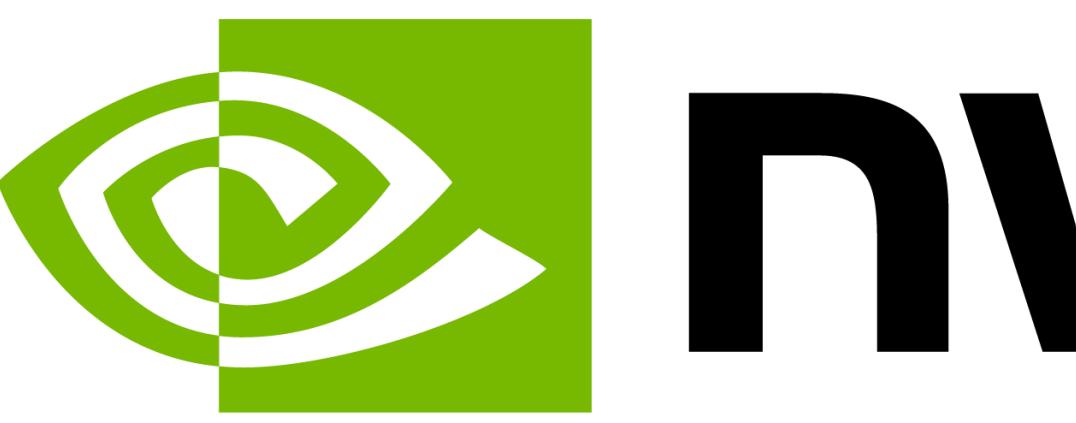


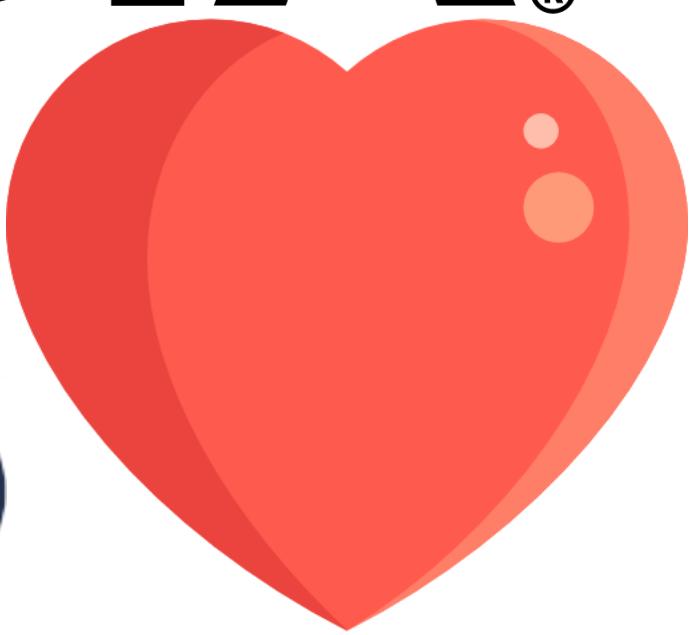


Be an Isaac ROS DevOps Hero with Containerized Development

