



NGC on Alibaba Cloud Virtual Machines

Release Notes and Setup Guide

Table of Contents

Chapter 1. NGC on Alibaba Cloud Virtual Machines.....	1
1.1. Security Best Practices.....	2
1.2. Before You Get Started.....	2
1.2.1. Set Up Your SSH Key Pair.....	2
1.2.2. Set Up Security Groups for the Virtual Machine.....	3
1.3. Creating an NGC Certified Virtual Machine using the Alibaba Cloud Console.....	4
1.3.1. Log in and Locate the Image.....	4
1.3.2. Configure the VM and Launch.....	5
1.3.3. Connect to Your VM Instance.....	5
1.3.4. Start/Stop/Delete Your VM Instance.....	6
1.4. Creating an NGC Certified Virtual Machine using the Alibaba Cloud CLI.....	6
1.4.1. Install the Alibaba CLI.....	6
1.4.2. Get the NVIDIA Image ID.....	7
1.4.3. Create a VM Instance.....	7
1.4.4. Assign a Public IP Address.....	8
1.4.5. Start the Instance.....	8
1.4.6. Connect to Your VM Instance.....	8
1.4.7. Start/Stop/Delete Your VM Instance.....	9
Chapter 2. NVIDIA Virtual Machine Images on Alibaba Cloud.....	10
2.1. NVIDIA GPU-Optimized VMI.....	10
2.1.1. Information.....	10
2.1.2. Release Notes.....	10
2.2. PyTorch from NVIDIA VMI.....	11
2.2.1. Information.....	11
2.2.2. Release Notes.....	11
2.3. TensorFlow from NVIDIA VMI.....	11
2.3.1. Information.....	11
2.3.2. Release Notes.....	12
Chapter 3. Known Security Vulnerabilities.....	13

Chapter 1. NGC on Alibaba Cloud Virtual Machines

NVIDIA makes available on Alibaba Cloud three different virtual machine images (VMIs). These are GPU-optimized VMIs for Alibaba Cloud VM instances with NVIDIA V100 or NVIDIA T4 GPUs.

- ▶ NVIDIA GPU-Optimized Image for Deep Learning, Machine Learning & HPC
The base GPU-Optimized image Includes Ubuntu Server, the NVIDIA driver, Docker CE, and the NVIDIA Container Runtime for Docker
- ▶ NVIDIA GPU-Optimized Image for TensorFlow
The base image with NVIDIA's GPU-Accelerated TensorFlow container pre-installed
- ▶ NVIDIA GPU-Optimized Image for PyTorch
The base image with NVIDIA's GPU-Accelerated PyTorch container pre-installed

For those familiar with the Alibaba Cloud platform, the process of launching the instance is as simple as logging in, selecting the NVIDIA GPU-optimized Image of choice, selecting and configuring a cloud instance with at least one supported NVIDIA GPU, and then launching the VM. After launching the VM, you can SSH into it and start using the wide range of GPU-accelerated containers, pre-trained models, and other resources available from the NGC Catalog.

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

Prerequisites

These instructions assume the following:

- ▶ You have an Alibaba account - <https://home-intl.console.aliyun.com/> with permissions to create resources.
- ▶ Browse the [NGC website](#) and identified an available NGC container and tag to run on the VMI.
- ▶ 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).

1.1. 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:

- ▶ [Alibaba Cloud Security](#)

