# TABLE OF CONTENTS

Chapter 1. MXNet Overview................................................................. 1
Chapter 2. Pulling A Container........................................................... 2
Chapter 3. Running MXNet................................................................. 3
Chapter 4. MXNet Release 18.05......................................................... 5
Chapter 5. MXNet Release 18.04......................................................... 7
Chapter 6. MXNet Release 18.03......................................................... 9
Chapter 7. MXNet Release 18.02....................................................... 11
Chapter 8. MXNet Release 18.01....................................................... 13
Chapter 9. MXNet Release 17.12....................................................... 15
Chapter 10. MXNet Release 17.11..................................................... 17
Chapter 11. MXNet Release 17.10..................................................... 19
Chapter 12. MXNet Release 17.09..................................................... 21
Chapter 13. MXNet Release 17.07..................................................... 23
Chapter 14. MXNet Release 17.06..................................................... 24
Chapter 15. MXNet Release 17.05..................................................... 25
Chapter 16. MXNet Release 17.04..................................................... 26
Chapter 17. MXNet Release 17.03..................................................... 27
Chapter 1.  
MXNET OVERVIEW

The NVIDIA Deep Learning SDK accelerates widely-used deep learning frameworks such as MXNet™.

MXNet is a deep learning framework designed for both efficiency and flexibility. It allows you to mix the flavors of symbolic programming and imperative programming to maximize efficiency and productivity.

In its core is a dynamic dependency scheduler that automatically parallelizes both symbolic and imperative operations on the fly. A graph optimization layer on top of that makes symbolic execution fast and memory efficient. The library is portable and lightweight, and it scales to multiple GPUs and multiple machines.

MXNet is also more than a deep learning project. It is also a collection of blueprints and guidelines for building deep learning systems and interesting insights of deep learning systems for hackers.

This document describes the key features, software enhancements and improvements, any known issues, and how to run this container.
Chapter 2.
PULLING A CONTAINER

You can access NVIDIA’s GPU accelerated containers for all three products, the NVIDIA DGX-1™, NVIDIA DGX Station™, and the NVIDIA® GPU Cloud™ (NGC). If you own a DGX-1 or DGX Station then you should use the NVIDIA® DGX™ container registry at https://compute.nvidia.com. This is a web interface to the Docker hub, nvcr.io (NVIDIA DGX container registry). You can pull the containers from there and you can also push containers there into your own account in the registry.

If you are accessing the NVIDIA containers from the NVIDIA® GPU Cloud™ (NGC) container registry via a cloud services provider such as Amazon® Web Services™ (AWS), then you should use NGC container registry at https://ngc.nvidia.com. This is also a web interface to the same Docker repository as for the DGX-1 and DGX Station. After you create an account, the commands to pull containers are the same as if you had a DGX-1 in your own data center. However, currently, you cannot save any containers to the NGC container registry. Instead you have to save the containers to your own Docker repository that is either on-premise or in the Cloud.

The containers are exactly the same, whether you pull them from the NVIDIA DGX container registry or the NGC container registry.

For all three products, the DGX-1, DGX Station, and the NVIDIA NGC Cloud Services, the location of the framework source is in /opt/<framework> in the container.

Before you can pull a container from the NGC container registry, you must have Docker and nvidia-docker installed as explained in Preparing to use NVIDIA Containers Getting Started Guide. You must also have access and logged into the NGC container registry as explained in the NGC Getting Started Guide.

For step-by-step instructions, see Container User Guide.
Chapter 3.
RUNNING MXNET

Before running the container, use the `docker pull` command to ensure an up-to-date image is installed. Once the pull is complete, you can run the container image.

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.

   ```bash
docker pull nvcr.io/nvidia/mxnet:18.05-py<x>
   ```

   Where `<x>` is the version of Python you want to pull.

2. Open a command prompt and paste the pull command. The pulling of the container image begins. Ensure the pull completes successfully before proceeding to the next step.

3. Run the container image. A typical command to launch the container is:

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

   Where:
   - `-it` means interactive
   - `--rm` means delete the container when finished
   - `-v` means mount directory
   - `local_dir` is the directory or file from your host system (absolute path) that you want to access from inside your container. For example, the `local_dir` in the following path is `/home/jsmith/data/mnist`.

