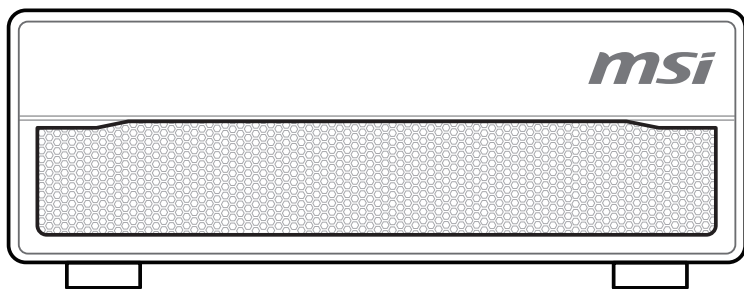
- 仅适用设备附带的适配器和电源线。若使用不同型号或额定功率较低的电源, 可能导致系统性能下降、无法启动或意外关机。
- 务必注意来自正在使用的适配器所产生的热量。
- 拔除 AC 电源线时, 务必握住电源线的接头部分。切勿直接扯拉电源线。

## 打开系统电源



## 系统放置

可水平安装此系统。

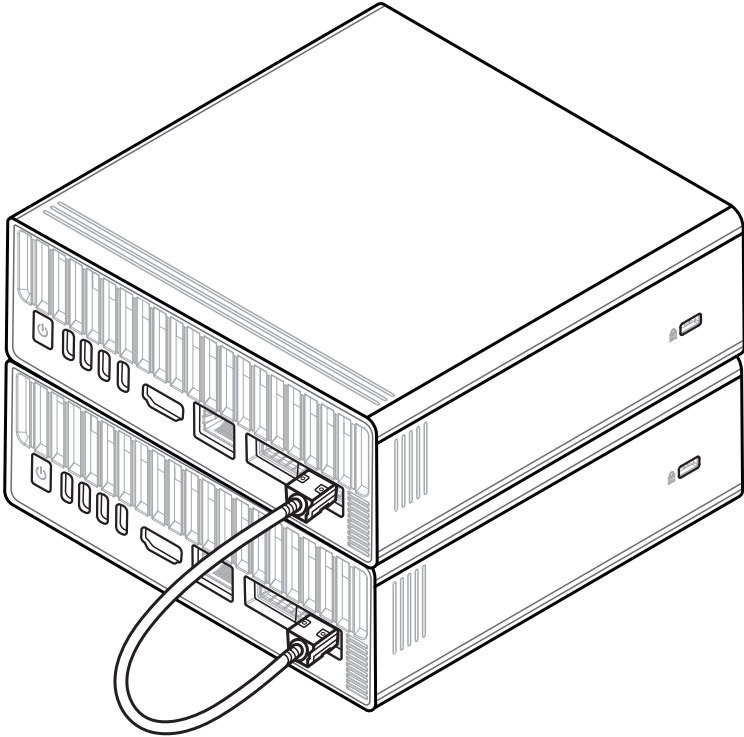


## 系统堆叠方式

使用选择性配置的 QSFP 电缆最多可堆叠两个系统。

### 注意

- 图中所示的第二个系统和连接用的 QSFP 电缆仅供安装说明, 不包含在包装中。
- 详细的互连过程, 请参阅系统堆叠说明。



# 初始设置



所有信息和屏幕截图如有变更,恕不另行通知。

## 什么是 NVIDIA DGX™ 操作系统

该设备预装了 NVIDIA DGX™ 操作系统,为运行 AI 和分析工作负载提供了一站式解决方案。初始系统配置将在首次启动后运行设置向导来完成。该设置向导为使用者提供快速的 DGX™ 系统入门体验。

NVIDIA DGX™ 操作系统提供定制版 Ubuntu Linux 安装,其中包含针对系统的优化和配置,附加驱动程序以及诊断和监控工具。它是一款稳定,经过全面测试且受支持的操作系统,可在此设备上运行 AI,机器学习和分析应用程序。

## 特色

- 预装 NVIDIA 驱动程序和 CUDA 工具包。
- 深度学习框架准备就绪 (如 TensorFlow, PyTorch)。
- 容器化支持 (NVIDIA GPU Cloud Containers + Docker)。
- 系统监控和诊断工具 (如 Data Center GPU Manager, NVIDIA 系统管理工具)。
- 支持整合 NGC 云端资源。协助开发者高效且无缝地在云端执行 AI/ML 机器学习工作负载。
- 优化的内核、网络堆栈和 I/O,以提高整体性能。

# 首次启动设置

本指南将引导完成系统的首次设置。您需要选择系统使用方式，并运行安装向导来完成所有设置。

## 你要执行的操作流程

本次设置包含以下步骤：

- 选择桌面模式或网络设备模式。
- 准备系统和连接。
- 运行安装向导以配置您的系统。

## 选择您的设置模式

您的系统可通过以下两种方式配置：

### 桌面模式

- 通过 USB 或蓝牙连接键盘和鼠标。
- 直接使用 Ubuntu 桌面进行工作。



### 注意

连接标准 USB 键盘或鼠标时，需要使用 USB-C 转 USB 适配器。

### 网络设备模式

- 通过网络远程访问系统。
- 作为服务器或计算节点使用。
- 无需本地显示器即可管理。



### 注意

您在此选择的模式将应用于整个初始设置流程。设置完成后，您可自由在桌面和网络设备模式之间切换。不受限于最初的选择。

## 准备事项

在开始之前,请确保您已具备以下条件:

