# TABLE OF CONTENTS

Chapter 1. Introduction to Using NGC with Google Cloud Platform ........................................... 1

Chapter 2. Before You Start ........................................................................................................ 2
  2.1. Prerequisites .................................................................................................................. 2
  2.2. About Shared Projects ................................................................................................. 2
  2.3. Setting Up SSH Keys .................................................................................................... 3
  2.4. Setting Firewall Rules .................................................................................................. 3

Chapter 3. Deploying an NVIDIA GPU Cloud Image from the GCP Console ............................... 6
  3.1. Creating Your GPU Instance ....................................................................................... 6
  3.2. Connecting to Your GPU Instance ............................................................................. 10
  3.3. Stopping and Restarting Your VM Instance ................................................................. 11

Chapter 4. Launching an NVIDIA GPU Cloud Image with the gcloud CLI ................................. 12
  4.1. Installing and Setting Up gcloud CLI .......................................................................... 12
  4.2. Preparing the Create Instance Options ..................................................................... 13
  4.3. Creating a Custom GPU Instance ............................................................................. 15
  4.4. Connecting to Your GPU Instance with SSH ............................................................ 15
  4.5. Stopping and Restarting Your GPU Instance ............................................................ 16
  4.6. More Advanced CLI Usage ....................................................................................... 16

Chapter 5. Creating Persistent SSD Disks for Datasets .............................................................. 17
  5.1. Creating a Persistent SSD Dataset Disk and Attaching it to the VM Instance from the GCP Console .............................................................................................................. 17
  5.2. Creating a Persistent SSD Dataset Disk and Attaching it to the VM Instance from the gcloud CLI .................................................................................................................. 19
  5.3. Creating a File System and Mounting the Volume ...................................................... 19
  5.4. Deleting a Persistent SSD Disk .................................................................................. 19

Chapter 6. Examples of Running Containers ............................................................................. 21
  6.1. Logging Into the NGC Container Registry ................................................................. 21
  6.2. Preparing to Run Containers ...................................................................................... 21
  6.3. Running a Container ................................................................................................... 22
  6.4. Example: MNIST Training Run Using PyTorch Container ......................................... 23
  6.5. Example: MNIST Training Run Using TensorFlow Container .................................. 24
NVIDIA makes available on the Google Cloud Platform (GCP) a customized image optimized for the NVIDIA Pascal™ and Volta™ GPUs. Running NGC containers on this virtual machine (VM) instance provides optimum performance for deep learning jobs.

Three flavors of the NVIDIA GPU Cloud image are available:

- Standard NVIDIA GPU Cloud Image
  - Includes Ubuntu Server, the NVIDIA driver, Docker CE, and the NVIDIA Container Runtime for Docker
- GPU Accelerated Image for TensorFlow
  - The standard image plus a built-in, ready-to-use TensorFlow container
- GPU Accelerated Image for PyTorch
  - The standard image plus a built-in, ready-to-use PyTorch container

For those familiar with the Google Cloud Platform, the process of launching the instance is as simple as logging into GCP and creating a deployment solution using the Google Cloud Launcher. After deploying the NVIDIA GPU Cloud Image solution, you can SSH into the GPU Cloud instance and start running deep learning jobs using framework containers from the NGC container registry.

This document provides step-by-step instructions for accomplishing this.
Chapter 2. 
BEFORE YOU START

Be sure you are familiar with the information in this chapter before starting to use the NVIDIA GPU Cloud Image on the Google Cloud Platform (GCP).

2.1. Prerequisites

These instructions assume the following:

‣ You have a Google Cloud account - https://console.cloud.google.com/.
‣ You have installed the gcloud SDK if you plan to use the CLI. See setup instructions below.
‣ You have SSH keys; see setup instructions below.
‣ Windows Users: The CLI code snippets are for bash on Linux or Mac OS X. If you are using Windows and want to use the snippets as-is, you can use the Windows Subsystem for Linux and use the bash shell (you will be in Ubuntu Linux).

