USING NGC WITH NVIDIA VIRTUAL GPU SOFTWARE

DU-09254-001_v04 | September 2019

Setup Guide
TABLE OF CONTENTS

Chapter 1. Introduction ............................................................................................................ 1
Chapter 2. Installing Docker and the NVIDIA Container Runtime for Docker ...................... 3
  2.1. Installing the Docker Repository .................................................................................. 3
  2.2. Installing the NVIDIA Container Runtime for Docker ............................................... 3
  2.3. Enabling GPU Support for NGC Containers ................................................................. 4
Chapter 3. Using NGC Containers ......................................................................................... 7
Chapter 1.
INTRODUCTION

NVIDIA® GPU Cloud (NGC) containers leverage the power of GPUs based on the NVIDIA Pascal™, Volta™, and Turing architectures. NGC containers can run in virtual machines (VMs) configured with NVIDIA virtual GPU (vGPU) software in NVIDIA vGPU and GPU pass-through deployments.

The document describes how to set up a VM configured with NVIDIA virtual GPU software to run NGC containers. Open the command line in the VM and paste the code blocks into the command line.

Prerequisites

These instructions assume that the following prerequisites are met:

- A guest VM running a supported Linux release is configured with an NVIDIA vGPU or a pass-through GPU.
- The NVIDIA virtual GPU software graphics driver is installed in the guest VM.

Ensure that the driver that is installed is the graphics driver bundled with the NVIDIA virtual GPU software.

- Any NVIDIA virtual GPU software products that you are using have been licensed with NVIDIA Quadro® Virtual Data Center Workstation (Quadro vDWS).

For instructions, visit https://docs/nvidia.com/grid.

Supported Software and Hardware

NVIDIA virtual GPU software release 7.0 to 9.0 is supported.

In NVIDIA vGPU deployments, the following vGPU types are supported only on NVIDIA GPU architectures after the Maxwell architecture:

- All Q-series vGPU types
- All C-series vGPU types
In GPU pass-through deployments, all GPUs based on NVIDIA GPU architectures after the NVIDIA Maxwell™ architecture that support NVIDIA vGPU software are supported.

Supported guest operating system: Ubuntu 16.04 LTS, Ubuntu 18.04 LTS

The following hypervisor software releases are supported.

- Citrix Hypervisor 7.0, 7.1, 7.5, and 7.6
- Microsoft Windows Server with Hyper-V role 2016 1709 and 2016 1607
- Nutanix AHV 5.5, 5.6, and 5.11
- Red Hat Enterprise Linux with Kernel Virtual Machine (KVM) 7.0-7.6
- Red Hat Virtualization (RHV) 4.3
- VMware vSphere 6.7, 6.5, and 6.0
Chapter 2.
INSTALLING DOCKER AND THE NVIDIA CONTAINER RUNTIME FOR DOCKER

The Docker runtime is required to run NGC containers. In addition, the NVIDIA Container Runtime for Docker (nvidia-docker2) ensures that the high performance power of the GPU is leveraged when running NVIDIA-optimized Docker containers.

2.1. Installing the Docker Repository

The following code block

1. Installs apt-transport-https.
2. Installs curl.
3. Installs the Docker prerequisites.
4. Adds the Docker official GPG key.
5. Adds the official stable Docker repository.
   Refer to https://docs.docker.com/engine/installation/linux/docker-ce/ubuntu/#install-docker-ce for more information.

```
sudo apt-get install -y apt-transport-https
curl ca-certificates
software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

2.2. Installing the NVIDIA Container Runtime for Docker
1. Issue the following commands to install the NVIDIA Container Runtime for Docker (nvidia-docker2) repository, install nvidia-docker2, and then set up permissions to use Docker without sudo each time (where $USER refers to the user name).

```bash
curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | \
sudo apt-key add -
curl -s -L https://nvidia.github.io/nvidia-docker/ubuntu16.04/amd64/nvidia-docker.list | \
sudo tee /etc/apt/sources.list.d/nvidia-docker.list
sudo apt update
sudo apt install -y nvidia-docker2
sudo usermod -aG docker $USER
```

For more information, see [https://github.com/NVIDIA/nvidia-docker](https://github.com/NVIDIA/nvidia-docker).

2. Reboot the system.