- 系统已连接电源。
- 具备提供有效互联网连接的以太网,或无需网页认证(如酒店/机场)的可用 WiFi 网络。
- 适用于桌面模式已连接显示器、键盘和鼠标(或可通过蓝牙连接)。
- 适用于网络设备模式请确保同一网络中存在可用于远程访问的计算机。



**注意**

显示器故障排除:部分显示器可能在初次使用时无法正常显示。若您通过 USB-C/DisplayPort 连接时无画面输出,请尝试改用 HDMI 接口。



**注意**

若您打算使用有线网络连接,请在开始安装前插入网线。避免在后续安装过程中出现连接问题。

## 运行安装向导

安装向导将引导您完成以下步骤:

- 启动并执行初始化系统。
- 选择您所需的设置模式。
- 下载并安装关键更新。
- 完成初始配置。



**注意**

重要注意事项:请勿在更新过程中关闭或重新启动系统。下载开始后,安装程序无法中断,若在更新过程中断电或关机可能会导致系统损坏。

## 快速入门

启动安装的方式取决于您选择的模式:

桌面模式

1. 打开系统电源。
2. 安装向导将自动在已连接的显示器上启动。
3. 使用已连接的有线键盘和鼠标进行操作导航。
4. 若未检测到键盘或鼠标,系统将提示您将蓝牙设备切换至配对模式。

USB 设备可随时插入并应可自动工作,即使检测不正确也无妨。蓝牙设备可在“快速入门”屏幕上进入配对模式,通常仍可成功配对(例外情况:需输入配对密码的键盘无法在此屏幕上使用)。一旦您点击“快速入门”,蓝牙配对将终止,需断电重启系统才能重新尝试配对。

### 网络设备模式

1. 打开系统电源。
2. 请通过以下任一方式连接至系统:
  - 认证页面将自动显示设置用的 HTTP 地址。此地址同时标注于快速入门指南上,格式应如下: `http://spark-abcd.local`
  - 打开浏览器访问认证页面显示的地址。
  - 如需有线连接,此时可插入以太网线(此步骤为选择性配置)。

## 设置过程预期事项

安装向导将引导您完成多个配置步骤。请按照屏幕上的提示逐步完成每一步设置。

### 设置流程步骤:

1. 语言和时区选择  
选择系统所使用的语言和时区设置。
2. 键盘布局选择(仅桌面模式)  
选择您的键盘布局(例如:美式键盘或俄式键盘)。此屏幕仅在桌面模式下显示。
3. 条款和条件  
查阅并接受条款与条件以继续安装流程。
4. 用户账户创建  
创建系统登录用的用户名和密码。请注意,由于输入字段较长,在键入时系统会即时筛选。
5. 信息共享设置(选择性配置)  
配置分析数据和故障报告偏好。可根据需要跳过此步骤。
6. WiFi 网络选择  
选择您的 WiFi 网络若已连接可提供互联网接入的以太网电缆,则会自动跳过此步骤。
7. WiFi 密码输入  
输入所选 WiFi 网络的密码。
8. 接入 WiFi 网络  
系统将连接至您的 WiFi 网络并关闭接入点。您的计算机将自动重新连接至默认网络。

## 注意

- 网络连接问题处理。
- 若您的计算机已自动连接与系统相同的网络，安装流程将无缝持续进行。
- 若未自动连接，请在设置应用程序等待网络配置完成时，您需要手动将计算机连接至与系统相同的网络。
- 若安装失败，必须连接显示器 / 键盘 / 鼠标以继续操作。
- 错误信息窗口将提示您尝试重新连接至系统热点并再次重试。此方法仅适用于系统实际未能成功接入网络 (例如：密码错误) 的情况，若您的笔记本电脑与系统之间完全无法通信，该方法将不适用。
- 若错误信息出现时未检测到系统热点，表明系统已成功接入网络，但您的笔记本电脑无法与其建立通信。造成此情况的可能原因包括：
  - ▶ 设备隔离
  - ▶ 由于网络的 mDNS 设置有问题，故未成功接入与系统相同的网络。(例如：复杂的企业网络环境)

## 9. 软件下载与安装

成功连接网络后，系统将自动下载并安装完整的软件映像。

## 注意

在此过程中请勿关机或重新启动系统。下载开始后，安装进程将不可中断。

## 10. 安装完成

安装完成后，设备将自动重新启动，然后您可正常使用。

# 系统堆叠

本指南说明如何通过简化的网络配置和 QSFP/CX7 电缆, 将两个系统连接为虚拟计算集群, 以实现高性能互连。

该方案旨通过 MPI (用于处理器间的 CPU 通信) 和 NCCL v2.28.3 (用于 GPU 加速集体运算), 在 Grace Blackwell GPU 之间启用分布式运算工作负载。

更多详细信息请在 [Connect Two Sparks](#) 中参阅。

## 系统要求

在开始之前, 请确保以下事项:

- 两个系统均配备 Grace Blackwell GPUs, 并以 QSFP/CX7 电缆相互连接, 同时运行 Ubuntu 24.04 (或更高版本) 且已安装 NVIDIA 驱动程序。



**注意**

- 这些端口仅支持以太网配置。适用于这些端口的认证电缆如下:
  - Amphenol: NJAANK-N911 (QSFP to QSFP112, 32AWG, 400mm, LSZH), NJAANK0006 is the 0.5m version of this cable.
  - Luxshare: LMTQF022-SD-R (QSFP112 400G DAC Cable, 400mm, 30AWG).
- 
- 具备互联网接入能力已进行初始软件设置。
  - 两个系统上均拥有 sudo/root 访问权限。