Raffaello Bonghi, Technical marketing | Isaac ROS Webinar / February 14th, 2023

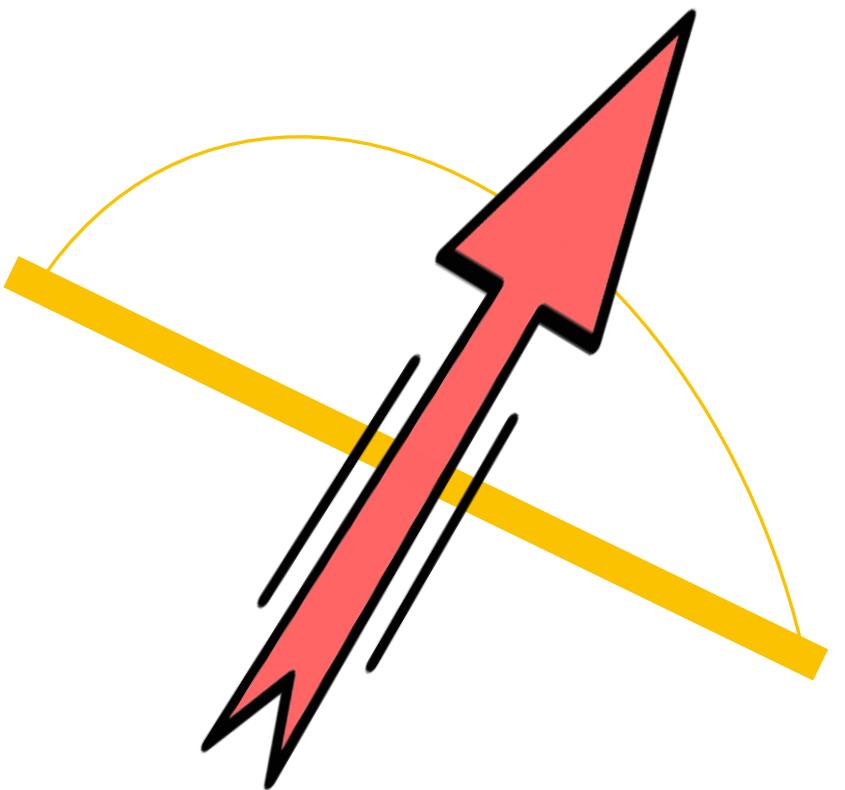
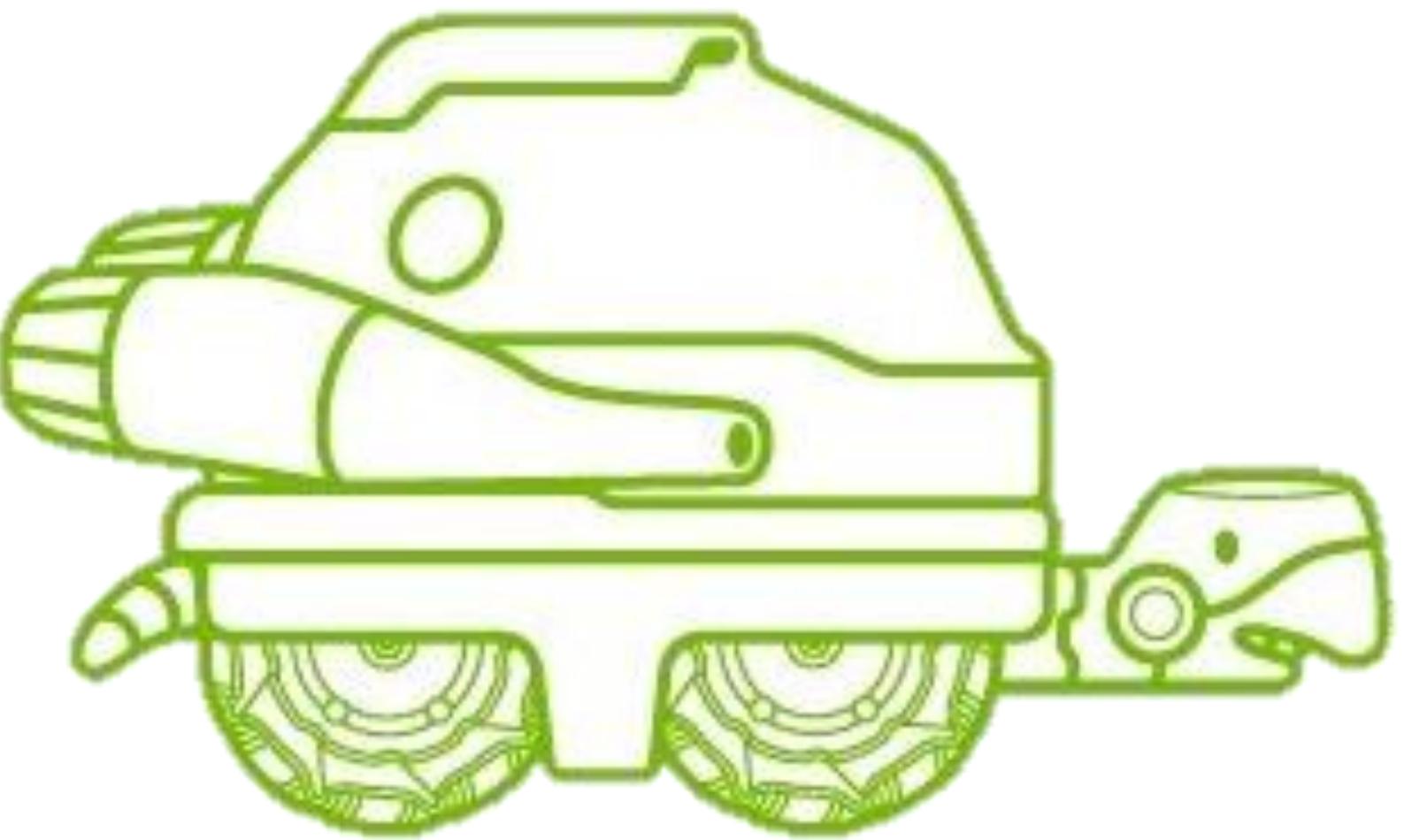


