



# CDW Documentation

## NVIDIA GPU Firmware Upgrade Runbook (Revised)

---

# NVIDIA GPU Firmware Upgrade Runbook (Revised)

## Process at a Glance

1. **Prepare** — record current versions, verify GPU health, stop all workloads
2. **Validate Package** — checksum the firmware file, preview what will change
3. **Phase 2 (in this exact order):**
  1. BMC firmware update
  2. BMC reset (mc reset cold)
  3. Motherboard tray update (SBIOS, CPLD, PSU, etc.)
  4. GPU tray update (VBIOS, NVSwitch, EROT, FPGA)
  5. Poll BackgroundCopyStatus until **Completed**
  6. Chassis power cycle (graceful, via Redfish/ipmitool)
  7. Re-check versions on staged components
  8. CommitImage on **HGX\_FW\_BMC\_0** and **HGX\_FW\_HostBIOS\_0**
  9. **AC power cycle** (full PDU/cord removal, 5-6 min)
  10. NIC, TPM, NVMe firmware (last — these are independent and tolerate the AC cycle)
4. **Validate** — confirm new versions, GPU health, NVLink, smoke test

<note warning> Flashing the wrong firmware file or interrupting mid-flash can permanently brick a GPU. Read the full runbook before executing.

**Critical ordering:** BMC must be updated and reset *before* the motherboard and GPU trays so the BMC speaks the new Redfish schema used by the rest of the bundle. Skipping the AC power cycle leaves EROT/CPLD images staged but not activated. </note>

---

## 1. Overview

This runbook covers the end-to-end firmware upgrade process for NVIDIA DGX/HGX H100/H200 systems, GB200/B200 (Blackwell) systems, and standalone PCIe GPUs.

### Scope — what this runbook covers:

- **Hopper:** DGX H100/H200, HGX H100/H200
- **Blackwell:** DGX B200, HGX B200, GB200 NVL72 compute tray (per-tray BMC), GB300
- **Standalone PCIe GPUs** via NVFlash
- Components: VBIOS, NVSwitch, EROT, FPGA, BMC, SBIOS, CPLD, PSU, NIC, TPM, NVMe, and (Blackwell only) Grace CPU FW, NVLink Switch tray FW, ConnectX-7/8 onboard NICs

### Blackwell-specific notes (GB200 / B200):

- GB200 NVL72 racks have **two BMCs per compute tray** (HMC for the GPU/CPU complex and a separate chassis BMC) plus an **NVLink Switch tray BMC**. Update **HMC first**, then chassis BMC, then NVLink Switch tray, then compute tray firmware.

- **Grace CPU firmware** (HGX\_FW\_GraceCPU\_0, HGX\_FW\_GraceCPU\_1) is updated with the SBIOS package on Blackwell. It activates on the chassis power cycle (no separate AC requirement).
- **NVLink Switch tray** firmware (NVSwitch\_Tray\_\*) is a separate .fwpkg and is updated from the rack-level management host, not the compute tray BMC.
- The mandatory **AC power cycle** for EROT/CPLD also applies on Blackwell — for GB200 NVL72 this means power-cycling the full rack PDU feeds (coordinate with DC ops).
- **nvwupd ≥ 3.0.0** is required for Blackwell (HMC + Grace + NVLink Switch tray Redfish paths). 2.0.4 is the floor for Hopper only.

**Scope — what this runbook does NOT cover:**

- DGX A100 (use the Firmware Update Container — separate runbook)
- Standalone Mellanox/ConnectX adapter firmware in non-DGX hosts (use mlxfwmanager separately)
- Quantum/Spectrum InfiniBand or Ethernet switch firmware
- Storage controller / RAID HBA firmware

## 2. Background

### Components Updated

Component	What It Controls	Activation Trigger
BMC	Out-of-band server management	BMC reset (mc reset cold)
SBIOS / UEFI	Motherboard initialisation	Chassis power cycle + CommitImage
CPLD	Board-level glue logic, power sequencing	<b>AC power cycle (mandatory)</b>
VBIOS	GPU boot, clocks, power limits, PCIe config	Chassis power cycle
NVSwitch FW	GPU-to-GPU interconnect (NVLink fabric)	Chassis power cycle
EROT	Hardware root of trust / secure boot	<b>AC power cycle (mandatory)</b>
FPGA	Power management and control plane	Chassis power cycle
PSU	Power supply unit firmware	Live, one at a time — redundant PSU carries load. Refuses to flash if redundancy is unhealthy.
NIC / ConnectX	Onboard NIC firmware (incl. ConnectX-7/8 on Blackwell)	Chassis power cycle (run AFTER the AC cycle in this runbook)
TPM	Trusted Platform Module	Chassis power cycle (run AFTER the AC cycle in this runbook)
NVMe	Onboard storage firmware	Chassis power cycle (run AFTER the AC cycle in this runbook)

