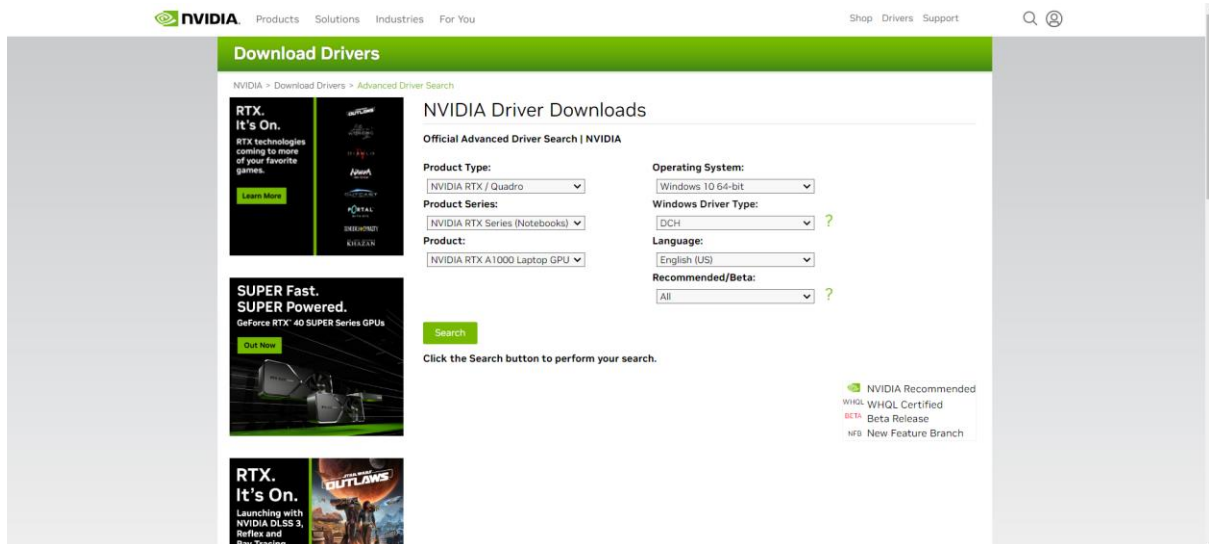


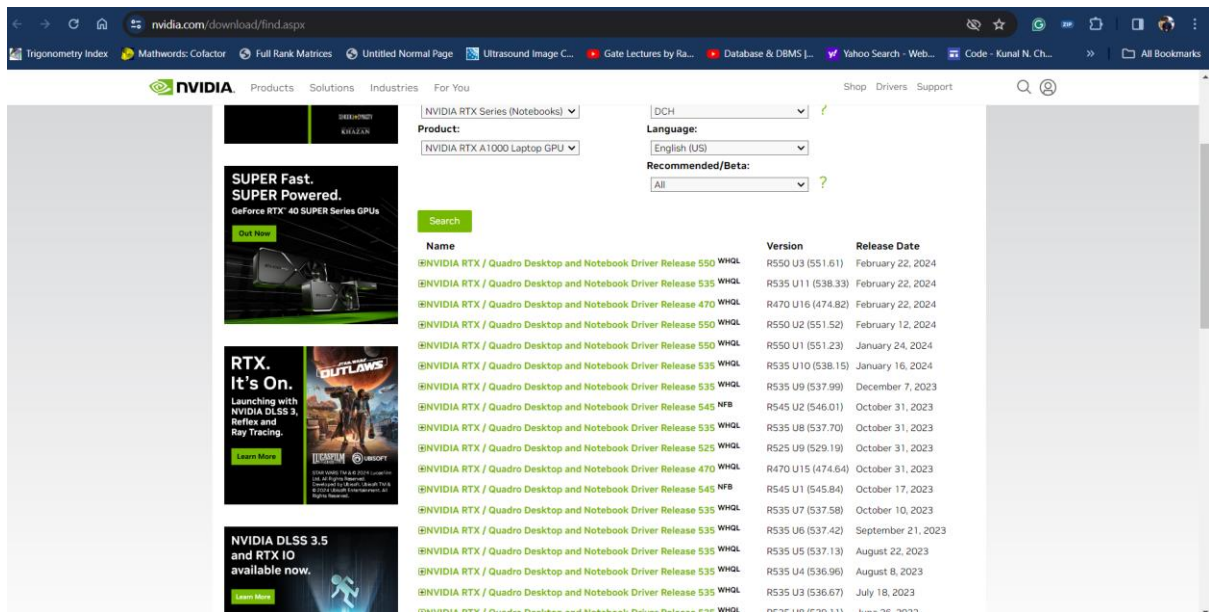
## Nvidia-Driver Installation:

Link: <https://www.nvidia.com/download/find.aspx>

Click on Link and select the appropriate option as shown below:



Click on search, it will give you list of available versions of driver.



Download the driver and install it in your system. In our case we have selected **551.52** versions

## Cuda Toolkit Installation:

Link: [https://developer.nvidia.com/cuda-downloads?target\\_os=Linux&target\\_arch=x86\\_64&Distribution=WSL-Ubuntu&target\\_version=2.0&target\\_type=deb\\_network](https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&Distribution=WSL-Ubuntu&target_version=2.0&target_type=deb_network)

Select the appropriate options:

Operating System	Linux	Windows																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
------------------	-------	---------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

It will give you to download and install appropriate driver. As mentioned in the screen shot.

Download Installer for Linux WSL-Ubuntu 2.0 x86\_64

The base installer is available for download below.

> Base Installer

Installation Instructions:

```

$ wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb
$ sudo dpkg -i cuda-keyring_1.1-1_all.deb
$ sudo apt-get update
$ sudo apt-get -y install cuda-toolkit-12-4

```

Additional installation options are detailed [here](#).

The CUDA Toolkit contains Open-Source Software. The source code can be found [here](#).  
The checksums for the installer and patches can be found in [Installer Checksums](#).  
For further information, see the [Installation Guide for Linux](#) and the [CUDA Quick Start Guide](#).

Follow the command to download and install cuda toolkit.

## Base Installer:

### Installation Instructions:

- `wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb`
- `sudo dpkg -i cuda-keyring_1.1-1_all.deb`
- `sudo apt-get update`
- `sudo apt-get -y install cuda-toolkit-12-4`

Additional installation options are detailed [here](#).

## Docker Installation in WSL

Link1:

[https://docs.rapids.ai/install?\\_gl=1\\*s0kbz1\\*\\_ga\\*ODkyMzg4MDA5LjE3MTA1OTAxMjY.\\*\\_ga\\_RKXFV6CM42\\*MTcxMDY1MDc2Mi4zLjAuMTcxMDY1MDc2Mi42MC4wLjA.#wsl2-docker](https://docs.rapids.ai/install?_gl=1*s0kbz1*_ga*ODkyMzg4MDA5LjE3MTA1OTAxMjY.*_ga_RKXFV6CM42*MTcxMDY1MDc2Mi4zLjAuMTcxMDY1MDc2Mi42MC4wLjA.#wsl2-docker)