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

1.2. Before You Get Started

1.2.1. Set Up Your SSH Key Pair

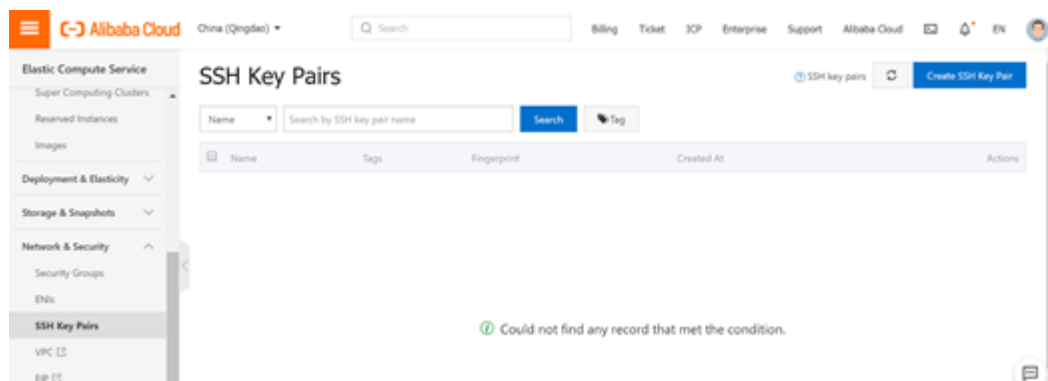
If you do not already have SSH keys set up specifically for Alibaba, you will need to set one up and have it on the machine you will use to SSH to the VM. In the examples, the key is named "alibaba-key".

1. From a browser, log in to the ECS console - <https://ecs.console.aliyun.com/>.
2. Open the left navigation menu tab and then click SSH Key Pairs from the Network & Security group.
3. From the upper right of the screen, click Create SSH Key Pair.
4. Give it a name, such as "alibaba-key" and click OK.
A .pem file will immediately download. This is the ONLY time you can download it.
5. After downloading the .pem file, move it to the .ssh directory.

```
mv alibaba-key.pem ~/.ssh/  
chmod 400 ~/.ssh/alibaba-key.pem
```

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

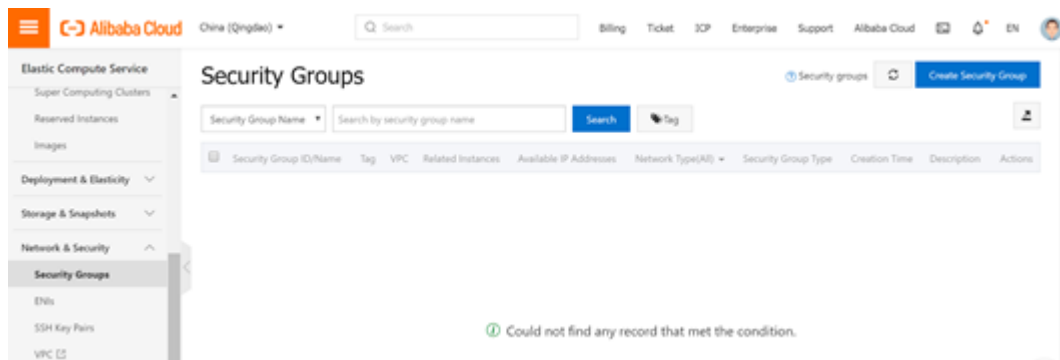
See the Alibaba documentation for [Creating an SSH key pair](#).



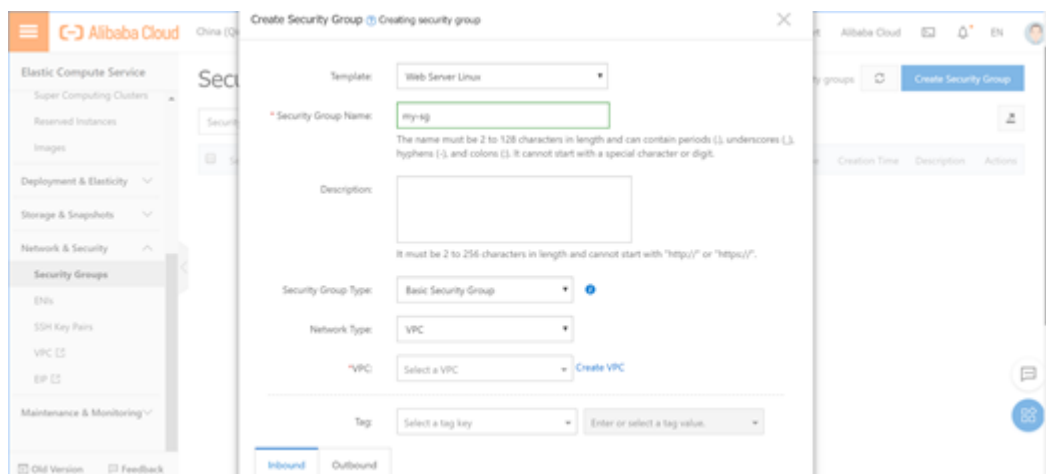
1.2.2. Set Up Security Groups for the Virtual Machine

In order to create instances, you need to put them in a Security Group.