<note important> EROT and CPLD images are staged into a secondary flash region by the update, but only activate on a full AC power removal — a warm reboot or chassis power cycle is not sufficient.  
</note>

## Driver Compatibility

The minimum driver depends on the **firmware package version**, not the GPU architecture alone. Refer to the **per-package release notes** shipped with the .fwpkg you intend to install. Do not rely on generic architecture → driver tables; they go stale quickly as new packages add features and ABI requirements.

## NVSwitch Chip vs NVLink Switch Tray – Do Not Confuse

- **NVSwitch chip** — silicon on the GPU baseboard. Inventory path: /redfish/v1/UpdateService/FirmwareInventory/HGX\_FW\_NVSwitch\_<n>. Updated via the compute tray BMC as part of the GPU tray package.
- **NVLink Switch tray** — separate rack-level switch chassis in GB200 NVL72 racks. Inventory path: /redfish/v1/UpdateService/FirmwareInventory/NVSwitch\_Tray\_\*. Updated via the **NVLink Switch tray BMC**, not the compute tray BMC, with its own .fwpkg.

Selecting the wrong target will return a Redfish 404 and silently skip the component.

## Time Budget (typical per-node)

Phase	Estimated Time
Pre-upgrade preparation (Phase 1)	20 min
BMC update + reset + verify (2.1-2.2)	10 min
Motherboard tray update (2.3)	15 min
GPU tray update (2.4)	25 min
BackgroundCopyStatus poll (2.5)	10-20 min
Chassis power cycle + recheck (2.6-2.7)	10 min
CommitImage (2.8)	2 min
AC power cycle (2.9, includes 6m drain)	20 min
NIC / TPM / NVMe + chassis cycle (2.10)	15 min
Validation (Phase 3)	15 min
<b>Total — Hopper DGX/HGX</b>	<b>~2h 30m</b>
GB200 NVL72 rack (add HMC, NVLink Switch tray, rack PDU cycle 10m+15m boot)	<b>add 1h 15m</b>

Schedule a **3-hour window for Hopper**, **4-hour window for Blackwell B200**, **5-hour window for GB200 NVL72**.

## 3. Tools

### Tool Overview

Tool	Purpose	Where It Runs
<b>nvwupd</b>	Full firmware bundle update for DGX/HGX via BMC Redfish API	Management host (remote)

Tool	Purpose	Where It Runs
<b>NVFlash</b>	VBIOs-only flash for standalone PCIe GPUs	GPU host (root required)
<b>nvidia-smi</b>	Query GPU health, firmware version, running processes	GPU host
<b>nvsm</b>	DGX system-wide health check	GPU host
<b>ipmitool</b>	Remote power cycle and BMC reset	Management host
<b>curl</b>	Redfish polling (BackgroundCopyStatus, CommitImage)	Management host

## nvfwupd Version Requirement

- **Hopper (H100/H200) minimum: nvfwupd 2.0.4** — earlier versions do not support CommitImage or the HGX\_FW\_\* inventory paths.
- **Blackwell (B200 / GB200 / GB300) minimum: nvfwupd 3.0.0** — required for HMC, Grace CPU, and NVLink Switch tray Redfish paths.
- Confirm with: `nvfwupd -version`
- Download from the NVIDIA Enterprise Support Portal.

