NGC on Azure Virtual Machines

Release Notes and Setup Guide
# Table of Contents

## Chapter 1. Using NGC on Azure Virtual Machines

1.1. Security Best Practices ................................................................. 1
1.2. Prerequisites ................................................................................. 1
1.3. Before You Start .......................................................................... 2
   1.3.1. Setting Up SSH Keys ............................................................. 2
   1.3.2. Setting Up a Security Group .................................................. 2
1.4. Creating an NGC Certified Virtual Machine using the Azure Console .......................................................................................... 3
   1.4.1. Log in and Launch the VM ....................................................... 3
   1.4.2. Connect to Your VM Instance .................................................. 4
   1.4.3. Start/Stop Your VM Instance ................................................... 5
   1.4.4. Delete VM and Associated Resources ...................................... 5
1.5. Launching an NVIDIA GPU Cloud VM Using Azure CLI ................. 5
   1.5.1. Set Up Environment Variables ............................................... 5
   1.5.2. Launch Your VM Instance ...................................................... 6
   1.5.3. Connect to Your VM Instance .................................................. 7
   1.5.4. Start/Stop Your VM Instance ................................................... 7
   1.5.5. Delete VM and Associated Resources ...................................... 7
1.6. Using Premium Storage SSDs for Datasets ...................................... 8
   1.6.1. Create a Data Disk Using the Azure Console ......................... 8
   1.6.2. Creating a Data Disk Using the Azure CLI .............................. 9
      1.6.2.1. Mounting a Data Disk .................................................... 9
   1.6.3. Deleting a Data Disk ............................................................. 10

## Chapter 2. NVIDIA Virtual Machine Images on Azure

2.1. NVIDIA GPU-Optimized VMI ..................................................... 11
   2.1.1. Information ............................................................................ 11
   2.1.2. Release Notes ....................................................................... 11
2.2. NVIDIA GPU-Optimized VMI with vGPU Driver for A10 Instances ............ 12
   2.2.1. Information ............................................................................ 12
   2.2.2. Release Notes ....................................................................... 12
2.3. NVIDIA GPU-Optimized PyTorch VMI ....................................... 13
   2.3.1. Information ............................................................................ 13
   2.3.2. Release Notes ....................................................................... 13
2.4. NVIDIA GPU-Optimized Image for TensorFlow ............................. 14
   2.4.1. Information ............................................................................ 15
   2.4.2. Release Notes ....................................................................... 15
2.5. NVIDIA HPC SDK GPU-Optimized VM Image ................................................................. 16
  2.5.1. Information .............................................................................................................. 16
  2.5.2. Release Notes .......................................................................................................... 17
2.6. NVIDIA Cloud Native Stack VM Image ........................................................................ 18
  2.6.1. Information .............................................................................................................. 18
  2.6.2. Release Notes .......................................................................................................... 18

Chapter 3. Known Security Vulnerabilities ........................................................................ 20
Chapter 1. Using NGC on Azure Virtual Machines

NVIDIA GPU-Optimized Virtual Machine Images are available on Microsoft Azure compute instances with NVIDIA A100, T4, and V100 GPUs.

For those familiar with the Azure platform, the process of launching the instance is as simple as logging into Azure, selecting the NVIDIA GPU-optimized Image of choice, configuring settings as needed, then launching the VM. After launching the VM, you can SSH into it and start building a host of AI applications in deep learning, machine learning and data science leveraging the plethora of GPU-accelerated containers, pre-trained models and resources from NGC.

This document provides step-by-step instructions for accomplishing this, including how to use the Azure CLI.

1.1. Security Best Practices

Cloud security starts with the security policies of your CSP account. Refer to the following link for how to configure your security policies for your CSP:

- Azure security fundamentals documentation

Users must follow the security guidelines and best practices of their CSP to secure their VM and account.

1.2. Prerequisites

These instructions assume the following:

- You have an Azure account - https://portal.azure.com, with either permissions to create a Resource Group or with a Resource Group already available to you.
- Browse the NGC website and identified an available NGC container and tag to run on the VirtualMachine Instance (VMI).
- If you plan to use the Azure CLI or Terraform, then the Azure CLI 2.0 must be installed.
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).

1.3. Before You Start

Be sure you are familiar with the information in this chapter before starting to use the NVIDIA GPU Cloud Machine Image on Microsoft Azure.

1.3.1. Setting Up SSH Keys

If you do not already have SSH keys set up specifically for Azure, you will need to set one up and have it on the machine you will use to SSH to the VM. In the examples, the key is named “azure-key”.

On Linux or OS X, generate a new key with the following command:

```bash
ssh-keygen -t rsa -b 2048 -f ~/.ssh/azure-key
```

On Windows, the location will depend on the SSH client you use, so modify the path above in the snippets or in your SSH client configuration.

