Run Docker Compose services with GPU access

- Enabling GPU access to service containers
  - Example of a Compose file for running a service with access to 1 GPU device
- Access specific devices

Compose services can define GPU device reservations if the Docker host contains such devices and the Docker Daemon is set accordingly. For this, make sure you install the prerequisites if you haven't already done so.

The examples in the following sections focus specifically on providing service containers access to GPU devices with Docker Compose.

Enabling GPU access to service containers

GPUs are referenced in a `compose.yaml` file using the device attribute from the Compose Deploy specification, within your services that need them.

This provides more granular control over a GPU reservation as custom values can be set for the following device properties:

- `capabilities`. This value is specified as a list of strings. For example, `capabilities: [gpu]`. You must set this field in the Compose file. Otherwise, it returns an error on service deployment.
- `count`. Specified as an integer or the value `all`, represents the number of GPU devices that should be reserved (providing the host holds that number of GPUs). If `count` is set to `all` or not specified, all GPUs available on the host are used by default.
- `device_ids`. This value, specified as a list of strings, represents GPU device IDs from the host. You can find the device ID in the output of `nvidia-smi` on the host. If no `device_ids` are set, all GPUs available on the host are used by default.

- `driver`. Specified as a string, for example `driver: 'nvidia'`
- `options`. Key-value pairs representing driver specific options.

> You must set the `capabilities` field. Otherwise, it returns an error on service deployment.

> `count` and `device_ids` are mutually exclusive. You must only define one field at a time.

For more information on these properties, see the Compose Deploy Specification.

Example of a Compose file for running a service with access to 1 GPU device

```
services:
  test:
    image: nvidia/cuda:12.9.0-base-ubuntu22.04
    command: nvidia-smi
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: 1
              capabilities: [gpu]
```

Run with Docker Compose:

```
$ docker compose up
Creating network "gpu_default" with the default driver
Creating gpu_test_1 ... done
Attaching to gpu_test_1
test_1  | +-----------------------------------------------------------------------------+
test_1  | | NVIDIA-SMI 450.80.02    Driver Version: 450.80.02    CUDA Version: 11.1     |
test_1  | |-------------------------------+----------------------+----------------------+
test_1  | | GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
test_1  | | Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
test_1  | |                               |                      |               MIG M. |
test_1  | |===============================+======================+======================|
```

```
test_1  | |   0  Tesla T4            On   | 00000000:00:1E.0 Off |                    0 |
test_1  | | N/A   23C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
test_1  | |                               |                      |                  N/A |
test_1  | +-------------------------------+----------------------+----------------------+
test_1  |
test_1  | +-----------------------------------------------------------------------------+
test_1  | | Processes:                                                                  |
test_1  | |  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
test_1  | |        ID   ID                                                   Usage      |
test_1  | |=============================================================================|
test_1  | |  No running processes found                                                 |
test_1  | +-----------------------------------------------------------------------------+
gpu_test_1 exited with code 0
```

On machines hosting multiple GPUs, the `device_ids` field can be set to target specific GPU devices and `count` can be used to limit the number of GPU devices assigned to a service container.

You can use `count` or `device_ids` in each of your service definitions. An error is returned if you try to combine both, specify an invalid device ID, or use a value of count that’s higher than the number of GPUs in your system.

```
$ nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.80.02    Driver Version: 450.80.02    CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| 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   | 00000000:00:1B.0 Off |                    0 |
| N/A   72C    P8    12W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            On   | 00000000:00:1C.0 Off |                    0 |
| N/A   67C    P8    11W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   2  Tesla T4            On   | 00000000:00:1D.0 Off |                    0 |
| N/A   74C    P8    12W /  70W |      0MiB / 15109MiB |      0%      Default |
```

```
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   3  Tesla T4            On   | 00000000:00:1E.0 Off |                    0 |
| N/A   62C    P8    11W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

Access specific devices

To allow access only to GPU-0 and GPU-3 devices:

```
services:
  test:
    image: tensorflow/tensorflow:latest-gpu
    command: python -c "import tensorflow as tf;tf.test.gpu_device_name()"
    deploy:
      resources:
        reservations:
          devices:
          - driver: nvidia
            device_ids: ['0', '3']
            capabilities: [gpu]
```

□□□ #2
□ □□□ □□□ 2026-10-31 19:08:04 UTC
□ □□□ □□□ 2026-10-31 19:08:05 UTC