Running a Sample Workload

- Running a Sample Workload with Docker
- Running a Sample Workload with Podman
- Running Sample Workloads with containerd or CRI-O

Running a Sample Workload with Docker

After you install and configure the toolkit and install an NVIDIA GPU Driver, you can verify your installation by running a sample workload.

- Run a sample CUDA container:

```
sudo docker run --rm --runtime=nvidia --gpus all ubuntu nvidia-smi
```

Your output should resemble the following output:

```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.86.10    Driver Version: 535.86.10    CUDA Version: 12.2     |
|-----------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                                   |                      |               MIG M. |
|===================================+======================+======================|
|   0  Tesla T4            On    | 00000000:00:1E.0 Off |                    0 |
| N/A   34C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                                   |                      |                  N/A |
+-----------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

Running a Sample Workload with Podman

After you install and configura the toolkit (including generating a CDI specification) and install an NVIDIA GPU Driver, you can verify your installation by running a sample workload.

› Run a sample CUDA container:

```
podman run --rm --security-opt=label=disable \
   --device=nvidia.com/gpu=all \
   ubuntu nvidia-smi
```

Your output should resemble the following output:

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.86.10    Driver Version: 535.86.10    CUDA Version: 12.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            On   | 00000000:00:1E.0 Off |                    0 |
| N/A   34C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Running Sample Workloads with containerd or CRI-O

These runtimes are more common with Kubernetes than desktop computing. Refer to About the NVIDIA GPU Operator in the NVIDIA GPU Operator documentation for more information.