Alternatively, you could also choose to authenticate using a username and password that can be setup while creating the VM. However, the SSH key method ensures optimal security.


1.3.2. Setting Up a Security Group

When creating your NVIDIA GPU Cloud VM, Azure sets up a network security group for the VM and you should choose to allow external access to inbound ports 22 (for SSH) and 443 (for HTTPS). You can add inbound rules to the network security group later for other ports as needed, such as port 8888 for DIGITS.

You can also set up a separate network security group so that it will be available any time you create a new NVIDIA GPU Cloud VM. This can be done ahead of time. Refer to the Microsoft instructions to Create, Change, or Delete a Network Security Group.

Add the following inbound rules to your network security group:

- **SSH**
  - Destination port ranges: 22
  - Protocol: TCP
  - Name: SSH

- **HTTPS**
  - Destination port ranges: 443
  - Protocol: TCP
Using NGC on Azure Virtual Machines

- Name: HTTPS
- **Others as needed**
  - Example: DIGITS
  - Destination port ranges: 8888
  - Protocol: TCP
  - Name: DIGITS

**Security Warning**

It is important to use proper precautions and security safeguards prior to granting access, or sharing your AMI over the internet. By default, internet connectivity to the AMI instance is blocked. You are solely responsible for enabling and securing access to your AMI. Please refer to Azure guides for managing security groups.

1.4. Creating an NGC Certified Virtual Machine using the Azure Console

1.4.1. Log in and Launch the VM

2. Select **Create a Resource** from the Azure Services menu.

3. On the **New** pane, search for “nvidia”, and then select the **NVIDIA GPU-Optimized image** of your choice from the list.

4. Select the latest release version [or version of your choice if required] from the software plan menu and then click **Create**
5. Complete the settings under the **Basics** tab as follows:

- **Subscription and Resource Group**: Choose relevant options to your subscription
- **Virtual Machine Name**: Name of choice
- **Region**: Select a region with instance types featuring the latest NVIDIA GPUs (NC-v3 Series). In this example we use the (US) East US region. A list of available instance types by region can be found [here](#).
- **Authentication Choice**: SSH, with username of choice
- **SSH public key**: Paste in your SSH public key that you previously generated

6. Click **Next** to select a Premium SSH and add data disks.

7. In the Networking section, select the Network Security Group you created earlier under the **Configure network security group** option.

8. Make other Settings selections as needed, then click **OK**.

   After the validation passes, the portal presents the details of your new image which you can download as a template to automate deployment later.

9. Click **Deploy** to deploy the image.

   The deployment starts, as indicated by the traveling bar underneath the Alert icon.

   It may take a few minutes to complete.

### 1.4.2. Connect to Your VM Instance

1. Open the VM instance that you created.
   
   a). Navigate to the Azure portal home page and click on Virtual Machines under the Azure services menu.
   
   b). Select the VM you created and want to connect to.

2. Click **Connect** from the action bar at the top and then select SSH.

   If the instructions to log in via SSH login do not work, refer to [Troubleshooting SSH connections to an Azure Linux VM that fails, errors out, or is refused](#) documentation for further troubleshooting.
1.4.3. **Start/Stop Your VM Instance**

1. Open the VM instance you created.
   a). Navigate to the Azure portal home page and click on Virtual Machines under the Azure services menu.
   b). Select the VM you created and want to manage.
2. Click **Start** or **Stop** from the action bar at the top.

1.4.4. **Delete VM and Associated Resources**

When you created your VM, other resources for that instance were automatically created for you, such as a network interface, public IP address, and boot disk. If you deleted your VM, you will also need to delete these resources.

1. Open the VM instance you created.
   a). Navigate to the Azure portal home page and click on Virtual Machines under the Azure services menu.
   b). Select the VM you created and want to delete.
2. Click **Delete** from the action bar at the top and confirm your choice by typing 'yes' on the pane to the left that pops up.

1.5. **Launching an NVIDIA GPU Cloud VM Using Azure CLI**

If you plan to use Azure CLI, then the CLI must be [installed](#).

Some of the CLI snippets in these instructions make use of **jq**, which should be [installed](#) on the machine from which you’ll run the CLI. You may paste these snippets into your own bash scripts or type them at the command line.

1.5.1. **Set Up Environment Variables**

Use the following table as a guide for determining the values you will need for creating your GPU Cloud VM. The variable names are arbitrary and used in the instructions that follow.

