CUDA DL

Release Notes

RN-11516-001_v25.02 | February 2025

Table of Contents

Chapter 1. Overview

NVIDIA's CUDA Deep Learning images extend the standard CUDA images by integrating enhanced networking support and a suite of optimized libraries—including cuDNN, cuTensor, NCCL, and HPC-x—to accelerate deep learning workloads. These images serve as a foundation for building your own GPU-accelerated containers.

Chapter 2. Pulling A Container

About this task

Release 25.02 is based on CUDA 12.8.0, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

Before you can pull a container from the NGC container registry:

- Install Docker.
  - For NVIDIA DGX™ users, see Preparing to use NVIDIA Containers Getting Started Guide.
  - For non-DGX users, see NVIDIA® GPU Cloud™ (NGC) container registry installation documentation based on your platform.
- Ensure that you have an NGC API Key to log in to the NGC container registry.

  Refer to NGC Getting Started Guide for more information.

Chapter 3. Running

Before you can run an NGC deep learning framework container, your Docker environment must support NVIDIA GPUs. To run a container, issue the appropriate command as explained in Running A Container and specify the registry, repository, and tags.

On a system with GPU support for NGC containers, when you run a container, the following occurs when running a container:

- The Docker engine loads the image into a container that runs the software.
- You define the container's runtime resources by including the additional flags and settings that are used with the command.

  These flags and settings are described in Running A Container.
- The GPUs are explicitly defined for the Docker® container, which defaults to all GPUs, but can be specified by using the `NVIDIA_VISIBLE_DEVICES` environment variable.

  For more information, refer to the nvidia-docker documentation.

Note: Starting in Docker 19.03, complete the steps below.

The method implemented in your system depends on the DGX OS version installed (for DGX systems), the specific NGC Cloud Image provided by a Cloud Service Provider, or the software that you have installed in preparation for running NGC containers on TITAN PCs, Quadro PCs, or vGPUs.

1. Issue the command for the applicable release of the container that you want.

   The following command assumes you want to pull the latest container:

   ```
   docker pull nvcr.io/nvidia/dgl:<xx.xx>-py3
   ```
2. Open a command prompt and paste the `pull` command.

   Ensure that the pull process completes successfully completes before proceeding to step 3.
3. Run the container image.

   To run the container, select one of the following modes:

   - **Interactive mode:**

If you have **Docker 19.03 or later**, a typical command to launch the container is:

```
docker run --gpus all -it --rm -v local_dir:container_dir nvcr.io/nvidia/dgl:<xx.xx>-py3
```

If you have **Docker 19.02 or earlier**, a typical command to launch the container is:

```
nvidia-docker run -it --rm -v local_dir:container_dir nvcr.io/nvidia/dgl:<xx.xx>-py3
```

- **Non-interactive mode:**

  If you have **Docker 19.03 or later**, a typical command to launch the container is:

  ```
  docker run --gpus all --rm -v local_dir:container_dir nvcr.io/nvidia/dgl:<xx.xx>-py3
   <command>
  ```

  If you have **Docker 19.02 or earlier**, a typical command to launch the container is:

  ```
  nvidia-docker run --rm -v local_dir:container_dir nvcr.io/nvidia/dgl:<xx.xx>-py3
   <command>
  ```

Note: If you use multiprocessing for multi-threaded data loaders, the default shared memory segment size that the container runs with might not be enough. To increase the shared memory size, run one of the following commands:

```
--ipc=host
```

or

```
--shm-size=<requested memory size>
```

in the command line to

```
docker run --gpus all
```

You might want to pull data and model descriptions from locations outside the container for use by DGL or save results to locations outside the container. The easiest method is to mount one or more host directories as Docker data volumes.

Chapter 4. Release 25.02

NVIDIA's CUDA Deep Learning images extend the standard CUDA images by integrating enhanced networking support and a suite of optimized libraries —including cuDNN, cuTensor, NCCL, and HPC-x—to accelerate deep learning workloads. These images serve as a foundation for building your own GPU-accelerated containers.

.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
https://www.nvidia.com

Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, DALI, DGX, DGX-1, DGX-2, DGX Station, DLProf, Jetson, Kepler, Maxwell, NCCL, Nsight Compute, Nsight Systems, NvCaffe, PerfWorks, Pascal, SDK Manager, Tegra, TensorRT, Triton Inference Server, Tesla, TF-TRT, and Volta are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
https://www.nvidia.com