

```

for /L %i in ( ) do @(cls & nvidia-smi & timeout /t 1 > nul)

from google.colab import files

files.download("gpu_usage.csv")

import os
import time
import pandas as pd

data = []

while True:
    gpu_info = os.popen("nvidia-smi --query-gpu=temperature.gpu,utilization.memory
--format=csv,noheader,nounits").read().strip().split(",")
    timestamp = time.time()
    temperature = int(gpu_info[0])
    memory_usage = int(gpu_info[1])

    data.append([temperature, memory_usage, timestamp])

    time.sleep(1)

    if len(data) >= 10:
        break

df = pd.DataFrame(data, columns=["Temperature (C)", "Memory Usage (MB)",
"Timestamp"])
df.to_csv("gpu_usage.csv", index=False)

import os
import time

while True:
    os.system("nvidia-smi")
    time.sleep(1)

```

[Chat history](#)
[New chat](#)
[Hide sidebar](#)
[Today](#)
[Watch n list NVIDIA cmd](#)

Yesterday
Enable RTX GPU Rendering
LiquidCrystal Library Fix
Previous 7 Days
Arduino Bluetooth Dot Matrix
June
DHT Control with NodeMCU & Blynk
Arduino AI Limitations
Arduino Decision Tree Model
May
Order Falooda
I2C LCD with Arduino
Arduino Human-Following Robot
AWS EventBridge Cron Job
Digital Twins Explained
NodeMCU Cost: \$5-\$15
Drums: No Musical Notes.
Summarize Request and Response.
Largest Kolam Record
Detect Water Flow
Upload SSL Certificates
Bluetooth Robotic Arm Code
Gesture Robotic Arm
Kinetic Energy Simplified
New chat
Big Bang Theory Explanation
Arduino Board Manager location.
Arduino Snake Game.
PC Boredom Solutions
Black Hole Research
Arduino Dino Game LCD.
Upgrade to Plus
NEW

User
Kavvin bbr

User
watch n list nvidia command in cmd prompt
ChatGPT
To watch and list NVIDIA devices using the command prompt, you can utilize the
NVIDIA System Management Interface (nvidia-smi) command. Here's how you can do it:

Open the command prompt:

Press the Windows key.
Type "cmd" (without quotes).
Press Enter.

Run the `nvidia-smi` command:

In the command prompt, type "`nvidia-smi`" (without quotes).

Press Enter.

The `nvidia-smi` command will display information about the NVIDIA GPUs installed on your system, including their current utilization, temperature, memory usage, and more. It provides a real-time snapshot of the GPU status.

Note: Make sure you have the NVIDIA display driver and the NVIDIA System Management Interface installed on your system for the `nvidia-smi` command to work.

User

immean to monitor gpu usage give me acommandnd to monition evry 1 second

ChatGPT

To monitor GPU usage and update the information every 1 second using the command prompt, you can combine the `nvidia-smi` command with the `timeout` command. Here's the command you can use:

```
shell
```

```
Copy code
```

```
:monitor
```

```
cls
```

```
nvidia-smi
```

```
timeout /t 1 /nobreak > nul
```

```
goto monitor
```

Here's a breakdown of how the command works:

The `:monitor` label creates a named section in the batch script.

`cls` clears the command prompt window before displaying the new GPU information.

`nvidia-smi` displays the current GPU usage and information.

`timeout /t 1 /nobreak` waits for 1 second before executing the next command. The `/nobreak` option prevents the user from interrupting the script by pressing a key.

`> nul` redirects the output of the `timeout` command to null, so it doesn't display any messages.

`goto monitor` sends the script back to the `:monitor` label, creating an infinite loop that updates the GPU information every second.

To use this command, follow these steps:

Open the command prompt.

Copy and paste the command into the command prompt.

Press Enter to start monitoring the GPU usage.

The command prompt window will continuously update with the latest GPU information every second until you manually stop the process by closing the window or pressing `Ctrl+C`.

User

give me a single command

ChatGPT

Certainly! Here's a single command that combines the `nvidia-smi` command with a loop to monitor GPU usage and update the information every 1 second:

shell

Copy code

```
for /L %i in ( ) do @cls & nvidia-smi & timeout /t 1 > nul)
```

To use this command, open the command prompt, copy and paste the command into the prompt, and press Enter. The command will continuously update the GPU information every second until you manually stop the process by pressing Ctrl+C.

Note: If you're using this command within a batch file, you'll need to use `%i` instead of `%i` in the command.

User

give for colab gpu monitoring

ChatGPT

To monitor GPU usage in a Google Colab notebook, you can use the `nvidia-smi` command within a loop. Here's an example code snippet that you can use:

python

Copy code

```
import os
```

```
import time
```

```
while True:
```

```
    os.system("nvidia-smi")
```

```
    time.sleep(1)
```

Here's how to use it:

Open a new Colab notebook.

Create a new code cell.

Copy and paste the code snippet into the cell.

Run the cell.

The code will continuously execute the `nvidia-smi` command every 1 second and display the GPU usage information in the notebook's output.

Note: Google Colab provides a limited GPU access, and the `nvidia-smi` command might not work in all cases, especially if the Colab runtime environment doesn't have a GPU available. However, if a GPU is assigned to your Colab session, this method should work for monitoring the GPU usage.

User

i need with graph csv export on tempereature vs usage mb and time

ChatGPT

To monitor GPU temperature and usage in a Google Colab notebook and export the data as a CSV file with temperature, usage in MB, and timestamp, you can use the `nvidia-smi` command in combination with the pandas library. Here's an example code snippet:

python

Copy code

```
import os
import time
import pandas as pd
```

```
data = []
```

```
while True:
```

```
    gpu_info = os.popen("nvidia-smi --query-gpu=temperature.gpu,utilization.memory
--format=csv,noheader,nounits").read().strip().split(",")
    timestamp = time.time()
    temperature = int(gpu_info[0])
    memory_usage = int(gpu_info[1])
```

```
    data.append([temperature, memory_usage, timestamp])
```

```
    time.sleep(1)
```

```
    if len(data) >= 10:
        break
```

```
df = pd.DataFrame(data, columns=["Temperature (C)", "Memory Usage (MB)",
"Timestamp"])
```

```
df.to_csv("gpu_usage.csv", index=False)
```

Here's how to use it:

Open a new Colab notebook.

Create a new code cell.

Copy and paste the code snippet into the cell.

Run the cell.

The code will continuously monitor the GPU temperature and usage every 1 second for a duration of 10 seconds (you can adjust this by modifying the condition in the if statement). It will store the data in a DataFrame and then export it to a CSV file named "gpu_usage.csv" in the current directory.

After running the code, you can download the CSV file by using the following code

snippet in a new cell:

python

Copy code

```
from google.colab import files
```

```
files.download("gpu_usage.csv")
```

This will enable you to download the generated CSV file containing the GPU temperature, memory usage, and timestamp information.

Regenerate response

Send a message

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT May 24 Version
ChatGPT