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>DESCRIPTION</th>
<th>EXAMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZ_VM_NAME</td>
<td>Name for your GPU Cloud VM</td>
<td>my-nvgpu-vmi</td>
</tr>
<tr>
<td><strong>AZRESOURCEGROUP</strong></td>
<td>Your resource group</td>
<td><strong>ACME_RG</strong></td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>AZIMAGE</strong></td>
<td>The NVIDIA GPU-Optimized Image. See the release notes NVIDIA Virtual Machine Images on Azure for the latest release.</td>
<td>NVIDIA-GPU-Cloud-Image</td>
</tr>
<tr>
<td><strong>AZLOCATION</strong></td>
<td>A zone that contains GPUs. Refer to <a href="https://azure.microsoft.com/en-us/global-infrastructure/services/">https://azure.microsoft.com/en-us/global-infrastructure/services/</a> to see available locations for NCv2 and NCv3 series SKUs.</td>
<td>westus2</td>
</tr>
<tr>
<td><strong>AZSIZE</strong></td>
<td>The SKU specified by the number of vCPUs, RAM, and GPUs. Refer to <a href="https://docs.microsoft.com/en-us/azure/virtual-machines/linux/sizes-gpu">https://docs.microsoft.com/en-us/azure/virtual-machines/linux/sizes-gpu</a> for the list of P40, P100, and V100 SKUs to choose from.</td>
<td>NC6s_v2</td>
</tr>
<tr>
<td><strong>AZSSHKEY</strong></td>
<td>&lt;path&gt;/&lt;public-azure-key.pub&gt;</td>
<td>~/.ssh/azure-key.pub</td>
</tr>
<tr>
<td><strong>AZUSER</strong></td>
<td>Your username</td>
<td>jsmith</td>
</tr>
<tr>
<td><strong>AZNSG</strong></td>
<td>Your network security group</td>
<td>my-nvgpu-nsg</td>
</tr>
</tbody>
</table>

### 1.5.2. Launch Your VM Instance

Be sure you have installed Azure CLI and that you are ready with the VM setup information listed in the section **Set Up Environment Variables**. You can then either manually replace the variable names in the commands in this section with the actual values or define the variables ahead of time.

1. Log in to the Azure CLI.
   
   ```bash
   az login
   ```

2. Enter the following:
   
   ```bash
   az vm create \\n   --name ${AZ_VM_NAME} \\n   --resource-group ${AZRESOURCEGROUP} \\n   --image ${AZIMAGE} \\n   --location ${AZLOCATION} \\n   --size ${AZSIZE} \\n   --admin-username ${AZUSER} \\n   --ssh-key-value ${AZSSHKEY} \\n   --nsg ${AZNSG}
   ```
   
   If successful, you should see output consisting of a JSON description of your VM. The GPU Cloud VM gets deployed.
Note the public IP address for use when establishing an SSH connection to the VM. You can also set up an AZ_PUBLIC_IP variable by defining an Azure JSON file for the VM as follows:

```bash
AZ_JSON=$(az vm create \
    --name ${AZURE_VM_NAME} \
    --resource-group ${AZ_RESOURCE_GROUP} \
    --image ${AZ_IMAGE} \
    --location ${AZ_LOCATION} \
    --size ${AZ_SIZE} \
    --ssh-key-value ${AZ_SSH_KEY} \
    --admin-username ${AZ_USER} \
    --nsg ${AZ_NSG})
AZ_PUBLIC_IP=$(echo $AZ_JSON | jq .publicIpAddress | sed 's/"//g') && \
    echo $AZ_JSON && echo AZ_PUBLIC_IP=$AZ_PUBLIC_IP
```

Azure sets up a non-persistent scratch disk for each VM. See the sections Persistent Data Storage for Azure Virtual Machines for instructions on setting up alternate storage for your datasets.

1.5.3. Connect to Your VM Instance

Using a CLI on Mac or Linux (Windows users: use OpenSSH on Windows PowerShell or use the Windows Subsystem for Linux), run ssh to connect to your GPU VM instance.

```bash
ssh -i $AZ_SSH_KEY $AZ_USER@$AZ_PUBLIC_IP
```

1.5.4. Start/Stop Your VM Instance

VMs can be stopped and started again without losing any of their storage and other resources.

To stop and deallocate a running VM:

```bash
az vm deallocate --resource-group $AZ_RESOURCE_GROUP --name $AZ_VM_NAME
```

To start a stopped VM:

```bash
az vm start --resource-group $AZ_RESOURCE_GROUP --name $AZ_VM_NAME
```

When starting a stopped VM, you will need to update the public IP variable, as it will change with the newly started VM.

```bash
AZ_PUBLIC_IP=$(az network public-ip show \
    --resource-group $AZ_RESOURCE_GROUP \n    --name $AZ_VM_NAME\PublicIP | jq .ipAddress | sed 's/"//g') && \
    echo AZ_PUBLIC_IP=$AZ_PUBLIC_IP
```

1.5.5. Delete VM and Associated Resources

When you created your VM, other resources for that instance were automatically created for you, such as a network interface, public IP address, and boot disk. If you deleted your instance, you will also need to delete these resources.