Additionally, if you plan to access locked NGC containers, you will need to perform the following steps from the NGC website (see NGC Getting Started Guide)

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

2.2. About Shared Projects

Shared projects allow multiple users to access any virtual machine instance created within the project. This means other users within the project could establish an SSH connection to your instance. To keep your VM instance and SSH key private, create a private project and then create and launch your VM instances from within your private project.
2.3. Setting Up SSH Keys

The Google Compute Engine generates and manages an SSH key automatically for logging into your instance (see the Google Cloud documentation Connecting to Instances.). However, to facilitate logging into the NGC container registry upon the initial connection to the VM instance, you need to

1. Generate your own SSH keys (see Creating a new SSH key for instructions), and then
2. Add them to the metadata for your project (see Adding or Removing Project-Wide Public SSH Keys for instructions).

If you do not prepare your SSH keys before launching and connecting to your VM instance, you will not be able to access the NGC container registry initially. In that case you will need to

1. Add yourself to the docker group after connecting to the instance.

   ```bash
   sudo usermod -aG docker $USER
   ```
2. Restart the session.

2.4. Setting Firewall Rules

NVIDIA recommends setting firewall rules to allow external access to ports 443 (HTTPS), 8888 (DIGITS), and any other ports that may be needed. This should be done before launching an instance to avoid having to stop the instance when setting any firewall rules later.

You can specify that HTTPS traffic be allowed using the VM Instance Details page, but changing that setting also requires that the instance be stopped.

1. Log in to https://console.cloud.google.com
2. Verify you are in the correct Project.
3. Click the Products and Services menu icon, then scroll down to the Networking section and click VPC Network->Firewall Rules.
4. Click **Create Firewall Rule**.

5. Enter the following information to specify the firewall rule you want to create.

   ▶ **Name**: NVIDIA recommends the following naming format
     
     For HTTPS: “default-allow-https”
     
     For DIGITS: “default-allow-digits”
     
     You can also create rules for other DIGITS versions, such as DIGITS4

   ▶ **Direction of traffic**: "Ingress"

   ▶ **Action on match**: "Allow"

   ▶ **Targets**: "All instances in the network"

   ▶ **Source filter**: "IP ranges"

   ▶ **Source IP ranges**: "0.0.0.0/0"

   ▶ **Protocols and ports**: "Specified protocols and ports", then enter
     
     For HTTPS: “tcp:443”
     
     For DIGITS: “tcp:8888”
     
     You can enter ports for other DIGITS versions as well

6. Click **Create**.

   Your new firewall rules should appear on the Firewall Rules page.
Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. Learn more.

Note: App Engine firewalls are managed here.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Targets</th>
<th>Filters</th>
<th>Protocols / ports</th>
<th>Action</th>
<th>Priority</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>default-allowed-digits</td>
<td>Ingress</td>
<td>Apply to all</td>
<td>IP ranges: 0.0.0.0/0</td>
<td>tcp:8888</td>
<td>Allow</td>
<td>1000</td>
<td>default</td>
</tr>
<tr>
<td>default-allowed-https</td>
<td>Ingress</td>
<td>Apply to all</td>
<td>IP ranges: 0.0.0.0/0</td>
<td>tcp:443</td>
<td>Allow</td>
<td>1000</td>
<td>default</td>
</tr>
</tbody>
</table>
3.1. Creating Your GPU Instance

2. Verify you are in the correct project.
   Click the Products and Services menu icon and select **Marketplace**.

5. Click either the standard NVIDIA GPU Cloud Image, the GPU Accelerated Image for TensorFlow, or the GPU Accelerated Image for PyTorch.
6. From the NVIDIA GPU Cloud Image information page, click **Launch on Compute Engine**.

7. Configure the NVIDIA GPU Cloud Image deployment.
   a) In "**Name**", enter your new deployment name.
b) In "Zone", select the zone to create the instance (select one that features the appropriate GPU).

c) Under the **GPU** section, select the GPU type and Number of GPUs.

