Enable NVIDIA GPU passthrough

- Requirements
- Turn on the feature
- Build the bundle
- Install the bundle
- Run a sandbox with a GPU
- Troubleshooting
  - The script reports ... not found in the driver download
  - GPU workloads fail after a Docker Sandboxes upgrade

> [!IMPORTANT] GPU passthrough is experimental. The `--gpu` flag, the driver bundle, and the setup steps on this page are subject to change.

GPU passthrough in local Docker Sandboxes runs workloads on a physical NVIDIA GPU.

GPU passthrough in Docker sandboxes works via VFIO, a Linux feature that assigns a PCI device directly to a virtual machine. The GPU is bound to VFIO instead of the host's driver, and the sandboxed workload drives the hardware itself.

Requirements

VFIO-based GPU passthrough is supported only on `x86_64` Linux hosts (not Arm) with NVIDIA GPUs, and requires a GPU that nothing else is using: a headless host with a GPU, or an additional GPU.

The host also needs IOMMU turned on in its BIOS, and the `iommufd` and `vfio_pci` kernel modules loaded:

```
sudo modprobe -a iommufd vfio_pci
```

For the sandbox to drive the GPU, it requires:

- The `nvidia` and `nvidia-uvm` kernel modules, built for the Docker Sandboxes guest kernel
- The NVIDIA userspace driver libraries and firmware

Docker Sandboxes looks for these dependencies, packaged as an EROFS image, at `/usr/libexec/nerdbox-nvidia-bundle.erofs`. This bundle isn't included with Docker Sandboxes. To build it, see Build the bundle.

Turn on the feature

The `--gpu` flag is hidden until you turn on experimental features and the GPU feature flag:

```
sbx settings set platform.allowExperimentalFeatures true
sbx settings set feature.sandbox-gpu true
```

Build the bundle

A zip archive containing all the components required to build the bundle is published with each Docker Sandboxes release, as `nerdbox-nvidia-modules-x86_64.zip`. The archive contains the kernel modules (`nvidia.ko` and `nvidia-uvm.ko`), the driver version they were built against (`VERSION`), and a build script.

The build script runs a `linux/amd64` container that downloads the matching NVIDIA driver, assembles the bundle, and installs it to `/usr/libexec/nerdbox-nvidia-bundle.erofs`. Writing to that path requires root, hence the `sudo` in the following commands.

Prerequisites:

- Network access to `download.nvidia.com`
- Docker

Download and unpack the archive, then run the script from the unpacked directory:

```
curl -fSLO https://github.com/docker/sbx-releases/releases/latest/download/nerdbox-nvidia-modules-x86_64.zip
unzip nerdbox-nvidia-modules-x86_64.zip -d nvidia-modules
cd nvidia-modules
```

```
sudo ./prepare-nvidia-bundle.sh
```

If the script can't write to the output path, it leaves the bundle in the current directory and prints the `install` command that finishes the job.

Two environment variables override the script's default behavior:

Variable	Default	Purpose
`OUTPUT`	`/usr/libexec/nerdbox-nvidia-bundle.erofs`	Where the finished bundle is written.
`ACCEPT_NVIDIA_LICENSE`	None	Set to `1` to accept the NVIDIA license non-interactively, for scripts or CI.

For example, the following command writes the finished bundle to `/mnt/some-place/nerdbox-nvidia-bundle.erofs`:

```
OUTPUT=/mnt/some-place/nerdbox-nvidia-bundle.erofs ./prepare-nvidia-bundle.sh
```

Install the bundle

If you ran the script on the `x86_64` Linux host that runs your GPU sandboxes, and you didn't override `OUTPUT`, the bundle is already in place. If you built it elsewhere, copy the `nerdbox-nvidia-bundle.erofs` file it produced into that host's `/usr/libexec` directory.

Run a sandbox with a GPU

To run a sandbox with GPU passthrough, pass the `--gpu` flag:

```
sbx create --gpu claude .
```

The `sbx run` command takes the same flag:

```
sbx run --gpu claude
```

The flag takes effect when the sandbox is created. Passing it when you re-attach to an existing sandbox has no effect.

> [!IMPORTANT] Each Docker Sandboxes release uses a specific guest kernel. The NVIDIA kernel modules in your bundle must match that kernel. After upgrading Docker Sandboxes, download the new release's `nerdbox-nvidia-modules-x86_64.zip` archive and run the script again to rebuild the bundle.

Troubleshooting

The script reports `... not found in the driver download`

The extracted driver didn't contain an expected library or firmware file. Confirm that the download completed. If a library is named differently in your driver version, adjust `DRIVER_LIB_FAMILIES` in the script.

GPU workloads fail after a Docker Sandboxes upgrade

The guest kernel or the pinned driver version likely changed. Download the new release's archive, re-run the script, and reinstall the bundle.

□□□ #2
□ □□□ □□□ 2026-10-31 18:58:39 UTC
□ □□□ □□□ 2026-10-31 18:58:40 UTC