



Deploying NGC Containers on Your Cloud Virtual Machine

Users Guide

Table of Contents

Chapter 1. Preparing to Run Containers.....	1
Chapter 2. Running a Container.....	2
Chapter 3. Example: MNIST Training Run Using PyTorch Container.....	3
Chapter 4. Example: MNIST Training Run Using TensorFlow Container.....	4

Chapter 1. Preparing to Run Containers

The VMI includes a mechanism for supporting GPUs within Docker containers to obtain the best performance. Depending on the NVIDIA VMI version, the mechanisms are as follows.

- ▶ Native GPU support with Docker-CE
 - Requires Docker-CE 19.03 or later (Included in NVIDIA VMIs 19.10 and later)
- ▶ NVIDIA Container Runtime with Docker-CE
 - Included in NVIDIA VMIs prior to 19.10

Using Native GPU Support with Docker-CE

Use this method with NVIDIA VMIs version 19.10 and later.

Use `docker run --gpus` to run GPU-enabled containers.

- ▶ Example using all GPUs

```
$ docker run --gpus all ...
```

- ▶ Example using two GPUs

```
$ docker run --gpus 2 ...
```

- ▶ Examples using specific GPUs

```
$ docker run --gpus "device=1,2" ... $ docker run --gpus "device=UUID-  
ABCDEF,1" ...
```

Using the NVIDIA Container Runtime with Docker-CE

Use this method with NVIDIA VMIs prior to version 19.10

Use `docker run` and specify `runtime=nvidia`.

```
$ docker run --runtime=nvidia ...
```

Chapter 2. Running a Container

This section explains the basic process for running a container on the various flavors of the NVIDIA GPU-Optimized Images used to launch virtual machines on the cloud provider of your choice.

Running the Built-in TensorFlow Container

To run the TensorFlow container in the VM created from the NVIDIA GPU-Optimized Image for TensorFlow, refer to the release notes for the correct **tag** to use, then enter the following command.

On NVIDIA VMs version 19.10 and later

```
docker run --gpus all --rm -it nvcr.io/nvidia/tensorflow:<tag>
```

On NVIDIA VMs prior to version 19.10

```
docker run --runtime=nvidia --rm -it nvcr.io/nvidia/tensorflow:<tag>
```

Running the Built-in PyTorch Container

To run the PyTorch container in the VM created from the NVIDIA GPU-Optimized Image for PyTorch, refer to the release notes for the correct **tag** to use, then enter the following command.

On NVIDIA VMs version 19.10 and later

```
docker run --gpus all --rm -it nvcr.io/nvidia/pytorch:<tag>
```

On NVIDIA VMs prior to version 19.10

```
docker run --runtime=nvidia --rm -it nvcr.io/nvidia/pytorch:<tag>
```

Running a Container Downloaded from NGC

To run containers from the NGC container registry, enter the following commands.

On NVIDIA VMs version 19.10 and later

```
docker run --gpus all --rm -it nvcr.io/nvidia/<container-image>:<tag>
```

On NVIDIA VMs prior to version 19.10

```
docker run --runtime=nvidia --rm -it nvcr.io/nvidia/<container-image>:<tag>
```

Chapter 3. Example: MNIST Training Run Using PyTorch Container

Once logged in to the NVIDIA GPU Cloud Image instance, you can run the MNIST example under PyTorch.

Note that the PyTorch example will download the MNIST dataset from the web.

1. Pull and run the PyTorch container:

```
docker pull nvcr.io/nvidia/pytorch:18.02-py3
```

On NVIDIA VMIs version 19.10 and later

```
docker run --gpus all --rm -it nvcr.io/nvidia/pytorch:18.02-py3.10
```

On NVIDIA VMIs prior to version 19.10

```
docker run --runtime=nvidia --rm -it nvcr.io/nvidia/pytorch:18.02-py3.10
```

2. Run the MNIST example:

```
cd /opt/pytorch/examples/mnist
```

```
python main.py
```

Chapter 4. Example: MNIST Training Run Using TensorFlow Container

Once logged in to the NVIDIA GPU Cloud image, you can run the MNIST example under TensorFlow.

Note that the TensorFlow built-in example will pull the MNIST dataset from the web.

1. Pull and run the TensorFlow container.

```
docker pull nvcr.io/nvidia/tensorflow:18.08-py3
```

On NVIDIA VMIs version 19.10 and later

```
docker run --gpus all --rm -it nvcr.io/nvidia/tensorflow:18.08-py3
```

On NVIDIA VMIs prior to version 19.10

```
docker run --runtime=nvidia --rm -it nvcr.io/nvidia/tensorflow:18.08-py3
```

2. Following this tutorial: https://www.tensorflow.org/get_started/mnist/beginners, run the `MNIST_with_summaries` example.

```
cd /opt/tensorflow/tensorflow/examples/tutorials/mnist
```

```
python mnist_with_summaries.py
```

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 and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2022 NVIDIA Corporation. All rights reserved.