If it appears, do not select the **k80 GPU** as it is not supported.

**GPU**

*GPU type*

- **nvidia-tesla-p100**

**Number of GPUs**

Instances with GPUs have specific restrictions that make them behave differently than other instance types. GPUs are only available in certain zones, and the number of GPU dies is linked to the number of CPU cores selected for this instance. Learn more

- 1

---

d) In the "**Machine Type**" section, click **Customize** to open the customize view.

**Machine type**

- 16 vCPUs
- 104 GB memory

---

e) Assign the Cores (vCPUs) and Memory.

The following ratio is recommended: 1x GPU : 10x vCPU: 60 GB mem)

f) In the "**Boot disk**" section, select Standard Persistent Disk.
g) Make other changes as needed for Networking, Firewall and IP.

8. Click **Deploy** from the bottom of the page.
   It may take a few minutes for the deployment process to complete.

Wait for the message that your solution has been deployed, then you can connect to your running instance.

The right side of the deployment page provides information about your deployed image.
Make a note of the name for your instance. You can stop and restart this instance, as well as attach additional storage to it for datasets.
3.2. Connecting to Your GPU Instance

Connect to your instance from the Deployment page or from the VM Instance details page.

1. Connect to your instance.
   - If you are still on the Deployment page, you can click **SSH** to connect to your instance.

   ![nvidia-nc-g-image]

<table>
<thead>
<tr>
<th>Instance</th>
<th>nvidia-ngc-image-3-vm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance zone</td>
<td>us-west1-b</td>
</tr>
<tr>
<td>Instance machine type</td>
<td>n1-highmem-8</td>
</tr>
</tbody>
</table>

   ▼ More about the software

   Get started with NVIDIA GPU Cloud Image
   
   ![SSH]

   ▼ If you are no longer on the Deployment page, you can return to your instance and connect as follows.
   1. Click the Products and Services menu icon, then scroll down to the Compute Engine section and click **VM Instances**.
   2. Either click **SSH** by your listed deployed instance, or click your deployed instance and then click **SSH** from the VM instance details page.

The latest NVIDIA drivers must be installed on the NVIDIA GPU Cloud Image instance before running. If the drivers have not yet been installed on this instance, then upon connecting, the instance startup script asks if you want to download and install the latest NVIDIA drivers.

Would you like to download the latest NVIDIA drivers so NGC can finish installing? (Y/n)

2. Press **Y** to install the latest NVIDIA drivers and proceed with the connection.
If you press N, then the connection process will abort and the instance will be stopped.

After you connect, you can use the terminal to run commands on your Linux instance. When you are done, use the `exit` command to disconnect from the instance.

### 3.3. Stopping and Restarting Your VM Instance

Click your GPU instance, either from the Deployment Manager->your deployment page or from the Compute Engine->VM Instances page.

The top menu lets you edit, stop a running instance, or start a stopped instance.

**Menu for a running instance**

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM instance details</td>
<td></td>
</tr>
<tr>
<td>EDIT</td>
<td></td>
</tr>
<tr>
<td>RESET</td>
<td></td>
</tr>
<tr>
<td>CLONE</td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td></td>
</tr>
</tbody>
</table>

**Menu for a stopped instance**

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM instance details</td>
<td></td>
</tr>
<tr>
<td>EDIT</td>
<td></td>
</tr>
<tr>
<td>RESET</td>
<td></td>
</tr>
<tr>
<td>CLONE</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td></td>
</tr>
</tbody>
</table>

**Caution** Do not click Delete to delete your instance. Doing so will make your deployment unusable as you will not be able to recreate the instance for the deployment. Instead, delete the deployment using the Deployment Manager.
This section explains how to create a GPU Cloud instance using the gcloud CLI.