## Key Commands Reference

```
# --- nvfwupd ---
nvfwupd --version # MUST be >=
2.0.4

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version -p
<pkg.fwpkg>

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> update_fw -p <pkg.fwpkg>
-y
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> update_fw -p <pkg.fwpkg>
-y -s selector.json

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update enable
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update status
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> force_update disable

# --- NVFlash ---
sudo nvflash --list
sudo nvflash --version
sudo nvflash --index=0 --save backup_$(date +%Y%m%d).rom
sudo nvflash --index=0 new_firmware.rom

# --- nvidia-smi ---
nvidia-smi -L
nvidia-smi -q
nvidia-smi -q | grep "VBIOs Version"
nvidia-smi --query-compute-apps=pid,name,used_memory --format=csv
nvidia-smi nvlink -s -i 0

# --- ipmitool ---
```

```
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> chassis power cycle
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> mc reset cold
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> mc info
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> sel list

# --- nvsm ---
nvsm show health
```

## 4. Prerequisites

<note important> Complete every item below before starting. Do not proceed if any item cannot be confirmed. </note>

### Access & Credentials

- [ ] BMC IP address, username, and password (administrator privileges)
- [ ] SSH access to the target server
- [ ] **Physical or remote-hands access for the AC power cycle step**
- [ ] Access to NVIDIA Enterprise Support Portal

### Tooling

- [ ] **nvfwupd**  $\geq$  **2.0.4** installed on management host
- [ ] ipmitool, curl, jq available on management host

### Firmware Package

- [ ] Correct .fwpkg for your specific platform (GPU tray + motherboard tray packages)
- [ ] SHA256 checksum verified against release notes
- [ ] Per-package release notes reviewed for driver minimums and known issues
- [ ] **Previous .fwpkg saved for rollback**

### BMC Version Hop (pre-1.1.3 systems)

- [ ] If current BMC firmware is **older than 1.1.3**, you **must** first update to the 1.1.3 intermediate package, reset BMC, and verify, before installing any 1.2.x+ bundle. Skipping this hop will brick the Redfish endpoint.
- [ ] Confirm current BMC version: `nvfwupd ... show_version | grep -i bmc`

### Environment

- [ ] All GPU workloads stopped
- [ ] `nvidia-smi` is not running (passive queries block VBIOS updates)

- [ ] Maintenance window: **3h Hopper / 4h B200 / 5h GB200 NVL72** (see Time Budget in Section 2)
- [ ] VBIOS backed up via NVFlash (standalone PCIe only)
- [ ] **MIG disabled** on all GPUs: `sudo nvidia-smi -mig 0` (some packages refuse to flash with MIG instances active)
- [ ] **nvidia-fabricmanager stopped**: `sudo systemctl stop nvidia-fabricmanager` (DGX/HGX only; prevents dirty shutdown during chassis cycle)
- [ ] **Confidential Computing (CC) mode disabled** if previously enabled — some .fwpkg versions refuse VBIOS flash while CC is on. Check the package release notes.
- [ ] **PSU redundancy healthy**: `nvsm show health | grep -i psu` (all PSUs OK; PSU update will refuse if redundancy is degraded)
- [ ] **Driver minimum verified against this package's release notes**: `nvidia-smi -query-gpu=driver_version -format=csv,noheader ≥ the minimum listed in the .fwpkg release notes`
- [ ] **Pre-upgrade SEL captured**: `ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> sel list > preupgrade_sel_$(date +%Y%m%d).log`

## System State

- [ ] Management host can ping the BMC IP
- [ ] Server on stable power
- [ ] No other firmware operations running on the same server
- [ ] `nvsm show health` clean

## Rollback Readiness

- [ ] Current firmware versions recorded in change ticket
- [ ] Previous firmware .fwpkg saved
- [ ] Rollback procedure reviewed (Section 9)

## 5. Best Practices

#	Practice	Why It Matters
1	Always run <code>show_version</code> first	Confirms exactly what will change
2	Back up VBIOS before flashing	No backup = no rollback path
3	Stop all GPU processes including <code>nvidia-smi</code>	Active processes block or corrupt VBIOS updates
4	Never interrupt mid-flash	Closing terminal or losing power bricks the GPU
5	Verify the package checksum	A corrupted .fwpkg silently fails
6	Wait for <code>BackgroundCopyStatus: Completed</code>	Rebooting early means firmware never activates
7	<b>Always AC-cycle after CommitImage</b>	EROT and CPLD only activate on full AC removal
8	Use a wired management network	VPN or WiFi drops leave flash incomplete
9	Disable <code>ForceUpdate</code> immediately after downgrade	Leaving it enabled is a security risk