## 系统间网络设置

选项 1: 请在两个系统节点上按照以下步骤使用 “netplan” 配置网络介面。以下命令应在终端会话 (本地或远程) 中运行。

1. 下载 netplan 配置文件。

```
sudo wget -O /etc/netplan/40-cx7.yaml https://github.com/NVIDIA/dgx-spark-playbooks/raw/main/nvidia/connect-two-sparks/assets/cx7-netplan.yaml
```

2. 为配置文件设置适当的权限。

```
sudo chmod 600 /etc/netplan/40-cx7.yaml
```

3. 应用 netplan 配置。

```
sudo netplan apply
```

选项 2:手动分配 IP (高级)。请按照以下步骤手动为专用集群网络分配 IP 地址。

1. 在节点 1 上,分配静态 IP 地址并启用网络介面。  

```
sudo ip addr add 192.168.100.10/24 dev enP2p1s0f1np1  
sudo ip link set enP2p1s0f1np1 up
```
2. 在节点 2 上,分配静态 IP 地址并启用网络介面。  

```
sudo ip addr add 192.168.100.11/24 dev enP2p1s0f1np1  
sudo ip link set enP2p1s0f1np1 up
```
3. 在节点 1 上验证连线,测试与节点 2 的连接。  

```
ping -c 3 192.168.100.11
```
4. 在节点 2 上验证连线,测试与节点 1 的连接。  

```
ping -c 3 192.168.100.10
```

## 运行系统发现脚本

此步骤将自动识别互连系统并设置无密码 SSH 认证。

以下命令请在两个节点的终端会话工作阶段 (本地或远程) 中执行。

1. 下载发现脚本。  

```
wget https://github.com/NVIDIA/dgx-spark-playbooks/raw/refs/heads/main/nvidia/  
connect-two-sparks/assets/discover-sparks
```
2. 将该脚本设置为可运行。  

```
chmod +x discover-sparks
```
3. 运行发现脚本。  

```
./discover-sparks
```

示例输出:

```
Found: 192.168.100.10 (spark-1b3b.local)  
Found: 192.168.100.11 (spark-1d84.local)  
Copying your SSH public key to all discovered nodes using ssh-copy-id.  
You may be prompted for your password on each node.  
Copying SSH key to 192.168.100.10 ...  
Copying SSH key to 192.168.100.11 ...  
nvidia@192.168.100.11's password:
```

SSH 密钥拷贝完成。现在这两个系统已建立互联通信。

## 安装必备软件并验证配置

完成网络配置并确保系统间可正常通信后，下一步需安装分布式工作负载所需的软件，通过运行测试工作负载来验证 GPU 间通信是否正常，并测量跨堆叠系统的性能。

欲取得完整的 NCCL 建置说明、NCCL 测试套件运行方式及结果解读，请参阅 [NCCL for two Sparks](#) 操作指南。

## 两个系统 NCCL 配置指南

在两个系统上安装并测试 NCCL。

### 1. 配置网络连接。

按照网络设置说明在系统节点之间建立连接。包括：

- 物理 QSFP 电缆连接。
- 网络介面配置 (自动或手动分配 IP)。
- 设置无密码 SSH。
- 验证网络连线。

### 2. 建立支持 Blackwell 的 NCCL。

在两个节点上执行这些命令，以从源代码建立支持 Blackwell 架构的 NCCL。

```
# Install dependencies and build NCCL
sudo apt-get update && sudo apt-get install -y libopenmpi-dev
git clone -b v2.28.3-1 https://github.com/NVIDIA/nccl.git ~/nccl/
cd ~/nccl/
make -j src.build NVCC_GENCODE="-gencode=arch=compute_121,code=sm_121"

# Set environment variables
export CUDA_HOME="/usr/local/cuda"
export MPI_HOME="/usr/lib/aarch64-linux-gnu/openmpi"
export NCCL_HOME="$HOME/nccl/build/"
export LD_LIBRARY_PATH="$NCCL_HOME/lib:$CUDA_HOME/lib64:$MPI_HOME/lib:$LD_LIBRARY_PATH"
```

### 3. 建立 NCCL 测试套件。

编译 NCCL 测试套件以验证通信性能。

```
# Clone and build NCCL tests
git clone https://github.com/NVIDIA/nccl-tests.git ~/nccl-tests/
cd ~/nccl-tests/
make MPI=1
```

### 4. 查找启用中的网络介面和其 IP 地址。

使用启用中的网络介面执行多节点 NCCL 性能测试。首先,确定可用且已启用的网络端口。

```
# Check network port status
ibdev2netdev
```

示例输出:

```
roceP2p1s0f0 port 1 ==> enP2p1s0f0np0 (Down)
roceP2p1s0f1 port 1 ==> enP2p1s0f1np1 (Up)
rocep1s0f0 port 1 ==> enp1s0f0np0 (Down)
rocep1s0f1 port 1 ==> enp1s0f1np1 (Up)
```

请使用在输出中显示为“(Up)”的介面。在此例子中,我们将使用 enp1s0f1np1。您可忽略以前缀 P2p<...> 开头的介面,只考虑以 enp1<...> 开头的介面。

您需要找到已启用的介面 IP 地址。在两个节点上,运行以下命令以取得 IP 地址,并记下以供下一步使用。

```
ip addr show enp1s0f0np0
ip addr show enp1s0f1np1
```