Perform the deletions in the following order.

1. Delete your VM.
   ```bash
   az vm delete -g $AZ_RESOURCE_GROUP -n $AZ_VM_NAME
   ```
2. Delete the VM OS disk.
a). List the disks in your Resource Group.

```
az disk list -g $AZ_RESOURCE_GROUP
```

The associated OS disk will have the name of your VM as the base name.

b). Delete the OS disk.

```
az disk delete -g $AZ_RESOURCE_GROUP -n MyDisk
```


3. Delete the VM network interface.

a). List the network interface resources in your Resource Group.

```
az network nic list -g $AZ_RESOURCE_GROUP
```

The associated network interface will have the name of your VM as the base name.

b). Delete the network interface resource.

```
az network nic delete -g $AZ_RESOURCE_GROUP -n MyNic
```


4. Delete the VM public IP address.

a). List the public IPs in your Resource Group.

```
az network public-ip list -g $AZ_RESOURCE_GROUP
```

The associated public IP will have the name of your VM as the base name.

b). Delete the public IP.

```
az network public-ip delete -g $AZ_RESOURCE_GROUP -n MyIp
```


### 1.6. Using Premium Storage SSDs for Datasets

You can create Premium Storage SSD from the Azure dashboard. Premium Storage SSDs are ideal for persistent storage of a large number of datasets and offer better performance.

#### 1.6.1. Create a Data Disk Using the Azure Console

1. Open the VM instance that you created.
   a). Navigate to the Azure portal home page and click on Virtual Machines under the Azure services menu.
   b). Select the VM that you created and want to manage.

2. Select Disks under the Settings category in the control panel on the left.
3. Click **Add Disk** and click **Create Disk** in the drop down menu upon clicking Name.

4. On the Create Managed Disk pane, Enter a disk name Select a resource group Select Premium SSD for Account type Enter a disk size

5. Click **Create**.

6. When the validation is completed, click **Save**.

### 1.6.2. Creating a Data Disk Using the Azure CLI

To create a new data disk and attach it to your VM, include the following option in the `az vm create` command.

```bash
--data-disk-sizes-gb <data-disk-size>
```

To attach an existing data disk to your VM when creating it, include the following option in the `az vm create` command.

```bash
--attach-data-disks <data-disk-name>
```

### 1.6.2.1. Mounting a Data Disk

1. Once the data disk is created, establish an SSH connection to your VM.

2. Create a filesystem on the data disk.

   You can view the volume by running `lsblk` command.

   ```bash
   ~# lsblk
   
   NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   sdb 8:16 0 1.5T 0 disk
   └─sdb1 8:17 0 1.4T 0 part /mnt
   sr0 11:0 1 628K 0 rom
   sdc 8:32 0 2T 0 disk
   ```
### 3. Mount the volume to a mount directory.

```bash
~# mount /dev/sdc1 /data
```

To mount the volume automatically every time the VM is stopped and restarted, add an entry to `/etc/fstab`.

When adding an entry to `/etc/fstab`, use a UUID based device path [See device-names-problem for details].

For example:

```
UUID=33333333-3b3b-3c3c-3d3d-3e3e3e3e3e3e /data ext4 defaults,nofail 1 2
```

### 1.6.3. Deleting a Data Disk

You can delete a Data Disk only if it not attached to a VM. Be aware that once you delete a Data Disk, you cannot undo the action.

1. Open the Azure Dashboard and click All resources from the left side menu.
2. Filter by Disks type, then locate and select the check box for your data disk.
3. Click Delete.
4. Enter ‘yes’ to confirm, then click Delete.
Chapter 2. NVIDIA Virtual Machine Images on Azure

NVIDIA makes available on the Microsoft Azure platform a customized machine image based on the NVIDIA® Tesla Volta™ and Pascal™ GPUs. Running NVIDIA GPU Cloud containers on this instance provides optimum performance for deep learning, machine learning, and HPC workloads.

See the Using NGC with Azure Setup Guide for instructions on setting up and using the VMI.

2.1. NVIDIA GPU-Optimized VMI

2.1.1. Information

The NVIDIA GPU-Optimized VMI is a virtual machine image for accelerating your Machine Learning, Deep Learning, Data Science and HPC workloads. Using this AMI, you can spin up a GPU-accelerated Azure Compute VM instance in minutes with a pre-installed Ubuntu OS, GPU driver, Docker and NVIDIA container toolkit.