#	Practice	Why It Matters
10	Update BMC first, everything else after	Subsequent components depend on the new BMC Redfish schema
11	Validate before restoring workloads	Confirms upgrade succeeded
12	Document pre and post versions	Required for change management

## 6. Phase 1 — Pre-Upgrade Preparation

### Step 1.1 — Record Current Firmware Version

```
# Pin the full pre-upgrade output to a file for the change ticket
nvwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version \
| tee preupgrade_versions_$(date +%Y%m%d_%H%M).txt

# Standalone PCIe GPU
nvidia-smi -q | grep "VBIOS Version" | tee preupgrade_vbios_$(date
+%Y%m%d).txt
```

Copy full output to your change ticket and fill in:

Component	Pre-Upgrade Version	Target Version
BMC		
SBIOS		
CPLD		
VBIOS		
NVSwitch FW		
EROT FW		
FPGA FW		
PSU		
NIC		
TPM		
NVMe		

### Step 1.2 — Verify GPU Health

```
nvidia-smi -L
nvsm show health
```

### Step 1.3 — Stop All GPU Workloads

```
kubectl drain <node-name> --ignore-daemonsets --delete-emptydir-data
nvidia-smi --query-compute-apps=pid,name,used_memory --format=csv
# Output must be empty
```

## Step 1.3b — Disable MIG and Stop Fabric Manager

```
# Disable MIG on all GPUs (no-op if already disabled)
sudo nvidia-smi -mig 0

# Stop fabric manager (DGX/HGX/Blackwell) – prevents a dirty shutdown
# when the chassis is power-cycled in Phase 2.
sudo systemctl stop nvidia-fabricmanager
systemctl is-active nvidia-fabricmanager # must report "inactive"

# Confidential Computing – disable if currently enabled and the package
# release notes require it. (Skip if your environment never uses CC mode.)
# nvidia-smi conf-compute -srs 0
```

## Step 1.4 — Backup VBIOS (Standalone PCIe Only)

```
sudo nvflash --index=0 --save gpu0_backup_$(date +%Y%m%d).rom
```

## Step 1.5 — Verify Package Integrity

```
sha256sum nvfw_DGX-HGX-H100-H200x8_<version>.fwpkg
```

## Step 1.6 — Preview What Will Change

```
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version -p
<pkg.fwpkg>
```

---

# 7. Phase 2 — Firmware Upgrade Execution

<note warning> **Strict ordering — do not deviate.** BMC first, then motherboard tray, then GPU tray, then poll, then chassis power cycle, then CommitImage, then AC cycle, then NIC/TPM/NVMe.  
</note>

## 7.1 DGX / HGX — nvfwupd

### Step 2.1 — Update BMC

```
cat > selector_bmc.json <<'EOF'
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_BMC_0"
```

```
]
}
EOF

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> \
update_fw -p nvfw_DGX_<version>.fwpkg -y -s selector_bmc.json
```

## Step 2.2 — Reset the BMC

```
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> mc reset cold
sleep 180
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> mc info
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version | grep -i
bmc
```

Confirm the BMC reports the new version before continuing.

## Step 2.3 — Update Motherboard Tray (SBIOS, CPLD, PSU)

```
cat > selector_mb.json <<'EOF'
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_HostBIOS_0",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_CPLDMB_0",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_PSU_0",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_PSU_1",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_PSU_2",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_PSU_3"
  ]
}
EOF

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> \
update_fw -p nvfw_DGX_<version>.fwpkg -y -s selector_mb.json
```

## Step 2.4 — Update GPU Tray (VBIOS, NVSwitch, EROT, FPGA)

```
cat > selector_gpu.json <<'EOF'
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GPU_SXM_0",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GPU_SXM_1",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GPU_SXM_2",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GPU_SXM_3",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GPU_SXM_4",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GPU_SXM_5",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GPU_SXM_6",
  ]
}
```

```

"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GPU_SXM_7",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_NVSwitch_0",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_NVSwitch_1",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_NVSwitch_2",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_NVSwitch_3",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_ERoT_GPU_SXM_0",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_FPGA_0"
]
}
EOF

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> \
  update_fw -p nvfw_DGX-HGX-H100-H200x8_<version>.fwpkg -y -s
selector_gpu.json

```

### Step 2.5 — Wait for Background Copy to Complete

```

curl -k -u <USER>:<PASS> \
  https://<BMC_IP>/redfish/v1/Chassis/HGX_ERoT_BMC_0

```

Poll every 60 seconds until BackgroundCopyStatus reports Completed. If Failed — do **not** proceed; see Troubleshooting.