示例输出：

```
# In this example, we are using interface enp1s0f1np1.
nvidia@dgx-spark-1:~$ ip addr show enp1s0f1np1
4: enp1s0f1np1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq
state UP group default qlen 1000
    link/ether 3c:6d:66:cc:b3:b7 brd ff:ff:ff:ff:ff:ff
    inet **169.254.35.62**/16 brd 169.254.255.255 scope link noprefixroute
enp1s0f1np1
        valid_lft forever preferred_lft forever
    inet6 fe80::3e6d:66ff:fecc:b3b7/64 scope link
        valid_lft forever preferred_lft forever
```

在此示例中，节点 1 的 IP 地址为 169.254.35.62。对节点 2 重复相同操作。

## 5. 运行 NCCL 通信测试。

在两个节点上执行以下命令以运行 NCCL 通信测试。将 IP 地址和介面名称替换为先前步骤中取得的值。

```
# Set network interface environment variables (use your Up interface from the
previous step)
export UCX_NET_DEVICES=enp1s0f1np1
export NCCL_SOCKET_IFNAME=enp1s0f1np1
export OMPI_MCA_btl_tcp_if_include=enp1s0f1np1

# Run the all_gather performance test across both nodes (replace the IP addresses
with the ones you found in the previous step)
mpirun -np 2 -H <IP for Node 1>:1,<IP for Node 2>:1 \
--mca plm_rsh_agent "ssh -o UserKnownHostsFile=/dev/null -o
StrictHostKeyChecking=no" \
-x LD_LIBRARY_PATH=$LD_LIBRARY_PATH \
$HOME/nccl-tests/build/all_gather_perf
```

您还可以使用较大的缓冲区来测试 NCCL 设置, 以充分利用 200Gbps 频宽。

```
# Set network interface environment variables (use your active interface)
export UCX_NET_DEVICES=enp1s0f1np1
export NCCL_SOCKET_IFNAME=enp1s0f1np1
export OMPI_MCA_btl_tcp_if_include=enp1s0f1np1

# Run the all_gather performance test across both nodes
mpirun -np 2 -H <IP for Node 1>:1,<IP for Node 2>:1 \
  --mca plm_rsh_agent "ssh -o UserKnownHostsFile=/dev/null -o
  StrictHostKeyChecking=no" \
  -x LD_LIBRARY_PATH=$LD_LIBRARY_PATH \
  $HOME/nccl-tests/build/all_gather_perf -b 16G -e 16G -f 2
```

注意: 在 mpirun 命令中, IP 地址后面会加上 :1。例如: mpirun -np 2 -H 169.254.35.62:1,169.254.35.63:1

## 6. 清理和回复设置。

```
# Rollback network configuration (if needed)
rm -rf ~/nccl/
rm -rf ~/nccl-tests/
```

## 7. 下一步。

您的 NCCL 环境已准备就绪, 可在系统上执行多节点分布式训练工作负载。接下来, 您可尝试运行较大的分布式工作负载, 例如 TRT-LLM 或 vLLM 推理。

## 故障排除

- 请确保 QSFP/CX7 介面处于活动状态并用于 IP 分配。
- 通过“ping”验证节点间连接状况。
- 使用“ip a”和“ethtool”检查介面绑定情况。
- 若发现脚本执行失败, 请手动验证节点间的 SSH 连接是否正常。
- 更多故障排除指南和支持选项, 请参阅 [Maintenance and Troubleshooting](#)。

# 升级 NVIDIA DGX™ 操作系统

如需升级至最新的操作系统或软件包, 请参阅:

[https://ipc.msi.com/product\\_download/Industrial-Computer-Box-PC/AI-Supercomputer/EdgeXpert-MS-C931](https://ipc.msi.com/product_download/Industrial-Computer-Box-PC/AI-Supercomputer/EdgeXpert-MS-C931)

## 重新安装 NVIDIA DGX™ 操作系统



重新安装操作系统将会清除存储在驱动器上的所有数据。包括 /home 分区下所有用户的文档、软件设置和其他个人文件。

这个设备已预装 NVIDIA DGX™ 操作系统, 仅在有限的情况下需要重新安装, 例如:

- 更换存储设备。
- 重建群集节点。
- 从系统故障中恢复。

## 创建可引导的 U 盘

在 Windows 系统上, 请参阅:

[https://ipc.msi.com/product\\_download/Industrial-Computer-Box-PC/AI-Supercomputer/EdgeXpert-MS-C931](https://ipc.msi.com/product_download/Industrial-Computer-Box-PC/AI-Supercomputer/EdgeXpert-MS-C931)

## 启动 NVIDIA DGX™ OS ISO 映像

1. 将包含操作系统映像的 U 盘插入系统。
2. 将显示器和键盘直接连接至系统。
3. 启动系统, 当出现 NVIDIA 徽标时按 F2 进入启动菜单。
4. 选择与所插入 U 盘对应的 USB 卷标名称, 并从此设备启动系统。

# NVIDIA Sync

NVIDIA Sync 是一款系统托盘实用程序, 提供用户在系统没有连上显示器或键盘时也能从另一台机器轻松存取该系统的方法。

## 安装

1. 请从 <https://build.nvidia.com/spark> 下载最新版 NVIDIA Sync。安装程序提供 Windows、macOS 和 Linux 版本。
2. 运行安装程序。
3. NVIDIA Sync 将自动扫描可远程连接至本系统的兼容应用程序。选择您需要使用的应用程序, 并点击“Next”。
4. 输入系统的计算机名称和您的登录凭据。