   ```bash
   -v /home/jsmith/data/mnist:/data/mnist
   ```

   If you are inside the container, for example, `ls /data/mnist`, you will see the same files as if you issued the `ls /home/jsmith/data/mnist` command from outside the container.

   - `container_dir` is the target directory when you are inside your container. For example, `/data/mnist` is the target directory in the example:

   ```bash
   -v /home/jsmith/data/mnist:/data/mnist
   ```

   - `<xx.xx>` is the tag. For example, 18.01.
MXNet is run simply by importing it as a Python module:

```bash
$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import mxnet as mx
>>> a = mx.nd.ones((2,3), mx.gpu())
>>> print (a*2).asnumpy()
[[ 2.  2.  2.]
 [ 2.  2.  2.]]
```

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

In order to share data between ranks, NVIDIA® Collective Communications Library™ (NCCL) may require shared system memory for IPC and pinned (page-locked) system memory resources. The operating system’s limits on these resources may need to be increased accordingly. Refer to your system’s documentation for details.

In particular, Docker® containers default to limited shared and pinned memory resources. When using NCCL inside a container, it is recommended that you increase these resources by issuing:

```bash
--shm-size=1g --ulimit memlock=-1
```

in the command line to

```bash
nvidia-docker run
```

4. See `/workspace/README.md` inside the container for information on customizing your MXNet image.

For more information about MXNet, including tutorials, documentation, and examples, see:

- MXNet website
- MXNet project
Chapter 4.  
MXNET RELEASE 18.05

The NVIDIA container image of MXNet, release 18.05, is available.

Contents of MXNet

This container image contains the complete source of the version of MXNet in /opt/mxnet. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04

- Container image 18.05-py2 contains Python 2.7; 18.05-py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)

- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.2

- NCCL 2.1.15 (optimized for NVLink™)

- ONNX exporter for CNN classification models

The ONNX exporter is being continuously improved. You can try the latest changes by pulling from the master branch.

- Amazon Labs Sockeye sequence-to-sequence framework (for machine translation)

Driver Requirements

Release 18.05 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This MXNet release includes the following key features and enhancements.
MXNet container image version 18.05 is based on MXNet 1.1.0.
- For this month, no upstream merges as we work toward incorporating the upcoming MXNet 1.2.0 release.
- Ubuntu 16.04 with April 2018 updates

**Announcements**

Starting with the next major version of CUDA release, we will no longer provide Python 2 containers and will only maintain Python 3 containers.

**Known Issues**

Those wishing to run the MXNet unit tests under `/opt/mxnet/tests/python` should install SciPy using `pip install scipy==1.0`, as the recently available SciPy v1.1 is not compatible with all the unit tests. For more information, see Broken test_sparse_operator.test_sparse_mathematical_core with scipy 1.1.0.
Chapter 5.
MXNET RELEASE 18.04

The NVIDIA container image of MXNet, release 18.04, is available.

Contents of MXNet

This container image contains the complete source of the version of MXNet in \texttt{/opt/mxnet}. It is pre-built and installed to the Python path.

The container also includes the following:

- **Ubuntu** 16.04

  Container image 18.04-py2 contains Python 2.7; 18.04-py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA\textsuperscript{®} Basic Linear Algebra Subroutines library\textsuperscript{™} (cuBLAS) 9.0.333 (see section 2.3.1)

- NVIDIA CUDA\textsuperscript{®} Deep Neural Network library\textsuperscript{™} (cuDNN) 7.1.1

- NCCL 2.1.15 (optimized for NVLink\textsuperscript{™})

- ONNX exporter for CNN classification models

  The ONNX exporter is being continuously improved. You can try the latest changes by pulling from the master branch.

- Amazon Labs Sockeye sequence-to-sequence framework (for machine translation)

Driver Requirements

Release 18.04 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This MXNet release includes the following key features and enhancements.
MXNet container image version 18.04 is based on MXNet 1.1.0.