### Step 2.6 — Chassis Power Cycle

```

ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> chassis power cycle

```

Wait 5-10 minutes for the server to fully boot.

### Step 2.7 — Recheck Versions

```

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version

```

Confirm staged versions are visible on BMC, SBIOS, GPU tray components.

### Step 2.8 — CommitImage (BMC and SBIOS)

CommitImage promotes the staged image to the active slot. **Required for BMC and SBIOS.**

```

# Commit BMC
curl -k -u <USER>:<PASS> -X POST \
  -H "Content-Type: application/json" \
  -d
'{"Targets":["/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_BMC_0"]}' \
https://<BMC_IP>/redfish/v1/UpdateService/Actions/Oem/NvidiaUpdateService.Co

```

```
mmitImage

# Commit SBIOS
curl -k -u <USER>:<PASS> -X POST \
  -H "Content-Type: application/json" \
  -d
  '{"Targets":["/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_HostBIOS_0"
  ]}' \
  https://<BMC_IP>/redfish/v1/UpdateService/Actions/Oem/NvidiaUpdateService.Co
mmitImage
```

## Step 2.9 — AC Power Cycle (MANDATORY)

<note warning> This step is **non-negotiable** for EROT and CPLD activation. A chassis power cycle is **not** sufficient. </note>

1. Power off the host via OS shutdown or `ipmitool chassis power off`.
2. **Physically remove all AC power cords** (or open the PDU breakers feeding the chassis).
3. **Wait 5-6 minutes** to allow standby rails to fully discharge.
4. Reconnect AC power and power the server back on.
5. Wait 5-10 minutes for full boot.

## Step 2.10 — Update NIC, TPM, NVMe (and ConnectX-7/8 on Blackwell)

These components are independent of the GPU/EROT activation chain and are updated last.

### Hopper (DGX/HGX H100/H200):

```
cat > selector_periph.json <<'EOF'
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_NIC_0",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_NIC_1",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_TPM_0",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_NVMe_0"
  ]
}
EOF

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> \
  update_fw -p nvfw_DGX_<version>.fwpkg -y -s selector_periph.json
```

### Blackwell (B200 / GB200) — also update onboard ConnectX-7/8 NICs:

```
cat > selector_periph_bw.json <<'EOF'
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_ConnectX_0",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_ConnectX_1",

```

```
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_ConnectX_2",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_ConnectX_3",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_TPM_0",
"/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_NVMe_0"
]
}
EOF

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> servertype=GB200 \
update_fw -p nvfw_GB200_<version>.fwpkg -y -s selector_periph_bw.json
```

Then host reboot:

```
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> chassis power cycle
```

## 7.1B Blackwell – GB200 / B200 Additional Steps

Blackwell systems insert the following **before** Step 2.1 (BMC update) and **between** Step 2.2 and 2.3.

### Step 2.0a – Update HMC (GB200 only)

The HMC manages the GPU/Grace complex on each compute tray and must be updated **before** the chassis BMC.

```
cat > selector_hmc.json <<'EOF'
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_HMC_0"
  ]
}
EOF

nvfwupd -t ip=<HMC_IP> user=<USER> password=<PASS> servertype=GB200 \
update_fw -p nvfw_GB200_<version>.fwpkg -y -s selector_hmc.json
```

Reset HMC and confirm:

```
curl -k -u <USER>:<PASS> -X POST \
https://<HMC_IP>/redfish/v1/Managers/HGX_HMC_0/Actions/Manager.Reset \
-d '{"ResetType":"GracefulRestart"}'
sleep 180
nvfwupd -t ip=<HMC_IP> user=<USER> password=<PASS> show_version | grep -i
hmc
```

### Step 2.2b – Update Grace CPU Firmware (Blackwell)

Bundled with the SBIOS package; activates on chassis power cycle.

```
cat > selector_grace.json <<'EOF'
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GraceCPU_0",
    "/redfish/v1/UpdateService/FirmwareInventory/HGX_FW_GraceCPU_1"
  ]
}
EOF

nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> servertime=GB200 \
update_fw -p nvfw_GB200_<version>.fwpkg -y -s selector_grace.json
```