1. Log in to the ECS console - <https://ecs.console.aliyun.com/>.
2. Open the left navigation menu tab and then click Security Groups from the Network & Security group.
3. From the upper right of the screen, click Create Security Group.



4. Give it a name and description, and create a Virtual Private Cloud (VPC) if one doesn't exist yet.
5. Under the inbound tab, configure the following options.
 - a). Add SSH and HTTPS.
 - b). At Custom Port Range, select TCP and then enter 5000/5000.
 - c). Set Authorization Object = 0.0.0.0/0 or the IP address from which you will access.
 - d). Click OK.



Security Warning

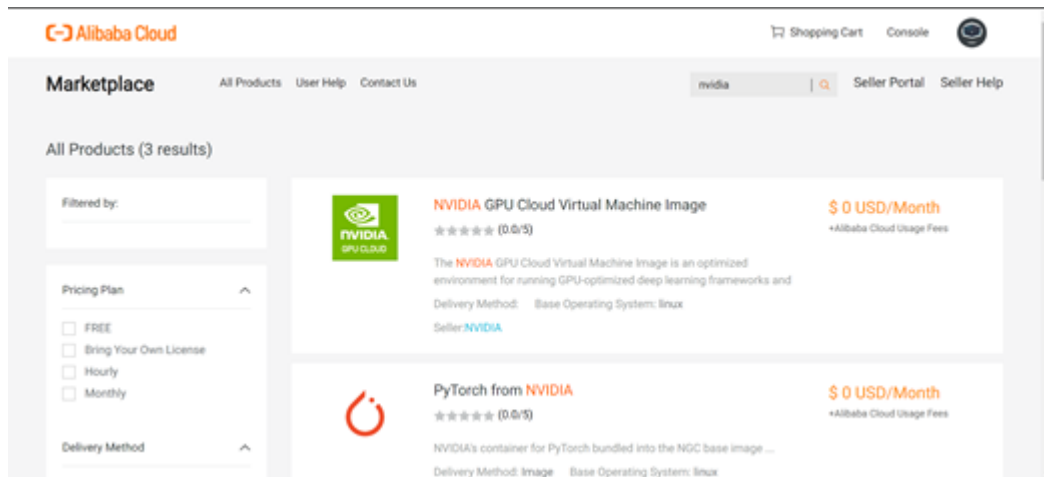
It is important to use proper precautions and security safeguards prior to granting access, or sharing your AMI over the internet. By default, internet connectivity to the

AMI instance is blocked. You are solely responsible for enabling and securing access to your AMI. Please refer to Alibaba guides for managing security groups.

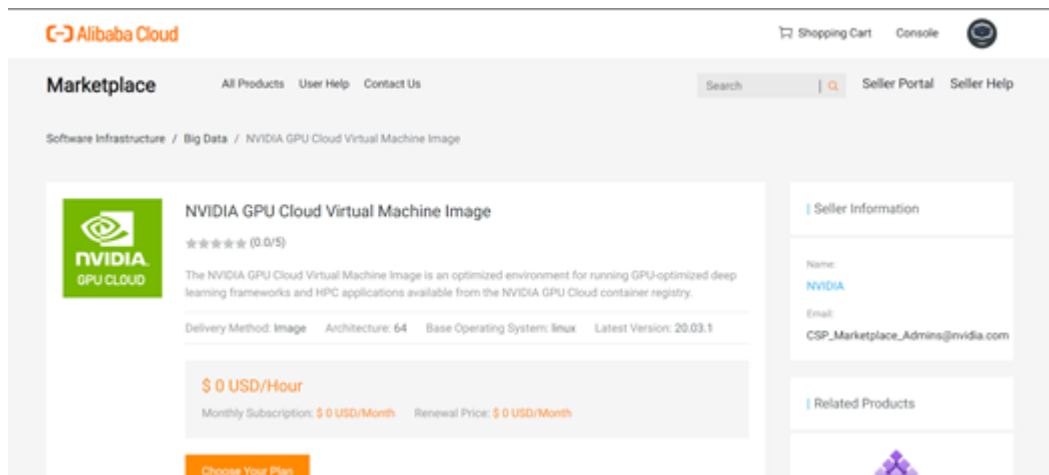
1.3. Creating an NGC Certified Virtual Machine using the Alibaba Cloud Console