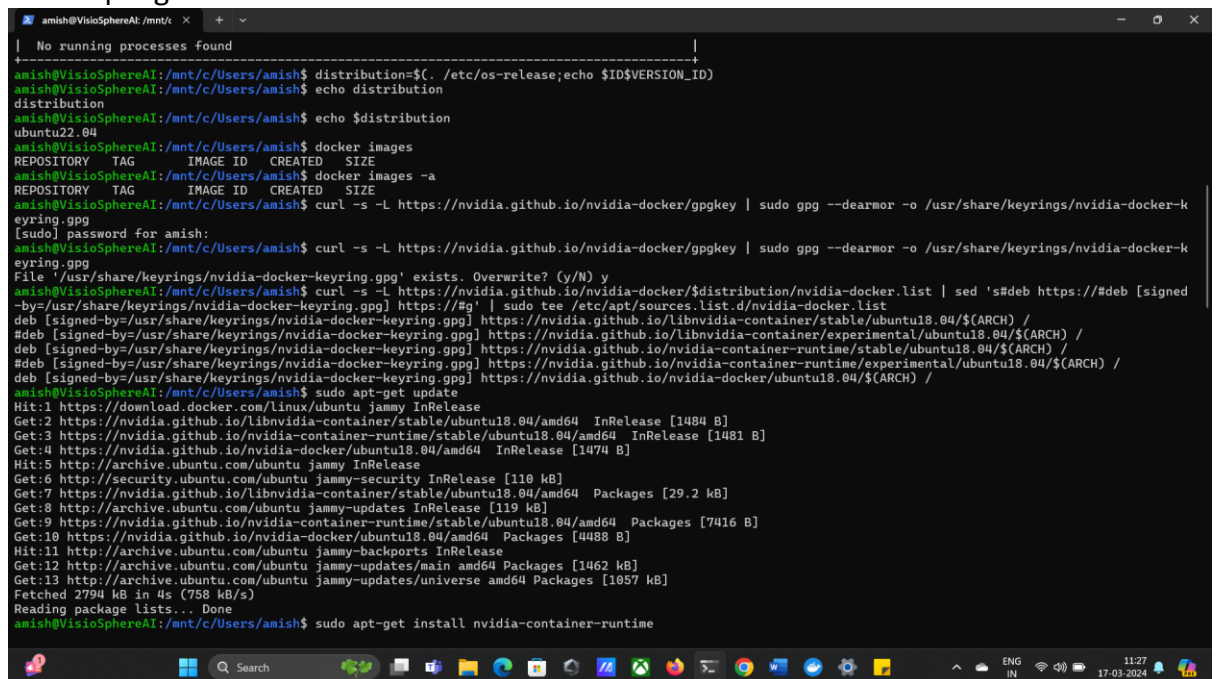
Link 2: <https://learn.microsoft.com/en-us/windows/wsl/tutorials/gpu-compute>

**Download and Install.** Copy command below to download and install the latest:

```
curl https://get.docker.com | sh
```

**Install Latest NVIDIA Docker.** Select the [appropriate supported distribution](#):

- `distribution=$(. /etc/os-release;echo $ID$VERSION_ID)`
- `curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-docker-keyring.gpg`
- `curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-docker-keyring.gpg] https://#g' | sudo tee /etc/apt/sources.list.d/nvidia-docker.list`
- `sudo apt-get update`
- `sudo apt-get install nvidia-container-runtime`



```
amish@VisioSphereAI: /mnt/c/Users/amish$ distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
amish@VisioSphereAI: /mnt/c/Users/amish$ echo distribution
distribution
amish@VisioSphereAI: /mnt/c/Users/amish$ echo $distribution
ubuntu22.04
amish@VisioSphereAI: /mnt/c/Users/amish$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
amish@VisioSphereAI: /mnt/c/Users/amish$ docker images -a
REPOSITORY TAG IMAGE ID CREATED SIZE
amish@VisioSphereAI: /mnt/c/Users/amish$ curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-docker-keyring.gpg
[sudo] password for amish:
amish@VisioSphereAI: /mnt/c/Users/amish$ curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-docker-keyring.gpg
File '/usr/share/keyrings/nvidia-docker-keyring.gpg' exists. Overwrite? (y/N) y
amish@VisioSphereAI: /mnt/c/Users/amish$ curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-docker-keyring.gpg] https://#g' | sudo tee /etc/apt/sources.list.d/nvidia-docker.list
deb [signed-by=/usr/share/keyrings/nvidia-docker-keyring.gpg] https://nvidia.github.io/libnvidia-container/stable/ubuntu18.04/${ARCH} /
#deb [signed-by=/usr/share/keyrings/nvidia-docker-keyring.gpg] https://nvidia.github.io/libnvidia-container/experimental/ubuntu18.04/${ARCH} /
deb [signed-by=/usr/share/keyrings/nvidia-docker-keyring.gpg] https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu18.04/${ARCH} /
#deb [signed-by=/usr/share/keyrings/nvidia-docker-keyring.gpg] https://nvidia.github.io/nvidia-container-runtime/experimental/ubuntu18.04/${ARCH} /
amish@VisioSphereAI: /mnt/c/Users/amish$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Get:2 https://nvidia.github.io/libnvidia-container/stable/ubuntu18.04/amd64 InRelease [1484 B]
Get:3 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu18.04/amd64 InRelease [1481 B]
Get:4 https://nvidia.github.io/nvidia-docker/ubuntu18.04/amd64 InRelease [1474 B]
Hit:5 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:6 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:7 https://nvidia.github.io/libnvidia-container/stable/ubuntu18.04/amd64 Packages [29.2 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:9 https://nvidia.github.io/nvidia-container-runtime/stable/ubuntu18.04/amd64 Packages [7416 B]
Get:10 https://nvidia.github.io/nvidia-docker/ubuntu18.04/amd64 Packages [4488 B]
Hit:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1462 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1057 kB]
Fetched 2794 kB in 4s (758 kB/s)
Reading package lists... Done
amish@VisioSphereAI: /mnt/c/Users/amish$ sudo apt-get install nvidia-container-runtime
```