```bash
sudo reboot
```

3. Upon reboot, test nvidia-smi with the latest official CUDA image.

```bash
docker run --runtime=nvidia --rm nvcr.io/nvidia/cuda:latest nvidia-smi
```

Docker pulls the nvidia/cuda container image layer by layer, then runs nvidia-smi.

When completed, the output should show the NVIDIA Driver version and a description of each installed GPU.

### 2.3. Enabling GPU Support for NGC Containers

To obtain the best performance when running NGC containers, three methods of providing GPU support for Docker containers have been developed:

- Native GPU support (preferred method if running Docker-ce 19.03 or later)
- `nvidia-docker2`
- `nvidia-docker`

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.

Refer to the following table to assist in determining which method is implemented in your system.

<table>
<thead>
<tr>
<th>GPU Support Method</th>
<th>When Used</th>
<th>How to Determine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native GPU Support</td>
<td>Included with Docker-ce 19.03 or later</td>
<td>Run <code>docker version</code> to determine the installed Docker version.</td>
</tr>
<tr>
<td><code>nvidia-docker2</code></td>
<td>If installed</td>
<td>Run <code>nvidia-docker version</code> and check for NVIDIA Docker version 2.0 or later</td>
</tr>
<tr>
<td><code>nvidia-docker</code></td>
<td>If installed</td>
<td>Run <code>nvidia-docker version</code> and check for NVIDIA Docker version 1.x</td>
</tr>
</tbody>
</table>
Each method is invoked by using specific Docker commands, described as follows.

**Using Native GPU support**

![Message Box]

If Docker is updated to 19.03 on a system which already has nvidia-docker or nvidia-docker2 installed, then the corresponding methods can still be used.

- To use the native support on a new installation of Docker, first enable the new GPU support in Docker.

  ```
  $ sudo apt-get install -y docker nvidia-container-toolkit
  ```

  This step is not needed if you have updated Docker to 19.03 on a system with nvidia-docker2 installed. The native support will be enabled automatically.

- 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 nvidia-docker2**

With the NVIDIA Container Runtime for Docker installed (nvidia-docker2), you can run GPU-accelerated containers in one of the following ways.

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

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

- Use `nvidia-docker run`.

  ```
  $ nvidia-docker run ...
  ```

  The new package provides backward compatibility, so you can still run GPU-accelerated containers by using this command, and the new runtime will be used.

- Use `docker run` with `nvidia` as the default runtime.

  You can set `nvidia` as the default runtime, for example, by adding the following line to the `/etc/docker/daemon.json` configuration file as the first entry.

  ```json
  "default-runtime": "nvidia",
  "runtimes": {
    "nvidia": {
      "path": "/usr/bin/nvidia-container-runtime",
      "runtimeArgs": []
    }
  }
  ```

  The following is an example of how the added line appears in the JSON file. Do not remove any pre-existing content when making this change.
You can then use `docker run` to run GPU-accelerated containers.

```
$ docker run ...
```

**Caution** If you build Docker images while `nvidia` is set as the default runtime, make sure the build scripts executed by the Dockerfile specify the GPU architectures that the container will need. Failure to do so may result in the container being optimized only for the GPU architecture on which it was built. Instructions for specifying the GPU architecture depend on the application and are beyond the scope of this document. Consult the specific application build process for guidance.

**Using nvidia-docker**

With the Docker Engine Utility for NVIDIA GPUs installed (`nvidia-docker`), run GPU-enabled containers as follows.

```
$ nvidia-docker run ...
```
Chapter 3.
USING NGC CONTAINERS

Make sure you have performed the following steps from the NGC website (see the NGC Getting Started Guide)

‣ Signed up for an NGC account at https://ngc.nvidia.com/signup.
‣ Created an NGC API key for access to the NGC container registry.
‣ Browsed the NGC website and identified an available NGC container and tag to run.

See the following documents for detailed instructions on using NGC Containers.

‣ NGC Container User Guide
‣ Using Deep Learning Containers
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 Volta are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries.

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries.

Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2019 NVIDIA Corporation. All rights reserved.