### Step 2.4b — Update NVLink Switch Tray (GB200 NVL72)

Run from the rack-level management host against the NVLink Switch tray BMC, **not** the compute tray BMC.

```
cat > selector_nvlsw.json <<'EOF'
{
  "Targets": [
    "/redfish/v1/UpdateService/FirmwareInventory/NVSwitch_Tray_BMC_0",
    "/redfish/v1/UpdateService/FirmwareInventory/NVSwitch_Tray_FPGA_0",
    "/redfish/v1/UpdateService/FirmwareInventory/NVSwitch_Tray_ERoT_0"
  ]
}
EOF

nvfwupd -t ip=<NVL_SW_BMC_IP> user=<USER> password=<PASS>
servertime=NVL_SWITCH \
update_fw -p nvfw_NVLSwitch_<version>.fwpkg -y -s selector_nvlsw.json
```

### Step 2.9b — Rack AC Cycle (GB200 NVL72)

For GB200 NVL72 the AC step is a **rack-level PDU cycle**, not a single chassis cord pull. Coordinate with DC operations:

1. Schedule a full rack power-down window.
2. Open both A-side and B-side PDU feeds (the rack is fed redundantly — pulling only one side leaves standby rails energised).
3. Wait **10 minutes** (longer than the 5–6 min for standalone DGX due to BBU/standby capacitance).
4. Restore power; allow 15 minutes for the full rack to come up before validation.

## 7.2 Standalone PCIe GPU — NVFlash

## Step 2.11 — Unload the NVIDIA Driver

```
sudo rmmod nvidia_uvm nvidia_drm nvidia_modeset nvidia
lsmod | grep nvidia
```

## Step 2.12 — Flash the VBIOS

```
sudo nvflash --index=0 new_vbios_firmware.rom
```

## Step 2.13 — Reboot

```
sudo reboot
```

---

# 8. Phase 3 — Validation

## Step 3.1 — Confirm New Firmware Version

```
nvfwupd -t ip=<BMC_IP> user=<USER> password=<PASS> show_version
nvidia-smi -q | grep "VBIOS Version"
```

Every component must report the new version. Pay particular attention to **EROT** and **CPLD** — if either still reports the old version, the AC power cycle was not effective; repeat Step 2.9.

## Step 3.2 — Verify GPU Detection

```
nvidia-smi -L
```

## Step 3.3 — Run Health Check

```
nvidia-smi -q
nvsm show health
```

## Step 3.4 — Check NVLink

```
nvidia-smi nvlink -s -i 0
```

## Step 3.5 — Smoke Test

```
python3 -c "import torch; print(torch.cuda.is_available());
print(torch.cuda.get_device_name(0))"
```

### Step 3.6 – Record Final State

Component	Pre-Upgrade	Post-Upgrade	Status
BMC			Pass / Fail
SBIOS			Pass / Fail
CPLD			Pass / Fail
VBIOS			Pass / Fail
NVSwitch FW			Pass / Fail
EROT FW			Pass / Fail
FPGA FW			Pass / Fail
PSU			Pass / Fail
NIC			Pass / Fail
TPM			Pass / Fail
NVMe			Pass / Fail

### Step 3.7 – Capture Post-Upgrade SEL and Restart Services

```
# Diff against the pre-upgrade SEL captured in Section 4
ipmitool -I lanplus -H <BMC_IP> -U <USER> -P <PASS> sel list \
  > postupgrade_sel_$(date +%Y%m%d).log
diff preupgrade_sel_*.log postupgrade_sel_*.log

# Restart fabric manager and confirm healthy
sudo systemctl start nvidia-fabricmanager
sudo systemctl status nvidia-fabricmanager # must be active (running)

# Re-enable MIG only if your workload requires it
# sudo nvidia-smi -mig 1
```

Investigate any new SEL entries (assertion events, thermal trips, fan failures) before restoring workloads.

### Step 3.8 – Restore Workloads

```
kubectl uncordon <node-name>
```

Monitor for 30–60 minutes.

## 9. Rollback / Downgrade

Same flow as the forward upgrade, but with the previous `.fwpkg` and `force_update enable set`

first. Disable `force_update` immediately after. EROT/CPLD downgrades also require the AC cycle.