For this month, no upstream merges as we develop a performant approach to MXNet’s recent operator refactoring to a stateless imperative style.

ResNet-50 performance improvement based on the automatic fusion of `add_relu` and `copy_split` backward pass.

Latest version of NCCL 2.1.15

Ubuntu 16.04 with March 2018 updates

**Announcements**

Starting with the next major version of CUDA release, we will no longer provide Python 2 containers and will only maintain Python 3 containers.

**Known Issues**

There are no known issues in this release.
Chapter 6.
MXNET RELEASE 18.03

The NVIDIA container image of MXNet, release 18.03, is available.

**Contents of MXNet**

This container image contains the complete source of the version of MXNet in `/opt/mxnet`. It is pre-built and installed to the Python path.

The container also includes the following:

- **Ubuntu 16.04**
- **Container image 18.03-py2 contains Python 2.7; 18.03-py3 contains Python 3.5.**
- **NVIDIA CUDA 9.0.176** (see Errata section and 2.1) including **CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333** (see section 2.3.1)
- **NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.1**
- **NCCL 2.1.2** (optimized for NVLink™)
- **ONNX exporter for CNN classification models**
  - The ONNX exporter is being continuously improved. You can try the latest changes by pulling from the master branch.
- **Amazon Labs Sockeye sequence-to-sequence framework** (for machine translation)

**Driver Requirements**

Release 18.03 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

**Key Features and Enhancements**

This MXNet release includes the following key features and enhancements.
- MXNet container image version 18.03 is based on MXNet 1.1.0.
- Incorporated all upstream changes from the MXNet master branch, specifically, PR 9749.
- Added compute-graph optimizations for improved ResNet performance.
- Latest version of cuBLAS 9.0.333
- Latest version of cuDNN 7.1.1
- Ubuntu 16.04 with February 2018 updates

**Announcements**

Starting with the next major version of CUDA release, we will no longer provide Python 2 containers and will only maintain Python 3 containers.

**Known Issues**

There are no known issues in this release.
Chapter 7.
MXNET RELEASE 18.02

The NVIDIA container image of MXNet, release 18.02, is available.
MXNet container image version 18.02 is based on MXNet 1.0.0.

Contents of MXNet

This container image contains the complete source of the version of MXNet in /opt/mxnet. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04

![Image of a container with text](container.png)

- NVIDIA CUDA 9.0.176 including:
  - CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.282 Patch 2 which is installed by default
  - cuBLAS 9.0.234 Patch 1 as a debian file. Installing Patch 1 by issuing the `dpkg -i /opt/cuda-cublas-9-0_9.0.234-1_amd64.deb` command is the workaround for the known issue described below.
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NCCL 2.1.2 (optimized for NVLink™)
- ONNX exporter for CNN classification models

![Image of a message box](message_box.png)

- The ONNX exporter is being continuously improved. You can try the latest changes by pulling from the master branch.

- Amazon Labs Sockeye sequence-to-sequence framework (for machine translation)
Driver Requirements

Release 18.02 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This MXNet release includes the following key features and enhancements.

- Latest version of cuBLAS
- Ubuntu 16.04 with January 2018 updates

Known Issues

cuBLAS 9.0.282 regresses RNN seq2seq FP16 performance for a small subset of input sizes. This issue should be fixed in the next update. As a workaround, install cuBLAS 9.0.234 Patch 1 by issuing the `dpkg -i /opt/cuda-cublas-9-0_9.0.234-1_amd64.deb` command.
Chapter 8.
MXNET RELEASE 18.01

The NVIDIA container image of MXNet, release 18.01, is available. MXNet container image version 18.01 is based on MXNet 1.0.0.

Contents of MXNet

This container image contains the complete source of the version of MXNet in /opt/mxnet. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04

  Container image 18.01-py2 contains Python 2.7; 18.01-py3 contains Python 3.5.

- NVIDIA CUDA 9.0.176 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.282
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NCCL 2.1.2 (optimized for NVLink™)
- ONNX exporter for CNN classification models

  The ONNX exporter is being continuously improved. You can try the latest changes by pulling from the master branch.