nVIDIA®



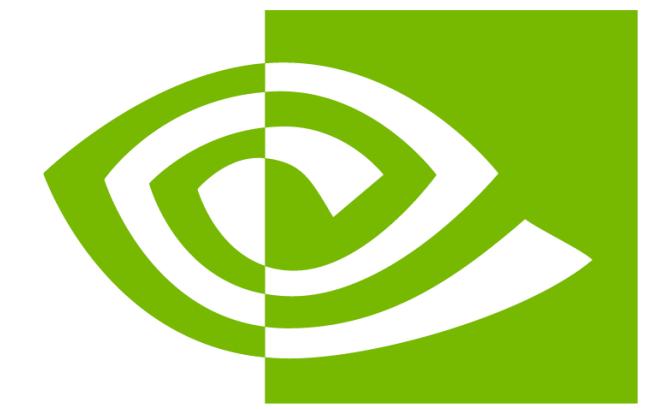
⋮ 2

ISAAC ROS HERO?





Isaac ROS Hero?

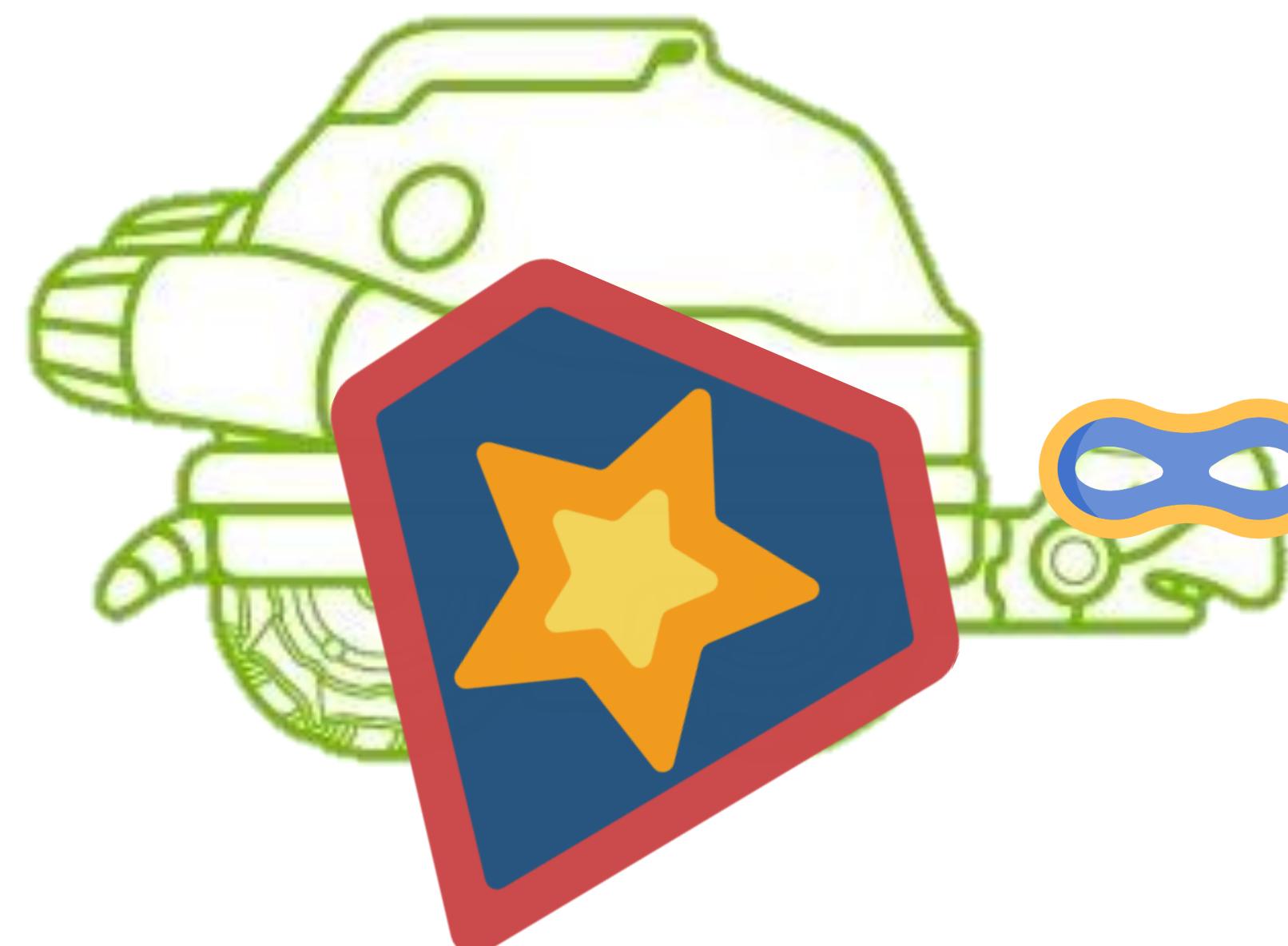


nVIDIA®



⋮ 2

With DevOps



Write my Isaac ROS workspace

The faster and cool way

my_isaac_ros.rosinstall

```
- git:  
  local-name: isaac_ros_common  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_common  
- git:  
  local-name: isaac_ros_nitros  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_nitros  
- git:  
  local-name: isaac_ros_image_pipeline  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_image_pipeline  
- git:  
  local-name: isaac_ros_apriltag  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_apriltag  
- git:  
  local-name: isaac_ros_visual_slam  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_visual_slam
```

```
FROM /opt/ros/humble  
  ros.rosinstall  
  rosinstall /my_isaac_ros.rosinstall  