1.3.1. Log in and Locate the Image

1. Log in to the Alibaba Console ([Alibaba Cloud Marketplace \(Find and Quickly Use Software as Images\)](#)).
2. Search for nvidia and select the NVIDIA GPU-Optimized image of your choice.

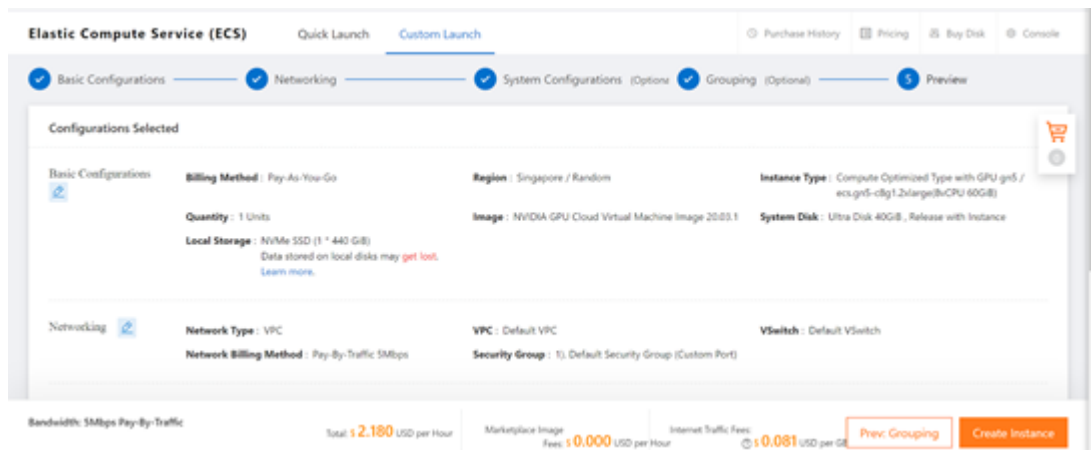


3. Click Choose your plan.



1.3.2. Configure the VM and Launch

1. Configure the following instance settings.
 - ▶ Billing Method: Pay-As-You-Go
 - ▶ Region: [Select a region](#) that has GPU instances (Note: Not all regions have GPUs)
 - ▶ Instance Type: Select Heterogeneous Computing and [select an instance type with NVIDIA V100 or T4 GPUs](#)
 - ▶ Image: Ensure the NVIDIA GPU-Optimized Image you chose previously is selected
 - ▶ Storage: Add a disk for dataset storage by clicking Add Disk under Data Disk, and then entering the storage size. Recommended minimum dataset storage size is 1 TB (1024 GB)
2. Click Next: Networking and select the security group you previously created in the Before You Get Started section.
3. Click Next: System Configuration and select the SSH Key Pair you previously created in the Before You Get Started section.
4. Click Preview, review the configuration and accept the terms of service, and then click Create Instance.



1.3.3. Connect to Your VM Instance

1. Click Console on the Create page.
2. Wait until the status of your VM displays “Running” and then connect via SSH using the actions section of the VM details.
3. Once started, you can SSH into your instance using the SSH key for the root user. If you followed the setup in this tutorial, your key is in `~/ .ssh/`.