**Using Example Python Scripts**

A comprehensive set of example Python scripts for automating the CLI are provided at [https://github.com/nvidia/ngc-examples/tree/master/ncsp](https://github.com/nvidia/ngc-examples/tree/master/ncsp). You can download the scripts and modify them to meet your requirements. The code examples that follow use similar environment variables and structure as the scripts.

**Using the Instructions in this Chapter**

This flow and the code snippets in this section are for Linux or Mac OS X. If you are using Windows, you can use the Windows Subsystem for Linux and use the bash shell (where you will be in Ubuntu Linux). Many of these CLI commands can have significant delays.

For more information about creating a deployment using gcloud CLI, see *Creating a Deployment using gcloud or the API*.

**4.1. Installing and Setting Up gcloud CLI**

Follow the instructions at [https://cloud.google.com/sdk/docs/quickstarts](https://cloud.google.com/sdk/docs/quickstarts). These include instructions for Linux, Mac, and Windows.

The instructions walk you through the platform specific install and initial gcloud login.

For at least the Mac, you will be given a large list of additional gcloud components to install such as extensions for GO, Python and Java. You can use the defaults for now, and use the `gcloud components` command later to list, install, or remove them.

Once the setup is complete, start a new shell since your environment has been updated.
4.2. Preparing the Create Instance Options

You will need to specify the following options when creating the custom GPU instance.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>VALUE</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;instance-name&gt;</td>
<td>Name of your choosing. Ex. “my-ngc-instance”</td>
<td>Must be all lowercase, with no spaces. Hyphens and numbers are allowed.</td>
</tr>
<tr>
<td>--project</td>
<td>&quot;&lt;my-project-id&gt;&quot;</td>
<td>This is the project in which the VM will be created. Use gcloud projects list to view PROJECT ID to use for this field.</td>
</tr>
<tr>
<td>--zone</td>
<td>One of the following zones that contain GPUs:</td>
<td>Pick one nearest you and with the GPUs you want to use.</td>
</tr>
<tr>
<td></td>
<td>&quot;us-west1-b&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;us-east1-c&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;us-east1-d&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;europe-west1-b&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;europe-west1-d&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;asia-east1-a&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;asia-east1-b&quot;</td>
<td></td>
</tr>
<tr>
<td>--machine-type</td>
<td>One of the following:</td>
<td>vCPU/Memory configuration of the VM in &quot;custom-&lt;#vCPUs&gt;-&lt;memory MB&gt;&quot; format. Recommended ratio is 1 GPU : 10 vCPUs : 60 GB memory</td>
</tr>
<tr>
<td></td>
<td>&quot;custom-10-61440&quot; (for 1x P100 or V100)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;custom-20-122880&quot; (for 2x P100)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;custom-40-212992&quot; (for 4x P100)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;custom-80-491520&quot; (for 8x V100)</td>
<td></td>
</tr>
<tr>
<td>--subnet</td>
<td>&quot;default&quot;, or the name of the VPC network to use</td>
<td></td>
</tr>
</tbody>
</table>
Launching an NVIDIA GPU Cloud Image with the gcloud CLI

- **--metadata**: "ssh-keys=<user-id>:ssh-rsa <ssh-key> <user-email>"
  
  What to do with your instance when Google performs maintenance on the host.

- **--maintenance-policy**: "TERMINATE"
  
  Compute Engine identity attached to the instance.

- **--service-account**: Compute Engine identity attached to the instance.

  Use `gcloud iam service-accounts list` to view the email for your account.

- **--scope**: "https://www.googleapis.com/auth/devstorage.read_only",
  "https://www.googleapis.com/auth/logging.write",
  "https://www.googleapis.com/auth/monitoring.write",
  "https://www.googleapis.com/auth/servicecontrol",
  "https://www.googleapis.com/auth/service.management.readonly",
  "https://www.googleapis.com/auth/trace.append"

  Default values (recommended). Specifies the permissions for your instance.