    Isaac ROS packages  
    /src \
```

```
  ws/src < /my_isaac_ros.rosinstall \  
  rosinstall \  
    using rosdep  
    RO/install/setup.sh \  
    
```

```
  {  
    lists/* \  
  }
```

```
  clean resources  
  ge-install --cmake-args -DCMAKE_BUILD_TYPE=Release \  
    and log
```

```
  [ "my.launch", "my_cool_package", "isaac_ros.forever.launch.py" ]
```

REWIND

NVIDIA Isaac ROS

NVIDIA AI Perception for the ROS community

AI modules that can plug into the ROS framework and get accelerated performance immediately



Optimize
ROS Framework

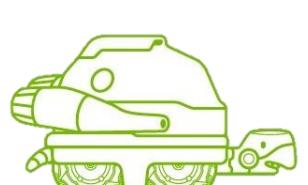
Accelerate
Hardware
Performance

2

850,000+
Developers

Scale with
AI Algorithms

750,000+
Downloads



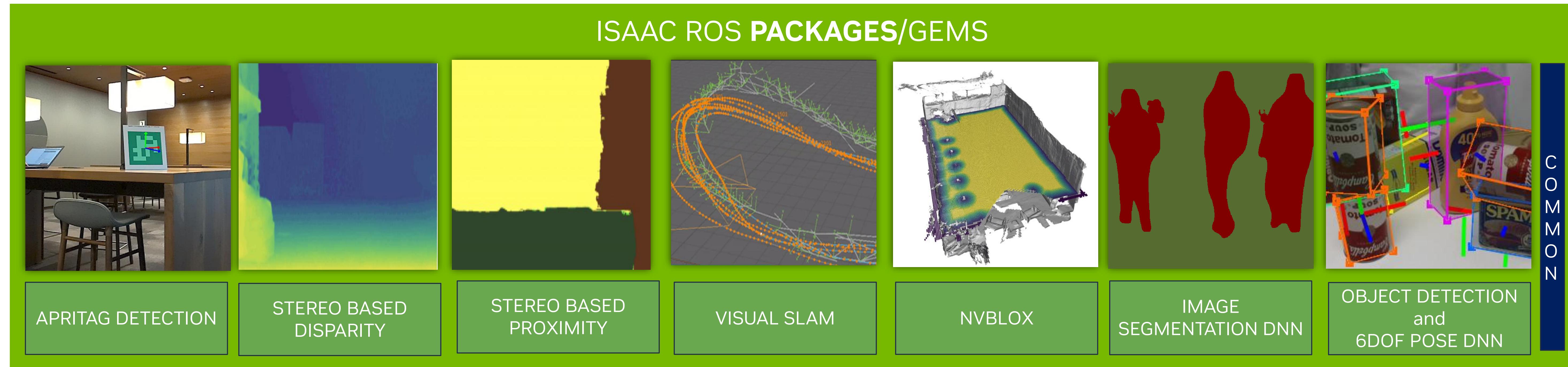
NVIDIA Isaac ROS GEMs

Hardware Accelerated Packages for ROS2

ROS native packages for bringing AI, CV and camera sensors to manipulation and navigation.

