Container

NVIDIA DGL

Deep Graph Library (DGL) is a Python package built for the implementation and training of graph neural networks on top of existing DL frameworks. The DGL NGC Container is built with the latest versions of DGL, PyTorch, and their dependencies.

NVIDIA AI Enterprise Supported

Get Container

25.08-py3 Signed

Overview Tags Layers Security Scanning

Publisher

NVIDIA

Latest Tag

25.08-py3

Updated

August 18, 2025

Compressed Size

11.11 GB

Multinode Support

Yes

Multi-Arch Support

Yes

Signed Images · Automotive / Transportation · DL · DLFW · ML

NSPECT-K7L7-L12G NVIDIA AI Enterprise Supported

What is inside this container?

Deep Graph Library (DGL) is a Python package built for the implementation and training of graph neural networks on top of existing DL frameworks. NGC Containers are the easiest way to get started with DGL. The DGL NGC Container is built with the latest versions of Deep Graph Library (DGL), PyTorch, and their dependencies.

The DGL NGC Container is optimized for GPU acceleration and contains a validated set of libraries that optimize GPU performance and software for accelerating data sampling and ETL:

- **CUDA**
- **cuBLAS**
- **NVIDIA cuDNN**
- **cuGraph**
- **NVIDIA NCCL**(optimized for **NVLink**)
- **NVIDIA NVSHMEM**
- **RAPIDS**
- **NVIDIA Data Loading Library (DALI)**
- **TensorRT**
- **Torch-TensorRT**

Use this link to access **Open Source Code**.

Prerequisites

There are two main prerequisites for DGL containers:

- **NVIDIA Drivers** NVIDIA Drivers 515.48.07+ is recommended. For a complete list of supported drivers (older versions), see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.
- **Docker (19.03+)**
- **(Optional) NGC API Key** for logging in to NVIDIA's registry. Details are available here.

Running the container

Use the following commands to run the container, where <xx.xx> is the container version.

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

For example, 24.07 for July 2024 release:

```
docker run --gpus all -it --rm nvcr.io/nvidia/dgl:24.07-py3
```

Running JupyterLab and examples

To start JupyterLab from the container and view all the included examples:

```
docker run --gpus all -it --rm -p 8888:8888 nvcr.io/nvidia/dgl:<xx.x
```

You might want to pull in your own data or persist code outside the DGL container. The easiest method is to mount one or more host directories as Docker bind mounts so your code changes persist.

We also have a GraphSAGE training example:

```
cd /workspace/examples/graphsage
python3 train_full.py --dataset cora --gpu 0
```

If you are looking for the original examples from DGL, you can find them in `/opt/dgl/dgl-source/`

Suggested Reading

For the latest Release Notes, see the DGL Release Notes.

For a full list of the supported software and specific versions that come packaged with this framework based on the container image, see the Frameworks Support Matrix. For more information about DGL, including tutorials, documentation, and examples, see:

- **DGL website**
- **DGL project**
- **DGL Documentation**
- **DGL Tutorials**

Security CVEs

To review known CVEs on this image, refer to the Security Scanning tab on this page.

Ethical AI

NVIDIA's platforms and application frameworks enable developers to build a wide array of AI applications. Consider potential algorithmic bias when choosing or creating the models being deployed. Work with the model's developer to ensure:

- The model meets the requirements for the relevant industry and use case
- The necessary instruction and documentation are provided to understand error rates, confidence intervals, and results
- The model is being used under the conditions and in the manner intended.

License

By pulling and using the container, you accept the terms and conditions of this End User License Agreement and Product-Specific Terms.