Moreover, this VMI provides easy access to NVIDIA’s NGC Catalog, a hub for GPU-optimized software, for pulling & running performance-tuned, tested, and NVIDIA certified docker containers. NGC provides free access to containerized AI, Data Science, and HPC applications, pre-trained models, AI SDKs and other resources to enable data scientists, developers, and researchers to focus on building solutions, gathering insights, and delivering business value.

This GPU-optimized VMI is provided free of charge for developers with an enterprise support option. For more information on enterprise support, please visit NVIDIA AI Enterprise.

2.1.2. Release Notes

Version 22.06.0

- Ubuntu Server 20.04
- NVIDIA Driver 515.48.07
NVIDIA Virtual Machine Images on Azure

Key Changes

- Updated NVIDIA Driver to 515.48.07
- Updated Docker-ce to 20.10.17
- Updated Nvidia Container Toolkit to Version 1.10.0-1
- Updated Nvidia Container Runtime to Version 3.10.0-1
- Packaged additional tools: Miniconda, JupyterLab, NGC-CLI, Git, Python3-PIP

Version 22.03.0

- Ubuntu Server 20.04
- NVIDIA Driver 470.103.01
- Docker-ce 20.10.12
- NVIDIA Container Toolkit 1.8.1
- NVIDIA Container Runtime 3.8.1
- Azure Command Line Interface [CLI]

2.2. NVIDIA GPU-Optimized VMI with vGPU Driver for A10 Instances

2.2.1. Information

The NVIDIA GPU-Optimized VMI with vGPU driver for A10 instances is a virtual machine image for accelerating your Machine Learning, Deep Learning, Data Science, and HPC workloads on Azure’s NVdsA10 v5-series instances. Using this AMI, you can spin up a GPU-accelerated Azure Compute VM instance with NVIDIA A10 GPU in minutes with a pre-installed Ubuntu OS, virtual GPU driver, Docker, and NVIDIA container toolkit with other CLI tools.

2.2.2. Release Notes
Version 22.08.0

- Ubuntu Server 20.04
- NVIDIA Driver 510.73.08
- Docker-ce 20.10.17
- NVIDIA Container Toolkit 1.10.0-1
- NVIDIA Container Runtime 3.10.0-1
- Azure Command Line Interface (CLI)
- Miniconda 4.13.0
- JupyterLab 3.4.3 and other Jupyter Core packages
- NGC CLI 3.4.1
- Git, Python3-PIP

2.3. NVIDIA GPU-Optimized PyTorch VMI

2.3.1. Information

NVIDIA NGC is the hub for GPU-optimized software for deep learning, machine learning, and high-performance computing (HPC). NGC provides free access to performance validated containers, pre-trained models, AI SDKs and other resources to enable data scientists, developers, and researchers to focus on building solutions, gathering insights, and delivering business value.

NVIDIA’s GPU-Optimized PyTorch container included in this image is optimized and updated on a monthly basis to deliver incremental software-driven performance gains from one version to another, extracting maximum performance from your existing GPUs. Combined with quick and easy access to any asset on NGC, this VM image helps fast track your end-to-end AI deployment and development process.

Supported Azure VM instances types are NCv2, NCv3, and ND series.

2.3.2. Release Notes

Version 22.10.0

- Ubuntu Server 20.04
- NVIDIA Driver 515.65.01
- Docker-ce 20.10.17
- NVIDIA Container Toolkit 1.10.1
- NVIDIA Container Runtime 3.10.1
NVIDIA Virtual Machine Images on Azure

- NVIDIA’s GPU-optimized PyTorch container 22.08-py3

Key Changes
- Updated NVIDIA Driver to 515.65.01
- Updated Docker Engine to 20.10.17
- Updated NVIDIA Container Toolkit to 1.10.1
- Updated NVIDIA Container Runtime to 3.10.1
- Updated NVIDIA PyTorch container to 22.08-py3

Version 22.03.0
- Ubuntu Server 20.04
- NVIDIA Driver 470.103.01
- Docker-ce 20.10.12
- NVIDIA Container Toolkit 1.8.1
- NVIDIA Container Runtime 3.8.1
- NVIDIA’s GPU-optimized PyTorch container 22.02-py3

Key Changes
- Updated NVIDIA Driver to 470.103.01
- Updated Docker Engine to 20.10.12
- Updated NVIDIA Container Toolkit to 1.8.1
- Updated NVIDIA Container Runtime to 3.8.1
- Updated NVIDIA PyTorch container to 22.02-py3

Version 21.11.0
- Ubuntu Server 20.04
- NVIDIA Driver Version: 470.82.01
- Docker-CE Version: 20.10.10
- NVIDIA Container Toolkit Version: 1.5.1-1
- NVIDIA Container Runtime: 3.5.0-1
- Azure CLI
- NVIDIA PyTorch Tag Version: 21.10-py3

2.4. NVIDIA GPU-Optimized Image for TensorFlow
2.4.1. Information

NVIDIA NGC is the hub for GPU-optimized software for deep learning, machine learning, and high-performance computing (HPC). NGC provides free access to performance validated containers, pre-trained models, AI SDKs and other resources to enable data scientists, developers, and researchers to focus on building solutions, gathering insights, and delivering business value.