Support for large collection of pre-trained models from NGC

Seamless integration with open source/custom ROS tools and packages.



Hardware accelerated libraries and engines enable highest performance and efficient resource utilization.

Triton Server

NvEngine

ROS Middleware

TensorRT

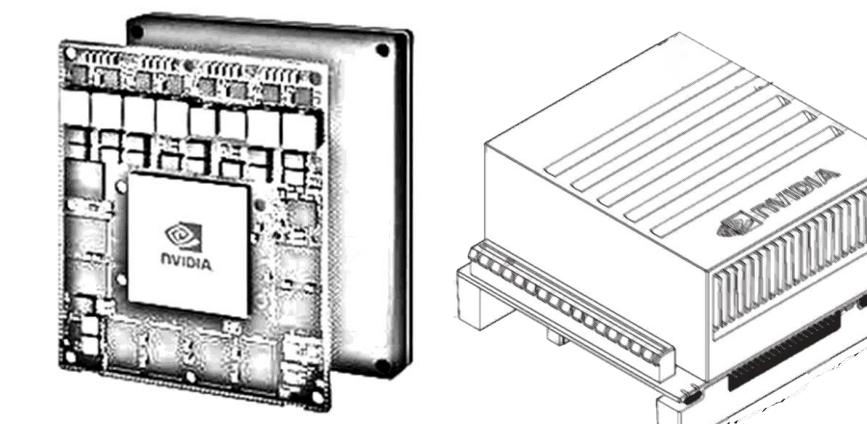
HARDWARE ACCELERATED LIBRARIES

VPI

CUDA

ARGUS

Optimized to run on all NVIDIA compute platform.



JETSON COMPUTER



GPU WORKSTATION



GPU SERVER

NVIDIA Isaac ROS

Roadmap

2021 - Aug.

Isaac ROS EA1

- Image Pipeline
- Apriltag
- Stereo Depth

2022 - Mar.

Isaac ROS EA3

- Visual SLAM
- NVBLOX
- Object Detection

2022 - Oct.

Isaac ROS DP2

- Visual Nav2 Pipeline
- Multi-modal odometry
- Mono SVIO
- P-STOP
- H.264 HW enc & dec
- Mission Dispatch & Client

2021 - Oct.

Isaac ROS EA2

- Visual Odometry
- Argus Camera
- Image Segmentation
- Pose Estimation

2022 - Jun

Isaac ROS DP1

- Jetson Orin Support
- NITROS (up to 7x perf)
- ESS DNN Stereo Disparity
- Bi3D Proximity Segmentation