## 支持的应用程序

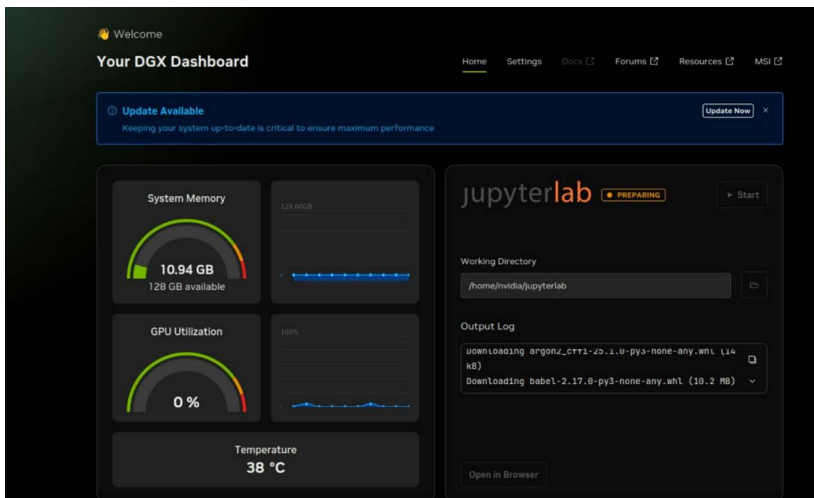
- AI Workbench
- Cursor IDE
- VSCode
- Windsurf

## 其他连接方法

- DGX™ 动态面板 (通过网页浏览器访问)
- SSH 终端 (由 NVIDIA Sync 自动管理 RSA 密钥)

# DGX™ 动态面板

该系统配备内置动态面板,可概览当前运行指标,并执行系统更新、修改部分系统设置以及访问本地 Jupyter Notebooks。



DGX™ 动态面板提供实时系统监控功能,并集成 JupyterLab 访问。



**注意**

要运行更新和变更设备名,您必须拥有“sudo”权限。系统初始设置阶段创建的账户将自动具备此访问权限。

## 集成 JupyterLab

DGX™ 动态面板集成的 JupyterLab 实例提供便利的开发环境,其特点如下:

- 启动 JupyterLab 时,会在指定的工作目录中创建一个虚拟环境,并自动安装一组推荐套组。
- 若指定新的工作目录并启动 JupyterLab,系统将会创建一个新的虚拟环境。
- 设备上的每个用户账户都会分配一个端口,位于 `/opt/nvidia/dgx-dashboardservice/jupyterlab_ports.yaml`。
- 若要远程访问 JupyterLab,必须像访问动态面板的方式建立通道。通道所用的端口可于上述配置文件中查询。使用 NVIDIA Sync 时,该通道将会自动管理,无需手动设置,即可直接使用。

## 访问动态面板

在本地的 Ubuntu 桌面左下角点击“显示应用程序”按钮。然后，在应用网格中选择“DGX 动态面板”快捷方式，在默认浏览器中打开动态面板。

远程访问可通过 NVIDIA Sync 或手动创建的 SSH 通道访问动态面板。

如果使用 NVIDIA Sync 连接后，只需单击“DGX 动态面板”按钮，动态面板将在默认浏览器中打开，网址为：`http://localhost:11000`。

若要手动访问 SSH，首先创建一个通道，例如：`ssh -L 11000:localhost:11000 <username>@<IP or spark-abcd.local>`。然后，在浏览器中打开动态面板 `http://<spark host ip>:11000`。

## NVIDIA Docker 容器运行时

NVIDIA 容器运行时可让 Docker 容器访问系统中的 GPU 资源。该运行时作为 Docker 和 NVIDIA 驱动程序间的桥梁，可使容器能够利用 GPU 加速能力运行 AI/ML 工作负载、CUDA 应用程序及其他利用 GPU 加速的软件。

主要优势：

- 实现容器内无缝访问 GPU 资源。
- 自动管理驱动程序和程序库。
- 支持多 GPU 配置。
- 兼容主流容器编排平台。

该运行时需与 NVIDIA 容器工具包配合使用，该工具包提供关键组件，可将 GPU 设备与 CUDA 程序库暴露给容器化应用程序使用。

### 安装

系统中已预安装并配置 NVIDIA 容器工具包。包括：

- NVIDIA 容器运行时。
- Docker 集成功能。
- GPU 设备访问配置。
- CUDA 程序库管理。

该运行时实现开箱即用，可立即运行 GPU 加速的容器应用程序。

## 选择性配置:将用户添加至 Docker 群组

默认情况下,运行 Docker 命令需具备 sudo 权限。将用户添加至 docker 群组后可在不使用 sudo 的情况下直接运行 Docker 命令,其优势如下:

- 便捷性:无需在每个 Docker 命令前输入 sudo。
- 工作流程优化:实现与开发工具及脚本的无缝集成。
- 操作减负:提升容器操作的迭代效率。

将用户添加至 docker 群组:

```
sudo usermod -aG docker $USER
```



**注意**

- 您必须登出并重新登录系统 (或重启会话) 以使群组成员资格生效。
- 此步骤为可选性配置:若您不想修改群组成员资格,可继续通过 sudo 权限使用 Docker。

## 使用指南

基本 GPU 访问。使用 `-gpus` 参数运行可访问 GPU 的容器,例如:

```
docker run -it --gpus=all nvcr.io/nvidia/cuda:13.0.1-devel-ubuntu24.04 nvidia-smi
```

此命令功能如下:运行互动式容器 (-it), 启用所有 GPUs 的访问权限 (-gpus=all), 使用 NVIDIA CUDA 开发映像, 执行 nvidia-smi 以显示 GPU 信息。