- Amazon Labs Sockeye sequence-to-sequence framework (for machine translation)

Driver Requirements

Release 18.01 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.
Key Features and Enhancements

This MXNet release includes the following key features and enhancements.

- Addition of Python 3 package
- Enhanced-performance cuDNN-based batched 1D convolutions (merged to upstream)
- Added **MxNet-to-ONNX** exporter for classification of CNN models (tested with LeNet-5, ResNet-50, etc.).
- Added the Sockeye sequence-to-sequence framework, along with a German-to-English translation model, based on the WMT'15 dataset and translation task. This model's launch script should reproduce the OpenNMT reference model when trained until convergence.
- Latest version of cuBLAS
- Latest version of cuDNN
- Latest version of NCCL
- Ubuntu 16.04 with December 2017 updates

Known Issues

cuBLAS 9.0.282 regresses RNN seq2seq FP16 performance for a small subset of input sizes. As a workaround, revert back to the 11.12 container.
Chapter 9.  
**MXNET RELEASE 17.12**

The NVIDIA container image of MXNet, release 17.12, is available. MXNet container image version 17.12 is based on MXNet 1.0.0.

**Contents of MXNet**

This container image contains the complete source of the version of MXNet in `/opt/mxnet`. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 9.0.176 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.234
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.5
- NCCL 2.1.2 (optimized for NVLink™)

**Driver Requirements**

Release 17.12 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

**Key Features and Enhancements**

This MXNet release includes the following key features and enhancements.

- Both the `nccl` and `nccl_allreduce` KVStore options now have the same, improved performance.
- Latest version of CUDA
- Latest version of cuDNN
- Latest version of NCCL
- Ubuntu 16.04 with November 2017 updates
Known Issues

There are no known issues in this release.
Chapter 10.  
MXNET RELEASE 17.11

The NVIDIA container image of MXNet, release 17.11, is available. 
MXNet container image version 17.11 is based on MXNet 0.12.0.

Contents of MXNet
This container image contains the complete source of the version of MXNet in /opt/mxnet. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 9.0.176 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.234
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.4
- NCCL 2.1.2 (optimized for NVLink™)

Driver Requirements
Release 17.11 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements
This MXNet release includes the following key features and enhancements.

- Added Sockeye to the container, including NCCL kvstore option
- Enabled mx.sym.batch_dot to use FP16
- Latest version of CUDA
- Latest version of cuDNN
- Latest version of NCCL
- Ubuntu 16.04 with October 2017 updates
Known Issues

There are no known issues in this release.
Chapter 11.
MXNET RELEASE 17.10

The NVIDIA container image of MXNet, release 17.10, is available.
MXNet container image version 17.10 is based on MXNet 0.11.0.

Contents of MXNet
This container image contains the complete source of the version of MXNet in /opt/mxnet. It is pre-built and installed to the Python path.
The container also includes the following:
  ▸ Ubuntu 16.04
  ▸ NVIDIA CUDA® 9.0
  ▸ NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.3
  ▸ NCCL 2.0.5 (optimized for NVLink™)

Driver Requirements
Release 17.10 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements
This MXNet release includes the following key features and enhancements.
  ▸ Mixed precision support for all optimizers
  ▸ New image input pipeline with faster speed and support for global shuffling after each epoch when used with IndexedRecordIO format.
  ▸ Latest version of CUDA
  ▸ Latest version of cuDNN
  ▸ Latest version of NCCL
  ▸ Ubuntu 16.04 with September 2017 updates
Known Issues

There are no known issues in this release.
Chapter 12.
MXNET RELEASE 17.09

The NVIDIA container image of MXNet, release 17.09, is available. MXNet container image version 17.09 is based on MXNet 0.11.0.rc3.

Contents of MXNet
This container image contains the complete source of the version of MXNet in `/opt/mxnet`. It is pre-built and installed to the Python path.