- **--accelerator**: nvidia-tesla-p100,count=[1,2,4]
  
  Which GPU to attach, and how many.

- **--min-cpu-platform**: "Intel Broadwell"
  
  (for P100 instances)

- **--image**: Name of the latest NVIDIA GPU Cloud Image (See the NGC GCP VMI Release Notes for the current name.)

- **--image-project**: "nvidia-ngc-public"
  
  Project name in which the NVIDIA GPU Cloud Image is located

- **--boot-disk-size**: 32


---

www.nvidia.com
Using NGC with Google Cloud Platform

DU-08962-001_v06 | 14
4.3. Creating a Custom GPU Instance

Use the Python scripts provided at https://github.com/nvidia/ngc-examples/tree/master/ncsp to create your custom GPU instance. You can also enter the following, using the information gathered in the previous section:

```
gcloud compute  
   --project "<project-id>"  
   instances create "<instance-name>"  
   --zone "<zone>"  
   --machine-type "<vCPU-mem-config>"  
   --subnet "<subnet-name>"  
   --metadata "<your-public-ssh-key>"  
   --maintenance-policy "<maintenance-policy>"  
   --service-account "<service-account-email>"  
   --accelerator type=<accelerator-type>  
   --min-cpu-platform "<CPU-platform>"  
   --image "<nvidia-gpu-cloud-image>"  
   --image-project "<project-name>"  
   --boot-disk-size "32"  
   --boot-disk-type "pd-standard"  
   --boot-disk-device-name "<boot-disk-name>"
```

The GPU Cloud instance starts running as soon as it is created.

4.4. Connecting to Your GPU Instance with SSH

If you ran the scripts from https://github.com/nvidia/ngc-examples/tree/master/ncsp you should be connected to your instance. Otherwise, run `ssh` to connect to your GPU instance, or enter the following gcloud command.

Command syntax:

```
gcloud compute --project "<project-id>" ssh --zone "<zone>" "<instance-name>"
```

See https://cloud.google.com/compute/docs/instances/connecting-to-instance for more information about connecting to your GPU instance.

The latest NVIDIA drivers must be installed on the NVIDIA GPU Cloud Image instance before running. If the drivers have not yet been installed on this instance, then upon connecting, the instance startup script asks if you want to download and install the latest NVIDIA drivers.
NVIDIA GPU Cloud (NGC) is an optimized software environment that requires the latest NVIDIA drivers to operate. If you do not download the NVIDIA drivers at this time, your instance will shut down. Would you like to download the latest NVIDIA drivers so NGC can finish installing? (Y/n)

Press Y to install the latest NVIDIA drivers and proceed with the connection.

If you press N, then the connection process will abort and the instance will be stopped.

4.5. Stopping and Restarting Your GPU Instance

Once an instance is running, you can stop and (re)start your instance.

Stop:

gcloud compute instances stop <instance-name>

Start or Restart:

gcloud compute instances start <instance-name> <zone>

4.6. More Advanced CLI Usage

For more CLI documentation, visit the gcloud Compute Documentation.
Chapter 5.
CREATING PERSISTENT SSD DISKS FOR DATASETS

GCP recommends using Persistent SSD Disks for Compute Engine storage. A minimum of 1 TB of storage is recommended for storing deep learning datasets. However, a much larger disk or a software RAID, using mdadm, can be used to create a volume with multiple SSD Persistent Disks for achieving the the maximum performance supported by GCP on a Compute Engine instance. See instructions on how to set up software RAID on local disks. Persistent SSD disks can also be set up for software RAID using the same instructions.

5.1. Creating a Persistent SSD Dataset Disk and Attaching it to the VM Instance from the GCP Console

You can create a persistent SSD dataset disk from the GCP console as follows.