设置 GPU 功能。可精确控制容器可用的 GPU 功能类型。

```
docker run -it --gpus '"capabilities=compute,utility"' nvcr.io/nvidia/cuda:13.0.1-devel-ubuntu24.04 nvidia-smi
```

挂载 CUDA 程序库。对于需要特定 CUDA 程序库的应用程序,可从主机直接挂载至容器中使用。

```
docker run -it --gpus=all \  
-v /usr/local/cuda:/usr/local/cuda:ro \  
nvcr.io/nvidia/cuda:13.0.1-devel-ubuntu24.04 bash
```

## 验证

测试 GPU 访问。

1. 运行测试命令以验证容器是否可正常访问 GPU。

```
docker run -it --gpus=all nvcr.io/nvidia/cuda:13.0.1-devel-ubuntu24.04 nvidia-smi
```

预期输出应显示以下信息:GPU 设备信息, 驱动程序版本, CUDA 版本, 内存使用情况与温度。

2. 检查运行时配置。

```
docker info | grep -A 10 "Runtimes"
```

3. 验证 NVIDIA 运行时是否可用。

```
docker run --rm --runtime=nvidia nvcr.io/nvidia/cuda:13.0.1-devel-ubuntu24.04 nvidia-smi
```

检查容器内的 GPU 资源访问权限。验证正在运行的容器中可用的 GPU 资源。

```
docker run -it --gpus=all nvcr.io/nvidia/cuda:13.0.1-devel-ubuntu24.04 bash
# Inside the container:
nvidia-smi
ls /dev/nvidia*
```

## 故障排除

若出现未找到运行时的错误信息。

1. 请验证 NVIDIA 容器工具包是否已正确安装。

```
nvidia-ctk --version
```

2. 检查 Docker 守护进程配置。

```
cat /etc/docker/daemon.json
```

3. 重新启动 Docker 服务。

```
sudo systemctl restart docker
```

若出现 CUDA 版本不匹配的提示。

1. 请检查主机上的 CUDA 驱动程序版本。

```
nvidia-smi
```

2. 请使用与 CUDA 版本兼容的容器镜像。

```
docker run -it --gpus=all nvcr.io/nvidia/cuda:12.0.1-devel-ubuntu24.04 nvidia-smi
```

若遇到权限错误。

1. 请确保您的用户已加入 docker 群组 (若不使用 sudo 的情况下)。  
`groups $USER`
2. 检查设备权限。  
`ls -la /dev/nvidia*`
3. 验证 Docker 守护进程是否具备访问 GPU 设备的权限。  
`sudo docker run -it --gpus=all nvcr.io/nvidia/cuda:13.0.1-devel-ubuntu24.04 nvidia-smi`

若容器无法启动。

1. 请检查 Docker 日志。  
`docker logs <container_id>`
2. 验证主机上 GPU 设备是否可用。  
`ls /dev/nvidia*`
3. 使用最小化容器进行测试。  
`docker run --rm --gpus=all nvcr.io/nvidia/cuda:13.0.1-devel-ubuntu24.04 echo "GPU test successful"`

## NGC

NVIDIA GPU Cloud (NGC) 是一个综合性的注册中心, 提供 GPU 优化容器、预训练模型及 AI/ML 软件, 能加速 AI 应用程序的开发与部署。对于用户而言, NGC 提供了最新框架, 工具及专为 Grace Blackwell 架构优化环境的访问权限。

用户主要优势:

- 优化容器: 提供预配置环境, 内含最新 AI/ML 框架、CUDA 及程序库, 针对 Grace Blackwell GPU 进行最佳优化。
- 预训练模型: 可使用最先进的模型及各类 AI 任务的模型集。
- 快速开发: 跳过复杂的环境设置, 专注于 AI/ML 项目研发。
- 先进软件: 访问最新的 NVIDIA 软件堆叠及实验性功能。

NGC 对用户特别有价值, 因为它为这个新平台提供了最先进和优化的软件堆叠, 确保您可以访问最新的性能优化和功能。

## 快速入门

创建 NGC 账户。

1. 访问 NGC 官方网站。
2. 点击 Sign Up 创建免费账户。
3. 验证您的电子邮件地址。
4. 补充完善个人信息。

生成 API 密钥。

1. 登录您的 NGC 账户。
2. 进入 Setup API Key 设置项。
3. 点击 Generate API Key。
4. 复制并安全存储您的 API 密钥。



您的 API 密钥用于拉取容器及访问 NGC 资源。请务必妥善保管，切勿公开分享。

安装 NGC CLI (选择性配置)。NGC 提供便捷的命令行方式访问 NGC 资源。

```
# Download and install NGC CLI
wget https://ngc.nvidia.com/downloads/ngccli_linux.zip
unzip ngccli_linux.zip
sudo mv ngc-cli/ngc /usr/local/bin/
ngc config set
```

认证及配置 Docker，以便访问 NGC 注册表。

```
# Login to NGC with Docker
docker login nvcr.io
# Username: $oauthtoken
# Password: <your-api-key>
```

## 基本用法

拉取并运行容器。可从热门的 AI/ML 框架容器开始入门。

```
# Pull a PyTorch container optimized for Grace Blackwell
docker pull nvcr.io/nvidia/pytorch:24.08-py3
# Run the container with GPU access
docker run -it --gpus=all nvcr.io/nvidia/pytorch:24.08-py3
```

浏览可用资源。通过网页介面探索 NGC 资源。

- 容器: AI/ML 框架、开发环境及专用工具。
- 模型: 涵盖计算机视觉、自然语言处理等领域的预训练模型。
- Helm Charts: Kubernetes 部署配置。
- Jupyter Notebooks: 交互式教程与示例。