NVIDIA’s GPU-Optimized PyTorch container included in this image is optimized and updated on a monthly basis to deliver incremental software-driven performance gains from one version to another, extracting maximum performance from your existing GPUs. Combined with quick and easy access to any asset on NGC, this VM image helps fast track your end-to-end AI deployment and development process.

Supported Azure VM instance types are NCv2, NCv3, and ND series.

2.4.2. Release Notes

Version 22.10.0

- Ubuntu Server 20.04
- NVIDIA Driver 515.65.01
- Docker-ce 20.10.17
- NVIDIA Container Toolkit 1.10.1
- NVIDIA Container Runtime 3.10.1
- NVIDIA’s distribution of TensorFlow 1 and 2, tags 22.08-tf2-py3 and 22.08-tf1-py3

Key Changes

- Updated NVIDIA Driver to 515.65.01
- Updated Docker Engine to 20.10.17
- Updated NVIDIA Container Toolkit to 1.10.1
- Updated NVIDIA Container Runtime to 3.10.1
- Updated NVIDIA Tensorflow container to 22.08-py3

Version 22.03.0

- Ubuntu Server 20.04
- NVIDIA Driver 470.103.01
- Docker-ce 20.10.12
- NVIDIA Container Toolkit 1.8.1
- NVIDIA Container Runtime 3.8.1
NVIDIA’s distribution of TensorFlow 1 and 2, tags 22.02-tf2-py3 and 22.02-tf1-py3

Key Changes

- Updated NVIDIA Driver to 470.103.01
- Updated Docker Engine to 20.10.12
- Updated NVIDIA Container Toolkit to 1.8.1
- Updated NVIDIA Container Runtime to 3.8.1
- Updated NVIDIA Tensorflow container to 22.02-py3

Version 21.11.0

- Ubuntu Server 20.04 NVIDIA
- Driver Version: 470.82.01
- Docker-CE Version: 20.10.10
- NVIDIA Container Toolkit Version: 1.5.1-1
- NVIDIA Container Runtime: 3.5.0-1
- Azure CLI
- NVIDIA TensorFlow Tags 21.10-tf1-py3 and 21.10-tf2-py3

2.5. NVIDIA HPC SDK GPU-Optimized VM Image

2.5.1. Information

The NVIDIA HPC SDK C, C++, and Fortran compilers support GPU acceleration of HPC modeling and simulation applications with standard C++ and Fortran, OpenACC directives, and CUDA. GPU-accelerated math libraries maximize performance on common HPC algorithms, and optimized communications libraries enable standards-based multi-GPU and scalable systems programming. Performance profiling and debugging tools simplify porting and optimization of HPC applications, and containerization tools enable easy deployment on-premises or in the cloud.

Key features of the NVIDIA HPC SDK for Linux include:

- Support for NVIDIA Ampere Architecture GPUs with FP16, TF32 and FP64 tensor cores
- NVC++ ISO C++17 compiler with Parallel Algorithms acceleration on GPUs, OpenACC and OpenMP
- NVFORTRAN ISO Fortran 2003 compiler with array intrinsics acceleration on GPUs, CUDA Fortran, OpenACC and OpenMP
- NVC ISO C11 compiler with OpenACC and OpenMP
NVIDIA Virtual Machine Images on Azure

- NVCC NVIDIA CUDA C++ compiler
- NVIDIA Math Libraries including cuBLAS, cuSOLVER, cuSPARSE, cuFFT, cuTENSOR and cuRAND
- Thrust, CUB, and libcu++ GPU-accelerated libraries of C++ parallel algorithms and data structures
- NCCL, NVSHMEM and Open MPI libraries for fast multi-GPU/multi-node communications
- NVIDIA Nsight Systems/Compute for interactive HPC applications performance profiler

2.5.2. Release Notes

Version 22.08.0
- Ubuntu Server 20.04
- NVIDIA Driver 515.65.01
- Docker-ce 20.10.17
- NVIDIA Container Toolkit Version: 1.10.1-1
- NVIDIA Container Runtime Version: 3.10.0-1
- Azure Command Line Interface (CLI)

Key Changes
- Updated NVIDIA Driver to 515.48.07
- Updated Docker-ce to 20.10.17
- Updated NVIDIA Container Toolkit to Version 1.10.0-1
- Updated NVIDIA Container Runtime to Version 3.10.0-1

Known Issues
- The version of Nsight Systems bundled with the HPC SDK 22.7 fails with the error ‘Agent launcher failed’ on some instance types. The issue is fixed in Nsight Systems version 2022.3.4 and later, which can be installed separately from the Nsight Systems downloads page. For more information, refer to the Nsight Systems documentation.