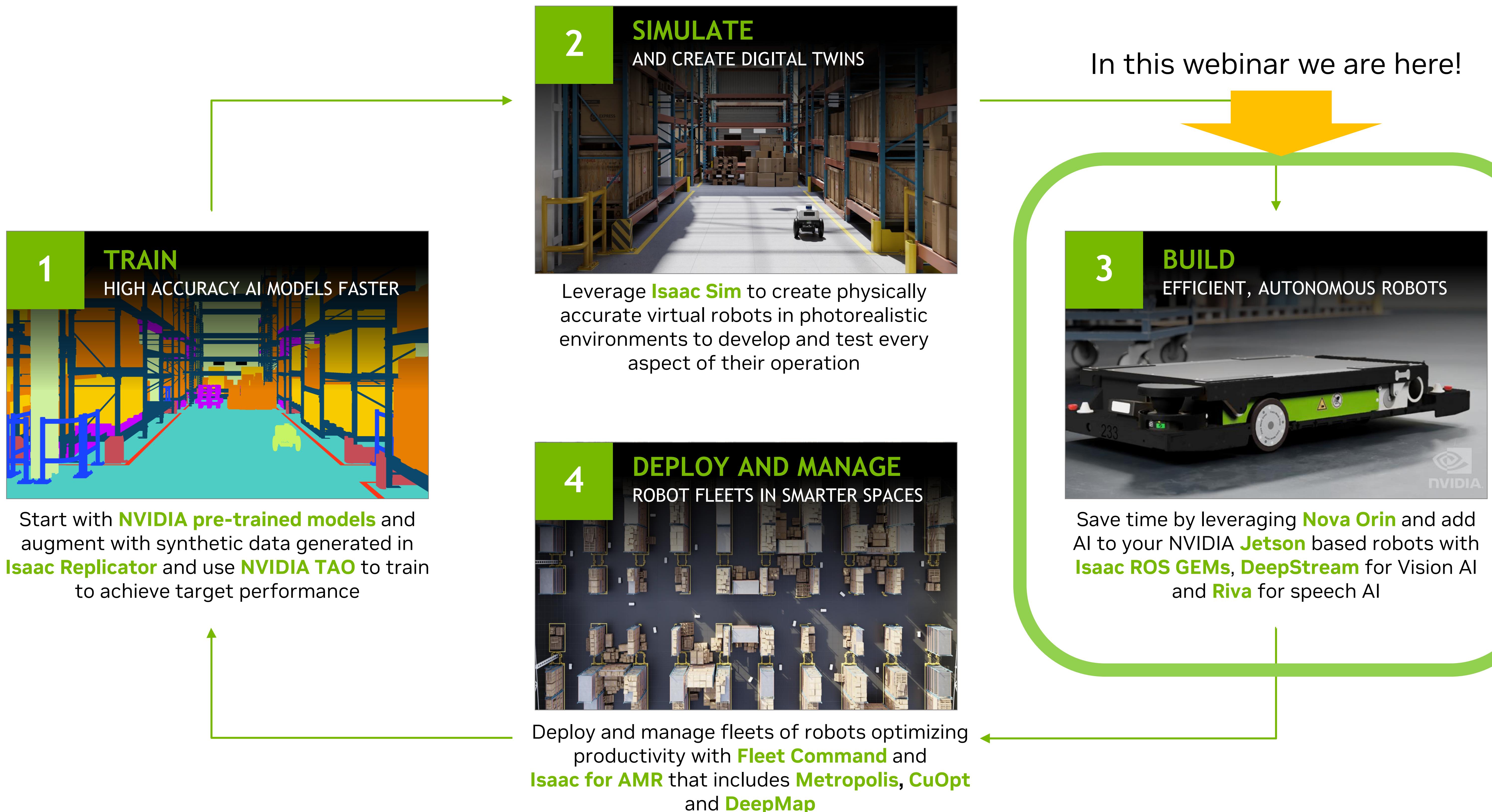
2023 - Mar.