## 常见工作流程

开发环境。使用 NGC 容器作为您的开发环境。

```
# Run a development container with persistent storage
docker run -it --gpus=all \
-v /path/to/your/project:/workspace \
nvcr.io/nvidia/pytorch:24.08-py3
```

模型推理与训练。访问预训练模型及训练脚本。

```
# Pull a model from NGC
ngc registry model download-version nvidia/bert-base-uncased:1
# Or use models directly in containers
docker run -it --gpus=all \
nvcr.io/nvidia/tensorflow:24.08-tf2-py3
```

## 最佳实践

容器管理。

- 锁定版本: 使用特定的容器标签以确保环境可重复。
- 定期更新: 定期更新到较新的容器版本以进行最新优化。
- 资源限制: 为工作负载设置适当的内存和 CPU 限制。

数据持久性。

- 卷挂载:将数据目录挂载至容器中以实现持久化存储。
- 模型存储:将训练好的模型和检查点保存在容器外部。
- 配置管理:将配置文件纳入版本控制系统。

安全。

- API 密钥安全:妥善保管您的 NGC API 密钥,并定期更新。
- 容器扫描:使用前扫描容器以检查潜在漏洞。
- 网络安全:为您的环境使用适当的网络配置。

## 故障排除

认证失败。

```
# Verify your API key is correct
docker login nvcr.io
# Check if your account has access to the requested resource
```

容器拉取故障。

```
# Check network connectivity
ping nvcr.io
# Verify container name and tag
docker search nvcr.io/nvidia/
```

GPU 访问异常。

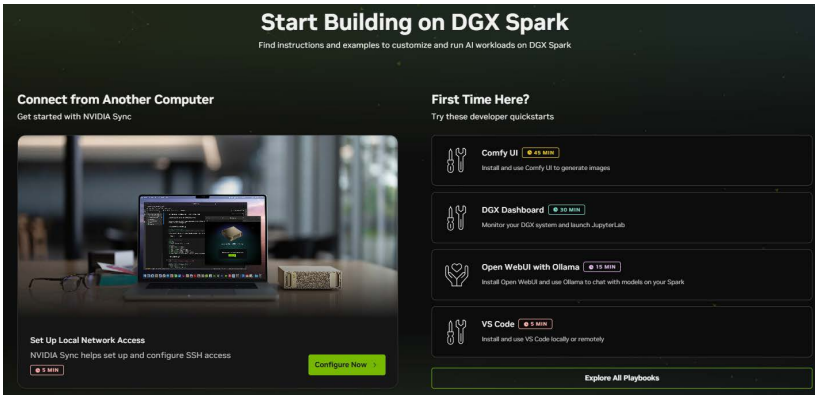
```
# Verify NVIDIA Container Runtime is installed
docker run --rm --gpus=all nvidia/cuda:12.0-base-ubuntu20.04 nvidia-smi
```

## 获取帮助

- NGC文档:请查阅 NGC 官方文档。
- 社区论坛:加入 NVIDIA 开发者论坛。

# 从 NVIDIA 官方网站获取和激活 AI 模型

如需了解定制和运行 AI 工作负载的说明与示例, 请访问 NVIDIA 开发者网站: <https://build.nvidia.com/spark>



## 固件更新

本章节提供系统固件组件的更新指导。



此更新信息仅适用于 Founders Edition。其他制造商的设备可能有不同的固件更新流程。

## 建议方法

NVIDIA 建议使用 DGX™ 动态面板来执行系统固件更新。DGX™ 动态面板提供用户友好的操作介面, 用于管理固件更新和系统维护任务。

有关 DGX™ 动态面板的访问和使用详情, 请参阅 DGX™ 动态面板使用指南。



- 确保系统连接稳定电源。
- 关闭所有运行中的应用程序并保存工作进度。
- 准备好系统恢复计划。
- 尽可能在系统维护期间安排更新。

## 手动更新方法

若无法使用 DGX™ 动态面板,可通过以下步骤手动更新固件:

1. 在系统中打开远程或本地终端。
2. 依次运行以下命令。

```
sudo apt update
sudo apt upgrade
sudo fwupdmgm refresh
sudo fwupdmgm upgrade
sudo reboot
```

## 故障排除

若在固件更新过程中遇到问题:

- 请确保更新过程持续供电稳定。
- 有关更多故障排除指南和支持选项,请参阅 [spark-maintenance-troubleshooting](#)。

## 其他资源

- 请访问 NVIDIA Spark 开发者入门网站:<https://build.nvidia.com/spark> 以获取最新指南、教程和更新信息。
- 有关最新的软件更新和功能,请参阅:[spark-release-notes](#)。
- 有关常见问题的故障排除,请参考:[spark-known-issues](#)。

您的系统现已准备就绪,可支持 AI 开发与部署工作流程!

