



SDK Installation

Table of contents

Prerequisites

Install the SDK

The section below refers to the installation of the Holoscan SDK referred to as the **development stack**, designed for NVIDIA Developer Kits (arm64), and for x86_64 Linux compute platforms, ideal for development and testing of the SDK.

Note

An alternative for the [IGX Orin Developer Kit](#) is the [deployment stack](#), based on [OpenEmbedded \(Yocto build system\)](#) instead of Ubuntu. This is recommended to limit your stack to the software components strictly required to run your Holoscan application. The runtime Board Support Package (BSP) can be optimized with respect to memory usage, speed, security and power requirements.

Prerequisites

Ingested Tab Module

- For RDMA Support, follow the instructions in the [Enabling RDMA](#) section.
- Additional software dependencies might be needed based on how you choose to install the SDK (see section below).
- Refer to the [Additional Setup](#) and [Third-Party Hardware Setup](#) sections for additional prerequisites.

Install the SDK

We provide multiple ways to install and run the Holoscan SDK:

Instructions

Ingested Tab Module

Not sure what to choose?

- The **Holoscan container image on NGC** is the safest way to ensure all the dependencies are present with the expected versions (including Torch and ONNX Runtime), and should work on most Linux distributions. It is the simplest way to run the embedded examples, while still allowing you to create your own C++ and Python Holoscan application on top of it. These benefits come at a cost:
 - large image size from the numerous (some of them optional) dependencies. If you need a lean runtime image, see [section below](#).
 - standard inconvenience that exist when using Docker, such as more complex run instructions for proper configuration.
- If you are confident in your ability to manage dependencies on your own in your host environment, the **Holoscan Debian package** should provide all the capabilities needed to use the Holoscan SDK, assuming you are on Ubuntu 22.04.
- If you are not interested in the C++ API but just need to work in Python, or want to use a different version than Python 3.10, you can use the **Holoscan python wheels** on PyPI. While they are the easiest solution to install the SDK, it might require the most work to setup your environment with extra dependencies based on your needs. Finally, they are only formally supported on Ubuntu 22.04, though should support other linux distributions with glibc 2.35 or above.

	NGC dev Container	Debian Package	Python Wheels
Runtime libraries	Included	Included	Included
Python module	3.10	3.10	3.8 to 3.11
C++ headers and CMake config	Included	Included	N/A
Examples (+ source)	Included	Included	retrieve from GitHub
Sample datasets	Included	retrieve from NGC	retrieve from NGC
CUDA runtime ¹	Included	automatically ² installed	require manual installation
NPP support ³	Included	automatically ² installed	require manual installation

TensorRT support ⁴	Included	automatically ² installed	require manual installation
Vulkan support ⁵	Included	automatically ² installed	require manual installation
V4L2 support ⁶	Included	automatically ² installed	require manual installation
Torch support ⁷	Included	require manual ⁸ installation	require manual ⁸ installation
ONNX Runtime support ⁹	Included	require manual ¹⁰ installation	require manual ¹⁰ installation
MOFED support ¹¹	User space included Install kernel drivers on the host	require manual installation	require manual installation
CLI support	Included	needs docker w/ buildx plugin	needs docker w/ buildx plugin

Need more control over the SDK?

The [Holoscan SDK source repository](#) is **open-source** and provides reference implementations as well as infrastructure for building the SDK yourself.

Attention

We only recommend building the SDK from source if you need to build it with debug symbols or other options not used as part of the published packages. If you want to write your own operator or application, you can use the SDK as a dependency (and contribute to [HoloHub](#)). If you need to make other modifications to the SDK, [file a feature or bug request](#).

Looking for a light runtime container image?

The current Holoscan container on NGC has a large size due to including all the dependencies for each of the built-in operators, but also because of the development tools and libraries that are included. Follow the [instructions on GitHub](#) to build a runtime container without these development packages. This page also includes detailed documentation to assist you in only including runtime dependencies your Holoscan application might need.

[1]

[CUDA 12](#) is required. Already installed on NVIDIA developer kits with IGX Software and JetPack.

[2]([1,2,3,4,5](#))

Debian installation on x86_64 requires the [latest cuda-keyring package](#) to automatically install all dependencies.

[3]

NPP 12 needed for the FormatConverter and BayerDemosaic operators. Already installed on NVIDIA developer kits with IGX Software and JetPack.

[4]

TensorRT 8.6.1+ and cuDNN needed for the Inference operator. Already installed on NVIDIA developer kits with IGX Software and JetPack.

[5]

Vulkan 1.3.204+ loader needed for the HoloViz operator (+ libegl1 for headless rendering). Already installed on NVIDIA developer kits with IGX Software and JetPack.

[6]

V4L2 1.22+ needed for the V4L2 operator. Already installed on NVIDIA developer kits with IGX Software and JetPack. V4L2 also requires libjpeg.

[7]

Torch support requires LibTorch 2.1+, TorchVision 0.16+, OpenBLAS 0.3.20+, OpenMPI (aarch64 only), MKL 2021.1.1 (x86_64 only), libpng and libjpeg.

[8](1,2)

To install LibTorch and TorchVision, either build them from source, download our [pre-built packages](#), or copy them from the holoscan container (in `/opt`).

[9]

ONNXRuntime 1.15.1+ needed for the Inference operator. Note that ONNX models are also supported through the TensorRT backend of the Inference Operator.

[10](1,2)

To install ONNXRuntime, either build it from source, download our [pre-built package](#) with CUDA 12 and TensorRT execution provider support, or copy it from the holoscan container (in `/opt/onnxruntime`).

[11]

Tested with MOFED 23.10

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024