Isaac ROS DP3

- New features
- Bug fixes

End to End Robotics with NVIDIA Isaac

Smarter Robots Developed Faster Leveraging NVIDIA AI and Omniverse





Disclaimer

ROS2 minimal experience
Linux



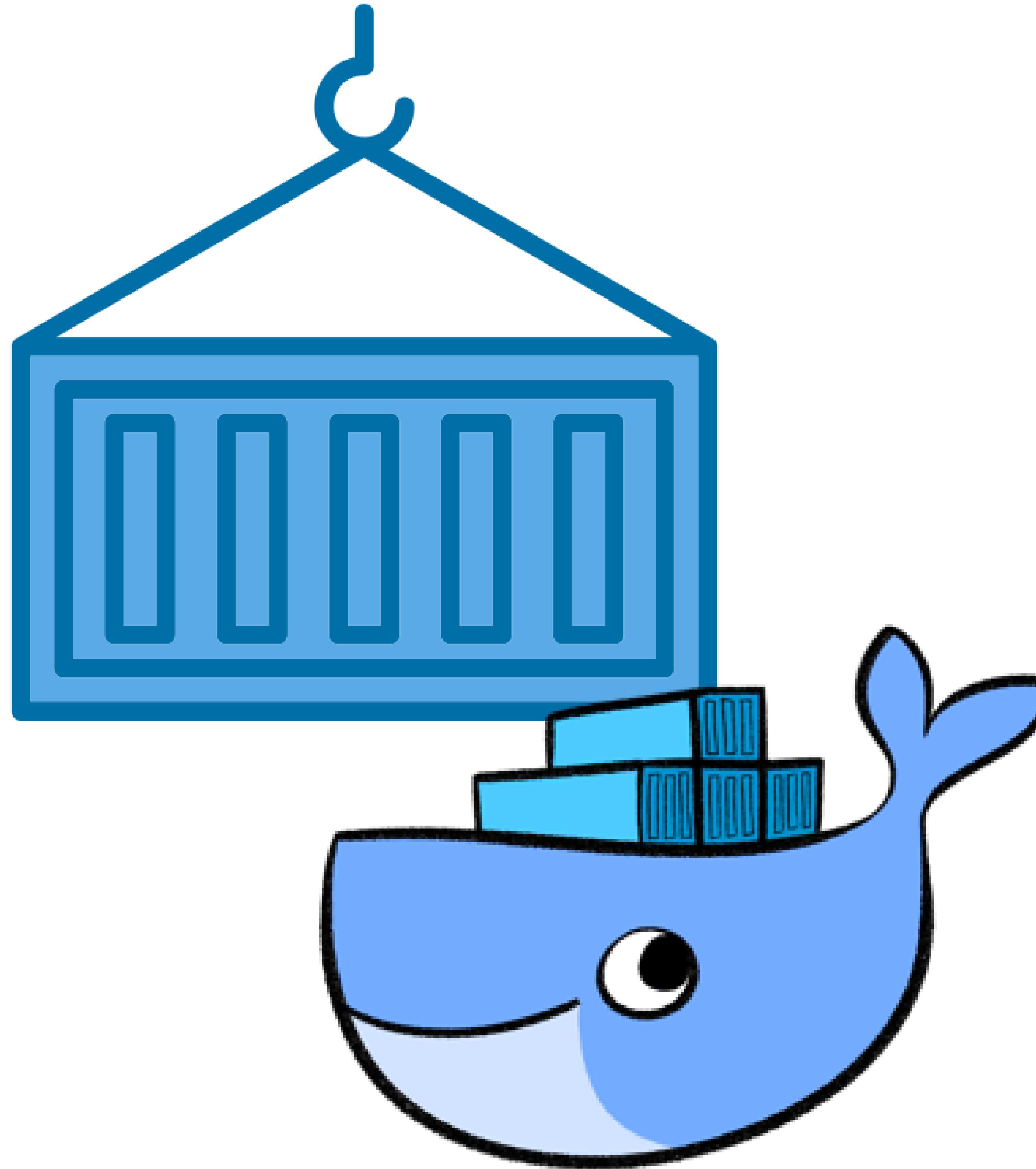
Agenda



- Introduction
- Docker/NVIDIA Docker
- Isaac ROS common
- Build my Docker
- Recap

What is docker?

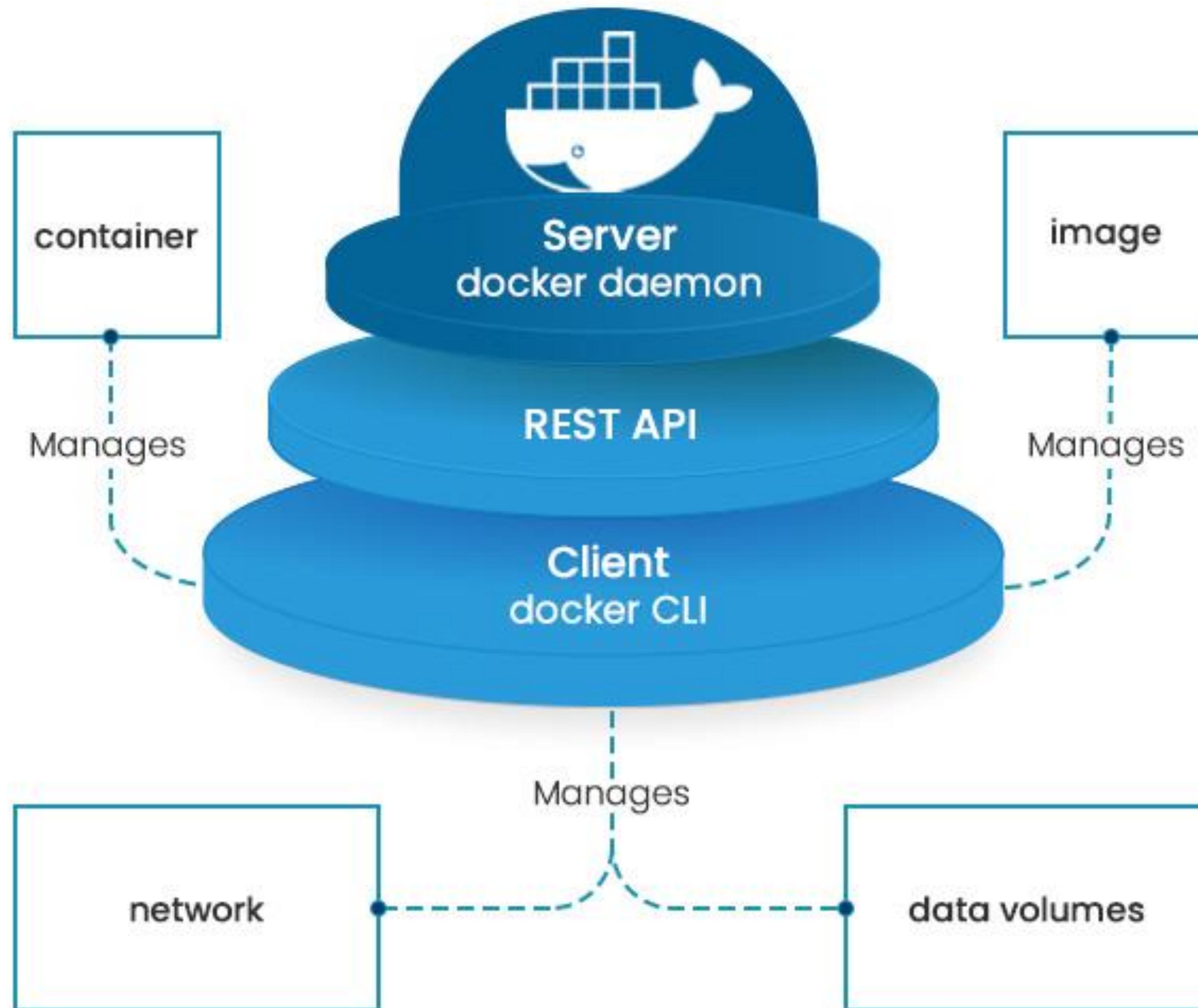
Docker makes development efficient and predictable