# 安全注意事项

- 请务必仔细阅读安全注意事项。
- 应留意设备上或用户指南中的所有小心和警告事项。
- 仅限有资质的技术人员来提供维修服务。
- IEC 60825-1:2014 过渡条款。本产品符合美国FDA/CDRH 激光产品性能标准,但不限於 IEC 60825-1 (第 3 版) 的要求。具体内容见 2019 年 5 月 8 日发布的 FDA 激光通知第 56 号。
- 本设备的 SFP 端口应配合 UL 认证的可选配收发器产品使用。该收发器需额定电压为 3.3Vdc,且符合 1 级激光标准 (Laser Class 1)。

## 电源

- 设备连接电源插座之前,请确保电源电压位于安全范围内并且适当调整至 100~240V 之间。
- 如果电源线带有 3 针插头,请勿关闭插头的保护接地针脚。电脑必须连接到接地的电源插座上。
- 请确认安装现场的配电系统应提供额定电压为 120/240V, 20A (最大值) 的断路器。
- 在安装任何附加卡或模块前,请务必拔掉电源线。
- 如果需要长时间停用设备,请务必拔掉电源线或者关闭插座开关,以实现零能耗。
- 放置电源线时应避免其被踩踏。请勿在电源线上放置任何物品。
- 如果此设备配备一个适配器,请仅使用 MSI 提供的交流适配器,该适配器可用于此设备。

## 电池

如果此设备随附电池,请采取特殊预防措施注意事项。

- 错误更换电池有爆炸的危险。仅使用制造商推荐的相同或等效类型的产品进行更换。
- 避免将电池丢入火中或热烤箱中,或机械破碎或切割电池,以免引起爆炸。
- 避免将电池放在高温或极低气压的环境中,以免引起爆炸或易燃液体或气体泄漏。
- 请勿摄入电池。如果硬币/纽扣电池被吞下,会导致严重的内部灼伤,甚至死亡。请将新旧电池放在儿童拿不到的地方。

### 欧盟:



电池,电池组和蓄电池不应该作为未分类的生活垃圾来处理。请依据当地法规来使用公共收集系统返回,回收或处理它们。

### 廢電池請回收:



廢電池請回收

为了更好的保护环境。废电池应该单独收集回收或特殊处理。

## 加州,美国:



按钮电池可能含有高氯酸盐材料,当回收或处置时需要特殊处理。  
更多信息请访问: <https://dtsc.ca.gov/perchlorate/>

## 环境

- 为减少因热引起的伤害或设备过热的可能性,请勿将设备放置在柔软,不稳定的表面上或将塞设备的通风孔。
- 仅在坚硬,平坦且稳定的表面上使用此设备。
- 使设备远离潮湿及高温场所,以避免火灾及触电。
- 请勿将设备置于存储温度高于 60°C 或低于 -20°C 的不良环境中,否则可能损坏设备。
- 工作温度范围约为0°C至35°C。
- 清洁设备时,请务必拔除电源插头。请使用软布来清洁设备,勿使用工业化学清洁剂。切勿将任何液体倒入设备开口,以免损坏设备或导致触电。
- 必须使设备远离带有强磁性或强电的物体。
- 若出现以下任何情形,请联系服务人员进行检修设备:
  - 电源线或插头损坏。
  - 液体进入设备。
  - 设备受潮。
  - 设备工作不正常,或者按照用户指南的说明无法正常使用。
  - 设备摔落和损坏。
  - 设备已明显破损。

产品中有害物质的名称及含有信息表

部件名称	有害物质									
	Pb	Hg	Cd	Cr(VI)	PBBs	PBDEs	DBP	DIBP	BBP	DEHP
电路板组件*	×	○	○	○	○	○	○	○	○	○
处理器和散热器	×	○	○	○	○	○	○	○	○	○
内存条/硬盘	×	○	○	○	○	○	○	○	○	○
电缆/连接器	×	○	○	○	○	○	○	○	○	○
输出输入设备	×	○	○	○	○	○	○	○	○	○
电源供应器/适配器	×	○	○	○	○	○	○	○	○	○
金属机构件	×	○	○	○	○	○	○	○	○	○

注1：○：表示该有害物质在该部件所有均质材料中的含量均不超出电器电子产品有害物质限制使用国家标准要求。  
 ×：表示该有害物质至少在该部件的某一均质材料中的含量超出电器电子产品有害物质限制使用国家标准要求。  
 注2：以上未列出的部件，表明其有害物质含量均不超出电器电子产品有害物质限制使用国家标准要求。  
 注3：上述表格标注“×”之部件，皆符合达标管理目录限用物质应用例外清单之限值要求。  
 \* 电路板组件：包括印刷电路板及其构成的零部件。

## 环境方针

- 本装置及其零部件在设计时即设定为再利用和回收，请勿在达到使用寿命时任意丢弃。
- 用户应联系当地的授权回收点，回收并处置达到使用寿命的产品。
- 如需更多回收信息，请访问微星网站 <[https://csr.msi.com/global/pevn\\_ewaste](https://csr.msi.com/global/pevn_ewaste)> 并找到最近的经销商。



## 升级与保修

如需进一步了解用户购买的产品，请联系本地经销商。如果未取得经销商和服务中心授权，请勿尝试升级或更换任何产品部件，否则将会导致保修条款失效。

## 版权与商标声明



Copyright © 微星科技股份有限公司所有。MSI 标志为微星科技公司注册所有，本文档提及及其他所有商标是其各自所有者的资产。我们精心准备了本文档，但不保证其内容准确无误。我们的产品会不断改进，因此保留进行变更的权利，恕不另行通知。



词语 HDMI™、HDMI™ High-Definition Multimedia Interface (高清晰度多媒体接口)、HDMI™ 商业外观和 HDMI™ 徽标均为 HDMI™ Licensing Administrator, Inc. 的商标或注册商标。

## 技术支持

若系统发生故障并且用户手册中未提供解决办法，请与销售商或当地经销商联系。此外，也请尝试下列资源。访问 MSI 网站以了解常见问题及解答、技术指南、BIOS 更新、驱动程序更新和其他信息：<https://www.msi.com/support/>