## 10. Troubleshooting

Symptom	Cause	Action
Invalid firmware file	Wrong .fwpkg	Re-check <code>show_version -p</code> against the component
servertype not recognised / Redfish 404	nvwupd defaulting to wrong server type	Add <code>servertype=DGX</code> (or HGX) to the <code>-t</code> target string
VBIOS update blocked — GPU activity	Process still using the GPU	Kill PIDs from <code>nvidia-smi -query-compute-apps</code> ; drain K8s node
BackgroundCopyStatus: Failed	Wrong firmware file or transient BMC error	Do <b>not</b> reboot. Retry. Contact NVIDIA Support
EROT/CPLD version unchanged after reboot	AC power cycle skipped or too short	Repeat Step 2.9 — full AC removal, 5-6 minute wait
Server does not boot after AC cycle	Failed flash	BMC serial console + <code>ipmitool sel list</code> ; contact NVIDIA Support
GPU missing after upgrade	PCIe or driver issue	<code>lspci</code> , <code>dmesg</code> for Xid; reload driver
NVLink ports Inactive	NVSwitch FW or fabric manager	Cold reboot; restart <code>nvidia-fabricmanager</code>
Firmware unchanged after reboot	Rebooted before background copy finished	Re-run, wait for Completed
Downgrade not allowed	ForceUpdate flag not enabled	See Section 9
BMC 1.1.x → 1.2.x update bricks Redfish	Skipped the 1.1.3 intermediate hop	See “BMC Recovery” below; flash 1.1.3 first
PSU update refuses to start	PSU redundancy degraded	<code>nvsm show health   grep -i psu</code> ; replace failed PSU before retrying
Fabric Manager fails to start post-upgrade	NVSwitch FW / driver mismatch	Confirm driver ≥ release notes min; <code>journalctl -u nvidia-fabricmanager</code> ; reload <code>nvidia</code> module
New SEL entries after upgrade	Thermal/fan/PSU event during reboot	Investigate each entry; do not restore workloads until cleared

### BMC Recovery (Bricked BMC)

If the BMC is unreachable after an update (no Redfish, no IPMI, no SSH):

1. Connect to the **BMC serial console** via the rear-panel serial port or the integrated serial-over-USB on the management cable. Default baud: 115200 8N1.
2. At the BMC u-boot/recovery prompt, confirm you have the recovery shell — ``help`` should list ``flash``, ``boot``, ``reset``.
3. Prepare a **USB flash drive (FAT32)** containing the BMC recovery image (`bmc_recovery_<version>.img`) downloaded from the NVIDIA Enterprise Support Portal — file naming and partition layout differ between H100 and B200; use the recovery image that matches the **chassis SKU**, not the package version you were trying to install.
4. Insert the USB into the BMC's dedicated recovery USB port (rear panel, labelled BMC RECOVERY)

- **not** a host USB port).
- 5. From the recovery shell: `flash usb bmc_recovery_<version>.img` then `reset`.
- 6. Allow 5 min for the BMC to reboot; then re-establish IPMI and re-run the failed update **at the intermediate version** (e.g., 1.1.3) before re-attempting the target version.
- 7. Only proceed to NVIDIA Support if recovery flash itself fails — they will require the serial console log.

## 11. Risk Register

Risk	Likelihood	Impact	Mitigation
Wrong firmware file flashed	Medium	Critical	<code>show_version</code> first; checksum
Power loss during flash	Low	Critical	Stable power; avoid power maintenance windows
Workload active during upgrade	Medium	High	Verify <code>nvidia-smi</code> apps query empty
Driver / firmware mismatch	Medium	High	Check per-package release notes
Network loss to BMC mid-update	Low	High	Wired network only
Previous .fwpkg unavailable	Medium	High	Save before every upgrade
ForceUpdate left enabled	Low	Medium	Disable + verify after downgrade
AC cycle skipped — EROT/CPLD not active	High	High	AC step is mandatory in Phase 2
BMC version-hop skipped (pre-1.1.3 → 1.2.x)	Medium	Critical	Stage to 1.1.3 first
Reboot before background copy completes	Medium	High	Poll Redfish until Completed

## 12. References

- [NVIDIA DGX H100/H200 Firmware Update Guide](#)
- Per-package release notes — **always consult for driver minimums and known issues**
- [NVIDIA Enterprise Support Portal](#)