1. Log on to the Google Cloud Platform.
2. Create the SSD disk.
   a) Click Compute Engine-> Disks in the left-side navigation pane.
   b) Click Create Disk from the top of the page.
   c) Specify the following:
      ▶ Zone: Select the same zone as the VM instance you created.
      ▶ Disk Type: SSD persistent disk
If you choose to provide your own encryption key, you must provide a key that is a 256-bit string encoded in RFC 4648 standard base64 to Compute Engine. See Customer-Supplied-Encryption-Keys for details on how to provide a custom Encryption Key globally for all your operations.

d) Click Create when done.

3. Attach the disk to the VM instance.
   a) Go to the Compute Engine->VM Instance page.
   b) Click your VM instance from the list.
   c) Click Stop.
You must stop a running VM Instance as changes cannot be performed when the instance is running.
d) Click Edit.
e) Scroll down to the Additional Disks and click + Add Item.
f) Under Name, select the disk that you created and want to attach to the VM instance.
g) Click Save.
h) Start the VM instance.

Refer to the section Creating a Filesystem and Mounting the Volume for instructions on formatting and mounting the drive for uploading datasets.

5.2. Creating a Persistent SSD Dataset Disk and Attaching it to the VM Instance from the gcloud CLI

1. Create the disk using the following command.

   $ gcloud compute disks create ngc-ssd --zone <zone> --description "<your-description>" --type=pd-ssd --size=1000GM

2. Attach the disk to a VM instance using the following command.

   $ gcloud compute instances attach-disk <instance-name> --disk ngc-ssd --zone <zone>

5.3. Creating a File System and Mounting the Volume

Refer to the Formatting and mounting a persistent disk section of the Google Cloud how-to guide Adding or Resizing Persistent Disks for instructions on creating a file system and mounting the volume.

5.4. Deleting a Persistent SSD Disk

Using gcloud CLI

   $ gcloud compute instances detach-disk <instance-name> --disk ngc-ssd --zone <zone>

Using the GCP Console

Be aware that once you delete a Persistent SSD Disk, you cannot undelete it.
1. Click the disk to delete from the **Compute Engine->Disks** page.
2. On the top of the page, click Delete.
3. Click Delete at the Delete a disk confirmation dialog.
Chapter 6.
EXAMPLES OF RUNNING CONTAINERS

This chapter walks you through the process of logging in to the NGC container registry, pulling and running a container, and using file storage and data disks for storage.

6.1. Logging Into the NGC Container Registry

You need to log in to the NGC container registry only if you want to access locked containers from the registry. Most of the NGC containers are freely available (unlocked) and do not require an NGC account or NGC API key.

You do not need to log into the NGC container registry if you are using either the GPU Accelerated Image with PyTorch or the GPU Accelerated Image with TensorFlow and intend to use the containers already built into the image.

If necessary, log in to the NGC container registry manually by running the following script from the VMI.

```
ngc-login.sh <your-NGC-API-key>
```

From this point you can run Docker commands and access locked NGC containers from the VM instance.

6.2. 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)
Examples of Running Containers

- 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 ...
  ```

### 6.3. Running a Container

This section explains the basic process for running a container on the GPU Accelerated Image for TensorFlow, the GPU Accelerated Image for PyTorch, and the basic GPU Accelerated Image.

**Running the Built-in TensorFlow Container**

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

- On NVIDIA VMIs version 19.10 and later
  
  ```
  docker run --gpus all --rm -it nvcr.io/nvidia/tensorflow:<tag>
  ```

- On NVIDIA VMIs 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 GPU Accelerated Image, refer to the release notes for the correct tag to use, then enter the following command.
On NVIDIA VMIs version 19.10 and later

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

On NVIDIA VMIs prior to version 19.10

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

**Running a Container from the NGC Container Registry**

To run containers from the NGC container registry,

1. If necessary, log in to the NGC container registry as explained in the previous section.
2. Enter the following commands.

   docker pull nvcr.io/nvidia/<container-image>:<tag>

On NVIDIA VMIs version 19.10 and later

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

On NVIDIA VMIs prior to version 19.10

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

### 6.4. 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
6.5. 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