**Start Docker.** In new terminal window run:

- `sudo service docker stop.`
- `sudo service docker start.`



```
amish@VisioSphereAI: /mnt/c/Users/amish$ sudo service docker stop
Warning: Stopping docker.service, but it can still be activated by:
        docker.socket
amish@VisioSphereAI: /mnt/c/Users/amish$ sudo service docker start
amish@VisioSphereAI: /mnt/c/Users/amish$ docker run --gpus all nvcr.io/nvidia/k8s/cuda-sample:nbody nbody --gpu-benchmark
```

**Test NVIDIA Docker.** In a terminal window run:

```
docker run --gpus all nvcr.io/nvidia/k8s/cuda-sample:nbody nbody -gpu -benchmark
```

- It will Check the docker image available in your local, if it is available, it will run it.
- If specific docker image is not available, it will pull it.

```

amish@VisioSphereAI:/mnt/c/Users/amish$ docker run --gpus all nvcr.io/nvidia/k8s/cuda-sample:nbody nbody -gpu -benchmark
Unable to find image 'nvcr.io/nvidia/k8s/cuda-sample:nbody' locally
nbody: Pulling from nvidia/k8s/cuda-sample
22c5ef69a68e: Pull complete
1939e42f8814: Pull complete
548afb82c856: Pull complete
a424d45fd86f: Pull complete
207b64ab7ce6: Pull complete
                                         e9bff09d04df: Pull complete
                                         2b60900a3ea5: Pull complete
                                         f65423f1b49b: Pull complete

edc14edf1b04: Pull complete
1f37f461c076: Pull complete
9026fb14bf88: Pull complete
Digest: sha256:59261e419d6d48a772aad5bb213f9f1588fcd042b115ceb7166c89a51f03363
Status: Downloaded newer image for nvcr.io/nvidia/k8s/cuda-sample:nbody
Run "nbody -benchmark [-numbodies=<numBodies>]" to measure performance.
  -fullscreen      (run n-body simulation in fullscreen mode)
  -fp64            (use double precision floating point values for simulation)
  -hostmem         (stores simulation data in host memory)
  -benchmark       (run benchmark to measure performance)
  -numbodies=<N>   (number of bodies (>= 1) to run in simulation)
  -device=<d>      (where d=0,1,2,... for the CUDA device to use)
  -numdevices=<i>  (where i=(number of CUDA devices > 0) to use for simulation)
  -compare         (compares simulation results running once on the default GPU and once on the CPU)
  -cpu            (run n-body simulation on the CPU)
  -tipsy=<file.bin> (load a tipsy model file for simulation)

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.

> Windowed mode
> Simulation data stored in video memory
> Single precision floating point simulation
> 1 Devices used for simulation
MapSMtoCores for SM 8.9 is undefined. Default to use 128 Cores/SM
MapSMtoArchName for SM 8.9 is undefined. Default to use Ampere
GPU Device 0: "Ampere" with compute capability 8.9

> Compute 8.9 CUDA device: [NVIDIA GeForce RTX 4060 Laptop GPU]
24576 bodies, total time for 10 iterations: 15.879 ms
= 380.360 billion interactions per second
= 7607.197 single-precision GFLOP/s at 20 flops per interaction

```