nvidia-smi GPU monitoring

START HERE

`nvidia-smi`
GPU summary

`nvidia-smi topo -m`
Link topology

`nvidia-smi -L`
List GPUs and UUIDs

`nvidia-smi -i <id> -q`
One GPU only

`nvidia-smi -l 2`
Refresh every 2 s

`nvidia-smi -q`
Full detail dump

`nvidia-smi pmon`
Per-process usage

`nvidia-smi dmon`
Rolling metrics

nvidia-smi summary table → `-q / --query-gpu` → **Detail or CSV** pick fields → `-l / dmon` → **Live monitor** loop or stream → `-pm / -pl` → **Tuning** needs root

CONCEPTS

Run with no arguments first: the table shows driver and CUDA versions, temperature, power, memory and processes per GPU.

Anything that changes state (persistence, power, clocks, reset) needs root and usually does not survive a reboot.

The GPU index order can differ from CUDA_VISIBLE_DEVICES order; use UUIDs for stable targeting.

INSPECT

`nvidia-smi`
Show GPU summary table

`nvidia-smi -L`
List GPUs with UUIDs

`nvidia-smi -q`
Show all details for every GPU

`nvidia-smi -i <id>`
Limit output to one GPU
<id> is an index, UUID, serial or PCI bus ID. Works with most options.

`nvidia-smi -q -d MEMORY,UTILIZATION`
Show only chosen sections
Other types: ECC, TEMPERATURE, POWER, CLOCK, PERFORMANCE.

INSPECT (cont.)

`nvidia-smi -q -x`
Print details as XML

`nvidia-smi -f <file> -q`
Write output to a file

`nvidia-smi --version`
Show tool and driver version

`nvidia-smi -h`
Print help

MONITOR

`nvidia-smi -l <sec>`
Repeat every N seconds

`nvidia-smi -lms <ms>`
Repeat every N milliseconds

`nvidia-smi dmon`
Stream power, temp, clocks, load

`nvidia-smi dmon -s um`
Choose metric groups to stream
Groups: p power/temp, u utilization, c clocks, v violations, m memory, e ECC, t PCIe.

`nvidia-smi pmon`
Show load per process

`nvidia-smi pmon -s um`
Per-process utilization and memory

`nvidia-smi dmon -c <n>`
Stop after N samples

MONITOR (cont.)

`watch -n 1 nvidia-smi`
Refresh the table in place

SCRIPTED QUERIES

`nvidia-smi --query-gpu=<fields> --format=csv`
Query chosen fields as CSV

`nvidia-smi --query-gpu=name,memory.used,memory.total --format=csv`
Memory use per GPU

`nvidia-smi --query-gpu=temperature.gpu --format=csv,noheader,nounits`
Bare numbers for scripts

`nvidia-smi --query-compute-apps=pid,process_name,used_memory --format=csv`
Compute processes and memory

`nvidia-smi --help-query-gpu`
List every queryable field

`nvidia-smi --query-gpu=<fields> --format=csv -l 1`
Log CSV every second

SCRIPTED QUERIES (cont.)

`nvidia-smi --query-supported-clocks=gr,mem --format=csv`
List supported clock pairs

`nvidia-smi --help-query-compute-apps`
List compute-app fields

PERSISTENCE AND COMPUTE MODE

`sudo nvidia-smi -pm 1`
Enable persistence mode
Keeps the driver loaded so launches are faster. Resets on reboot; NVIDIA recommends the persistence daemon.

`sudo nvidia-smi -c <mode>`
Set compute mode
DEFAULT, EXCLUSIVE_PROCESS or PROHIBITED. Resets on reboot.

`sudo nvidia-smi -r`
Reset the GPU
Fails if processes still use the GPU.

`sudo nvidia-smi -e <0|1>`
Disable or enable ECC
Takes effect after a reboot or reset.

`sudo nvidia-smi -am 1`
Turn on process accounting

POWER AND CLOCKS

`sudo nvidia-smi -pl <watts>`
Set power limit
Must stay within the range shown by -q -d POWER.

`sudo nvidia-smi -lgc <min>,<max>`
Lock GPU clock range in MHz

`sudo nvidia-smi -rgc`
Unlock GPU clocks

`sudo nvidia-smi -lmc <min>,<max>`
Lock memory clock range in MHz

`sudo nvidia-smi -rmc`
Unlock memory clocks

`nvidia-smi -q -d CLOCK`
Show current and max clocks

`nvidia-smi -q -d PERFORMANCE`
Show throttle reasons

TOPOLOGY AND OTHERS

`nvidia-smi topo -m`
Show GPU interconnect matrix
Shows NVLink, PCIe and CPU affinity between devices.

`nvidia-smi nvlink -s`
Show NVLink status

TOPOLOGY AND OTHERS (cont.)

`nvidia-smi mig -lgi`
List MIG GPU instances
MIG is Ampere and newer; enabling it needs -mig 1 plus a reset.

`nvidia-smi -mig 1`
Enable MIG mode

`nvidia-smi drain -h`
Show GPU drain options

RECIPES

Log GPU load to a CSV

```
nvidia-smi --query-gpu=
timestamp,utilization.gpu,
memory.used,power.draw \
  --format=csv,noheader -l 5 >>
gpu.csv
```

Who is using the GPU?

```
nvidia-smi --query-compute-
apps=pid,used_memory --
format=csv,noheader
ps -o pid,user,cmd -p <pid>
```

Prepare a shared box

```
sudo nvidia-smi -pm 1
sudo nvidia-smi -pl 250
nvidia-smi -q -d POWER | head
-20
```

Watch one GPU in a terminal

```
nvidia-smi -i 0 -l 1 \
  --query-gpu=temperature.gpu,
utilization.gpu,memory.used --
format=csv
```