The container also includes the following:
- Ubuntu 16.04
- NVIDIA CUDA® 9.0
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.0.2
- NCCL 2.0.5 (optimized for NVLink™)

Driver Requirements
Release 17.09 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements
This MXNet release includes the following key features and enhancements.
- Tensor Core support in convolutions, deconvolutions, and fully connected layers on Volta
- Support for mixed precision training with SGD optimizer
- Streamlined FP16 examples for image classification
- Optimized input pipeline for image processing
- Latest version of CUDA
- Latest version of cuDNN
- Latest version of NCCL
- Ubuntu 16.04 with August 2017 updates
Known Issues

There are no known issues in this release.
The NVIDIA container image of MXNet, release 17.07, is available. MXNet container image version 17.07 is based on MXNet 0.10.0.

Contents of MXNet

This container image contains the complete source of the version of MXNet in `/opt/mxnet`. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61.2 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) Patch 2
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.21
- NCCL 2.0.3 (optimized for NVLink™)

Key Features and Enhancements

This MXNet release includes the following key features and enhancements.

- Support for multi-precision SGD
- cuBLAS back-end for `FullyConnected` operation
- Ubuntu 16.04 with June 2017 updates

Known Issues

There are no known issues in this release.
Chapter 14.
MXNET RELEASE 17.06

The NVIDIA container image of MXNet, release 17.06, is available. MXNet container image version 17.06 is based on MXNet 0.10.0.

Contents of MXNet
This container image contains the complete source of the version of MXNet in /opt/mxnet. It is pre-built and installed to the Python path.
The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.21
- NCCL 1.6.1 (optimized for NVLink™)

Key Features and Enhancements
This MXNet release includes the following key features and enhancements.

- Implemented double buffering in ResNet v1 example
- Ubuntu 16.04 with May 2017 updates

Known Issues
There are no known issues in this release.
The NVIDIA container image of MXNet, release 17.05, is available. MXNet container image version 17.05 is based on MXNet 0.9.3a+.

Contents of MXNet
This container image contains the complete source of the version of MXNet in `/opt/mxnet`. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.21
- NCCL 1.6.1 (optimized for NVLink™)

Key Features and Enhancements
This MXNet release includes the following key features and enhancements.

- Latest cuDNN release
- Improved IO pipeline for increased multi-GPU performance
- Optimized SGD weight update
- Added the `nccl_allreduce` option for gradient communication
- Added support for dilated deconvolutions
- Improved convolutional neural network (CNN) performance by removing unnecessary computations
- Added options to show the cuDNN algorithms that are chosen for convolutions
- Ubuntu 16.04 with April 2017 updates

Known Issues
There are no known issues in this release.
The NVIDIA container image of MXNet, release 17.04, is available.

MXNet container image version 17.04 is based on MXNet 0.9.3a+.

Contents of MXNet

This container image contains the complete source of the version of MXNet in `/opt/mxnet`. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.20
- NCCL 1.6.1 (optimized for NVLink™)

Key Features and Enhancements

This MXNet release includes the following key features and enhancements.

- Support for cuDNN accelerated dilated convolutions
- Ubuntu 16.04 with March 2017 updates

Known Issues

There are no known issues in this release.
Chapter 17.
MXNET RELEASE 17.03

The NVIDIA container image of MXNet, release 17.03, is available. MXNet container image version 17.03 is based on MXNet 0.9.3.

Contents of MXNet

This container image contains the complete source of the version of MXNet in /opt/mxnet. It is pre-built and installed to the Python path.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA® 8.0.61
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 6.0.20

Key Features and Enhancements

This MXNet release includes the following key features and enhancements.

- Ubuntu 16.04 with February 2017 updates
- Improved input pipeline for image processing
- Support for FP16 training of AlexNet
- Optimized embedding layer of CUDA kernels
- Optimized tensor broadcast and reduce CUDA kernels

Known Issues

There are no known issues in this release.
Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do 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 guide. NVIDIA does not accept any 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 guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, cuDNN, cuFFT, cuSPARSE, DIGITS, DGX, DGX-1, DGX Station, GRID, Jetson, Kepler, NVIDIA GPU Cloud, Maxwell, NCCL, NVLink, Pascal, Tegra, TensorRT, Tesla and Volta are trademarks and/or registered trademarks of NVIDIA Corporation in the Unites States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2018 NVIDIA Corporation. All rights reserved.

www.nvidia.com