Version 22.03.0
- Ubuntu Server 20.04
- NVIDIA Driver Version: 470.103.01
- Docker-ce 20.10.12
- NVIDIA Container Toolkit Version: 1.8.1-1
- NVIDIA Container Runtime Version: 3.8.1-1
- MOFED Version: 5.5-1.0.3.2
- NVIDIA Peer Memory Version: 1.3
NVIDIA Virtual Machine Images on Azure

Key Changes

- Updated Docker-ce to 20.10.12
- Updated NVIDIA Container Toolkit to Version 1.8.1-1
- Updated NVIDIA Container Runtime to Version 3.8.1-1
- Updated NVIDIA MOFED to Version 5.5-1.0.3.2
- Updated NVIDIA Peer Memory to Version 1.3
- Updated NVIDIA HPC SDK Version: 22.3

Version 22.01.0

- Ubuntu Server 20.04
- NVIDIA Driver 470.103.01
- Docker-ce 20.10.11
- NVIDIA Container Toolkit 1.7.0-1
- NVIDIA Container Runtime 3.7.0-1
- MOFED Version: 5.4-1.0.3.0
- NVIDIA Peer Memory Version: 1.2
- NVIDIA HPC SDK Version: 22.1

2.6. NVIDIA Cloud Native Stack VM Image

2.6.1. Information

NVIDIA Cloud Native Stack VMI is a GPU-accelerated VMI that is pre-installed with Cloud Native Stack, which is a reference architecture that includes upstream Kubernetes and the NVIDIA GPU and Network Operator. NVIDIA Cloud Native Stack VMI allows developers to build, test and run GPU-accelerated containerized applications that are orchestrated by Kubernetes.

2.6.2. Release Notes

Version 6.2

- Ubuntu Server 20.04
- Containerd 1.6.5
- Kubernetes 1.23.8
- Helm 3.8.2
- GPU Operator 1.11.0
- NVIDIA Driver 515.65.01
Chapter 3. Known Security Vulnerabilities

The NVIDIA GPU-Optimized VMI includes conda by default in order to use jupyter-lab notebooks. The internal Python dependencies may be patched in newer Python versions, but conda must use the specific versions in the VMI. These vulnerabilities are not directly exploitable unless there is a vulnerability in conda itself. An attacker would need to obtain access to a VM running conda, so it is important that VM access must be protected. See the security best practices section.

The following releases are affected by the vulnerabilities:

- NVIDIA GPU-Optimized VMI 22.06
- NVIDIA GPU-Optimized VMI (ARM64) 22.06

The list of vulnerabilities are:

- GHSA-3gh2-xw74-jmcw: High; Django 2.1; SQL injection
- GHSA-6r97-cj55-9hrq: Critical; Django 2.1; SQL injection
- GHSA-c4qh-4vgv-qc6g: High; Django 2.1; Uncontrolled resource consumption
- GHSA-h5jv-4p7w-64jg: High; Django 2.1; Uncontrolled resource consumption
- GHSA-hmr4-m2h5-33qx: Critical; Django 2.1; SQL injection
- GHSA-v6rh-hp5x-86rv: High; Django 2.1; Access control bypass
- GHSA-v9qq-3j8p-r63v: High; Django 2.1; Uncontrolled recursion
- GHSA-vfq6-hq5r-27r6: Critical; Django 2.1; Account hijack via password reset form
- GHSA-wh4h-v3f2-r2pp: High; Django 2.1; Uncontrolled memory consumption
- GHSA-32gy-6cf3-wcmq: Critical; Twisted 18.7.0; HTTP/2 DoS attack
- GHSA-65rm-h285-5cc5: High; Twisted 18.7.0; Improper certificate validation
- GHSA-92x2-jw7w-xvxx: High; Twisted 18.7.0; Cookie and header exposure
- GHSA-c2jg-hw38-jrqq: High; Twisted 18.7.0; HTTP request smuggling
- GHSA-h96w-mmrf-2h6v: Critical; Twisted 18.7.0; Improper input validation
- GHSA-p5xh-vx83-mxcj: Critical; Twisted 18.7.0; HTTP request smuggling
- GHSA-5545-2q6w-2gh6: High; numpy 1.15.1; NULL pointer dereference
- CVE-2019-6446: Critical; numpy 1.15.1; Deserialization of untrusted data
- GHSA-h4m5-qpfp-3mpv: High; Babel 2.6.0; Arbitrary code execution
- GHSA-ffqj-6fqr-9h24: High; PyJWT 1.6.4; Key confusion through non-blocklisted public key formats
- GHSA-h7wm-ph43-c39p: High; Scrapy 1.5.1; Uncontrolled memory consumption
- CVE-2022-39286: High; jupyter_core 4.11.2; Arbitrary code execution
- GHSA-55x5-fj6c-h6m8: High; lxml 4.2.4; Crafted code allowed through lxml HTML cleaner
- GHSA-wrxv-2j5q-m38w: High; lxml 4.2.4; NULL pointer dereference
- GHSA-gpvw-69j7-gwj8: High; pip 8.1.2; Path traversal
- GHSA-hj5v-574p-mj7c: High; py 1.6.0; Regular expression DoS
- GHSA-x84v-xcm2-53pg: High; requests 2.19.1; Insufficiently protected credentials
- GHSA-mh33-7rrq-662w: High; urllib3 1.2.3; Improper certificate validation
- CVE-2021-33503: High; urllib3 1.2.3; Denial of service attack
- GHSA-2m34-jcjv-45xf: Medium; Django 2.1; XSS in Django
- GHSA-337x-4q8g-prc5: Medium; Django 2.1; Improper input validation
- GHSA-68w8-qjq3-2gfm: Medium; Django 2.1; Path traversal
- GHSA-6c7v-2f49-8h26: Medium; Django 2.1; Cleartext transmission of sensitive information
- GHSA-6mx3-3vqg-hpp2: Medium; Django 2.1; Django allows unprivileged users can read the password hashes of arbitrary accounts
- GHSA-7rp2-fm2h-wchj: Medium; Django 2.1; XSS in Django
- GHSA-hvmf-r92r-27hr: Medium; Django 2.1; Django allows unintended model editing
- GHSA-wpjr-j57x-wxfw: Medium; Django 2.1; Data leakage via cache key collision in Django
- GHSA-9x8m-2xfp-crp3: Medium; Scrapy 1.5.1; Credentials leakage when using HTTP proxy
- GHSA-cjvr-mfj7-j4j8: Medium; Scrapy 1.5.1; Incorrect authorization and information exposure
- GHSA-jwqp-28gf-p498: Medium; Scrapy 1.5.1; Credential leakage
- GHSA-mfjm-vh54-3f96: Medium; Scrapy 1.5.1; Cookie-setting not restricted
- GHSA-6cc5-2vg4-cc7m: Medium; Twisted 18.7.0; Injection of invalid characters in URI/ method
- GHSA-8r99-h8j2-rw64: Medium; Twisted 18.7.0; HTTP Request Smuggling
- GHSA-vg46-2rrj-3647: Medium; Twisted 18.7.0; NameVirtualHost Host header injection
- GHSA-39hc-v87j-747x: Medium; cryptography 37.0.2; Vulnerable OpenSSL included in cryptography wheels
- GHSA-hggm-jpg3-v476: Medium; cryptography 2.3.1; RSA decryption vulnerable to Bleichenbacher timing vulnerability
- GHSA-jq4v-f5q6-mjqj: Medium; lxml 4.2.4; XSS
- GHSA-pgww-xf46-h92r: Medium; lxml 4.2.4; XSS
- GHSA-xp26-p53h-6h2p: Medium; lxml 4.2.4; Improper Neutralization of Input During Web Page Generation in LXML
- GHSA-6p56-wp2h-9hx9: Medium; numpy 1.15.1; NumPy Buffer Overflow, very unlikely to be exploited by an unprivileged user
- GHSA-f7c7-j99h-c22f: Medium; numpy 1.15.1; Buffer Copy without Checking Size of Input in NumPy
- GHSA-fpfv-jqm9-f5jm: Medium; numpy 1.15.1; Incorrect Comparison in NumPy
- GHSA-5xp3-jfq3-5q8x: Medium; pip 8.1.2; Improper Input Validation in pip
- GHSA-w596-4wxj-9j66: Medium; py 1.6.0; ReDoS in py library when used with subversion
- GHSA-hwfp-hg2m-9vr2: Medium; pywin32 223; Integer overflow in pywin32
- GHSA-r64q-w8jr-g9qp: Medium; urllib3 1.23; Improper Neutralization of CRLF Sequences
- GHSA-wqvq-5m8c-6g24: Medium; urllib3 1.23; CRLF injection
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.