Command Syntax

```
$ ssh -i <KEYPATH> root@<IP>
```

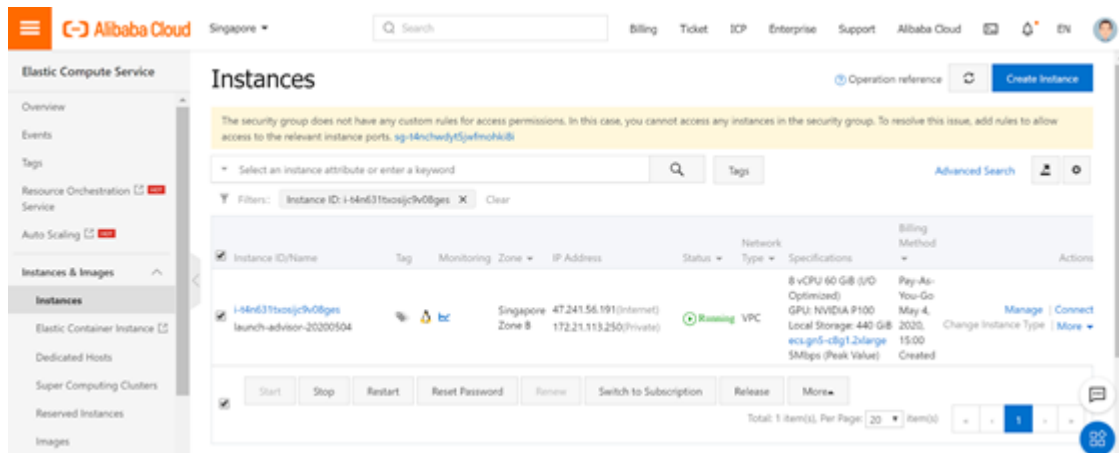
Example

```
$ ssh -i ~/.ssh/alibaba-key.pem root@47.89.248.188
```

Refer to [Connect to a Linux Instance](#) for more instructions on connecting to your instance.

1.3.4. Start/Stop/Delete Your VM Instance

1. Navigate to Instances under the Instances & Images section in the navigation pane on the left.
2. Select the virtual machine instance you wish to manage and use the options bar at the bottom to start/stop, and release to terminate the instance and delete any associated resources.



1.4. Creating an NGC Certified Virtual Machine using the Alibaba Cloud CLI

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 command can have significant delays.

For complete CLI documentation and sample scripts visit the [Alibaba Documentation Center](#).

1.4.1. Install the Alibaba CLI

To use the Alibaba CLI, follow the [Alibaba CLI Install Instructions](#) and also install the ECS SDK.

1. Install the ECS SDK.

```
sudo pip install aliyun-python-sdk-ecs
```

2. Configure the CLI with your keys.

```
aliyuncli configure
```

1.4.2. Get the NVIDIA Image ID

Once started, you can SSH into your instance using the SSH key for the root user. If you followed the setup in this tutorial, your key is in `~/.ssh/`.

You need to specify a source ImageID when creating an instance. Use this command to find the latest ImageID of the NVIDIA-GPU-Cloud-Machine-Image:

```
aliyuncli ecs DescribeImages --RegionId us-west-1 \
  --ImageName "NVIDIA-GPU-Cloud-Virtual-Machine" \
  --output json --filter Images.Image[0].ImageId
```

It will output the Image ID such as "m-rj9iy0xjiod3ghkyhz4p"

1.4.3. Create a VM Instance

Creating an instance with the CLI is done using the `aliyuncli ecs CreateInstance` command.`

Full syntax documentation - <https://www.alibabacloud.com/help/doc-detail/25499.htm>

Recommended Instance Options

- ▶ "--InternetMaxBandwidthOut 10" sets the peak outbound network bandwidth to 10 Mbps. The valid range is [1, 200].
- ▶ "--InstanceChargeType PostPaid" sets the billing method to pay-as-you-go. Change this to "PrePaid" to set it to a subscription billing.

Other Notable Create Instance Options

- ▶ The inbound network bandwidth defaults to 200 Mbps. Use "--InternetMaxBandwidthIn" to change this. The valid range is [1, 200].
- ▶ To change the size of the system disk (default is 40 GB), use the "--SystemDiskSize" option. Valid values are [40, 500].
- ▶ To add a data disk (up to 16), use the "--DataDiskNSize" and "--DataDiskNCategory" options where "N" is [1, 16]. Valid values are:

DataDiskNCategory	DataDiskNSize	Description
cloud	[5, 2000]	(default) Basic cloud disk
cloud_efficiency	[20, 32768]	Ultra cloud disk

cloud_ssd	[20, 32768]	Cloud SSD
ephemeral_ssd	[5, 800]	Ephemeral SSD

Launch Example

Launch the instance and capture the resulting JSON:

```
aliyuncli ecs CreateInstance \
  --RegionId us-west-1 \
  --ImageId "m-rj9iy0xjiod3ghkyhz4p" \
  --SecurityGroupId "sg-rj94krsusal2k5l6gnnz" \
  --InstanceType ecs.gn5-c4g1.xlarge \
  --InstanceName "my-instance" \
  --InternetMaxBandwidthOut 10 \
  --InstanceChargeType PostPaid \
  --KeyPairName alibaba-key
```

The output shows the instance ID.

```
{
  "InstanceId": "i-rj9a0iw25hryafj0fm4v",
  "RequestId": "440ECC70-09F9-492C-AB9E-21AA9C4E0531"
}
```

1.4.4. Assign a Public IP Address

Instances created via CLI are not automatically given a public IP address.

To assign a public IP address to the instance you just created, run:

```
aliyuncli ecs AllocatePublicIpAddress --RegionId us-west-1 \
  --InstanceId "i-rj9a0iw25hryafj0fm4v"
```

Successful completion of the command will return the IP address:

```
{
  "IpAddress": "47.89.248.188",
  "RequestId": "65EB59AE-FA75-446F-B5C7-2BA0F9A77CDC"
}
```

1.4.5. Start the Instance

Instances created via CLI are not started automatically.

To start the instance you just created, run:

```
aliyuncli ecs StartInstance --InstanceId "i-rj9a0iw25hryafj0fm4v"
```

1.4.6. Connect to Your VM Instance

Once started, you can SSH into your instance using the SSH key for the root user. If you followed the setup in this tutorial, your key is in `~/.ssh/`.

Command syntax:

```
ssh -i <KEYPATH> root@<IP>
```

Example:

```
ssh -i ~/.ssh/alibaba-key.pem root@47.89.248.188
```

Refer to [Connect to a Linux Instance](#) for more instructions on connecting to your instance.

1.4.7. Start/Stop/Delete Your VM Instance

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

Stop:

```
aliyuncli ecs StopInstance --InstanceId INSTANCE_ID
```

Start or Restart:

```
aliyuncli ecs StartInstance --InstanceId INSTANCE_ID
```

Delete:

```
aliyuncli ecs DeleteInstance --InstanceId INSTANCE_ID
```

Chapter 2. NVIDIA Virtual Machine Images on Alibaba Cloud

NVIDIA makes available on the Alibaba Cloud platform a customized NGC virtual machine image optimized for the NVIDIA® Volta™ GPU. Running NVIDIA GPU Cloud containers on this instance provides optimum performance for deep learning jobs.

See the [NGC with Alibaba Cloud 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 VMI, you can spin up a GPU-accelerated Alibaba 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
- ▶ Docker-ce 20.10.17
- ▶ NVIDIA Container Toolkit 1.10.0-1
- ▶ NVIDIA Container Runtime 3.10.0-1
- ▶ AWS Command Line Interface (CLI)
- ▶ Miniconda 4.13.0
- ▶ JupyterLab 3.4.3 and other Jupyter core packages
- ▶ NGC-CLI 3.0.0
- ▶ Git, Python3-PIP

2.2. PyTorch from NVIDIA VMI

2.2.1. 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.

2.2.2. Release Notes

Coming soon.

2.3. TensorFlow from NVIDIA 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.

2.3.2. Release Notes

Coming soon.

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-v9qg-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-32gv-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-xvvx: 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-gpvv-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 1.23; Improper certificate validation
- ▶ CVE-2021-33503: High; urllib3 1.23; Denial of service attack
- ▶ GHSA-2m34-jciv-45xf: Medium; Django 2.1; XSS in Django
- ▶ GHSA-337x-4q8g-prc5: Medium; Django 2.1; Improper input validation
- ▶ GHSA-68w8-qj3-2gfm: 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-2xpf-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-9hxr: 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-w596-4wvx-j9j6: 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

© 2025 NVIDIA CORPORATION & AFFILIATES. All rights reserved.