- Docker is a container platform
- Containers package apps and dependencies
- Ensure consistent behaviour across environments
- Speed up app development, testing, and deployment
- Facilitate app management and scaling.

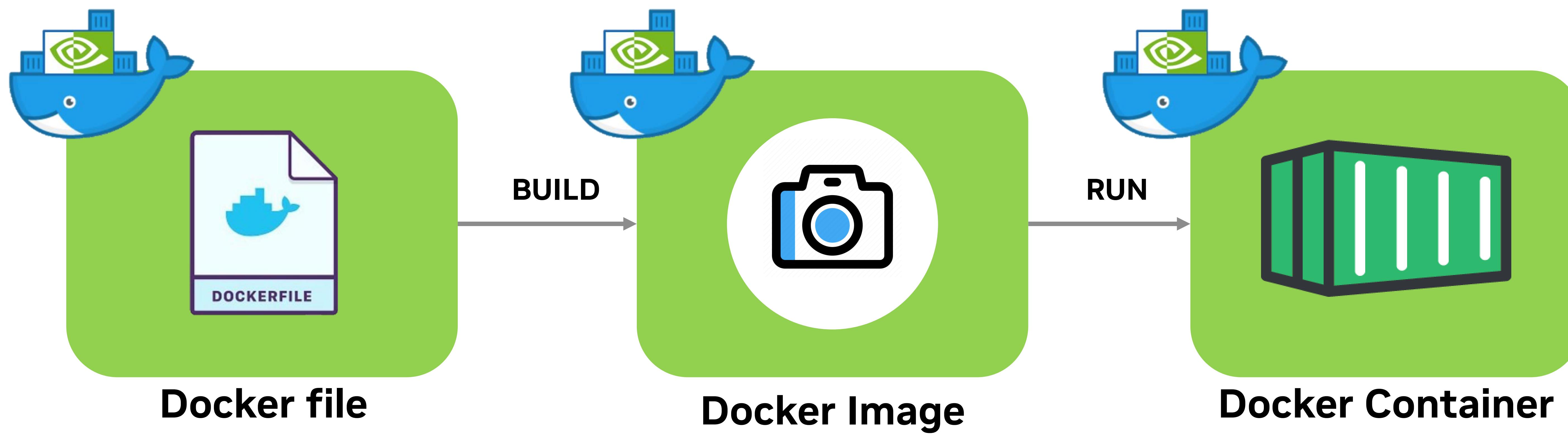
Docker engine

What there is behind

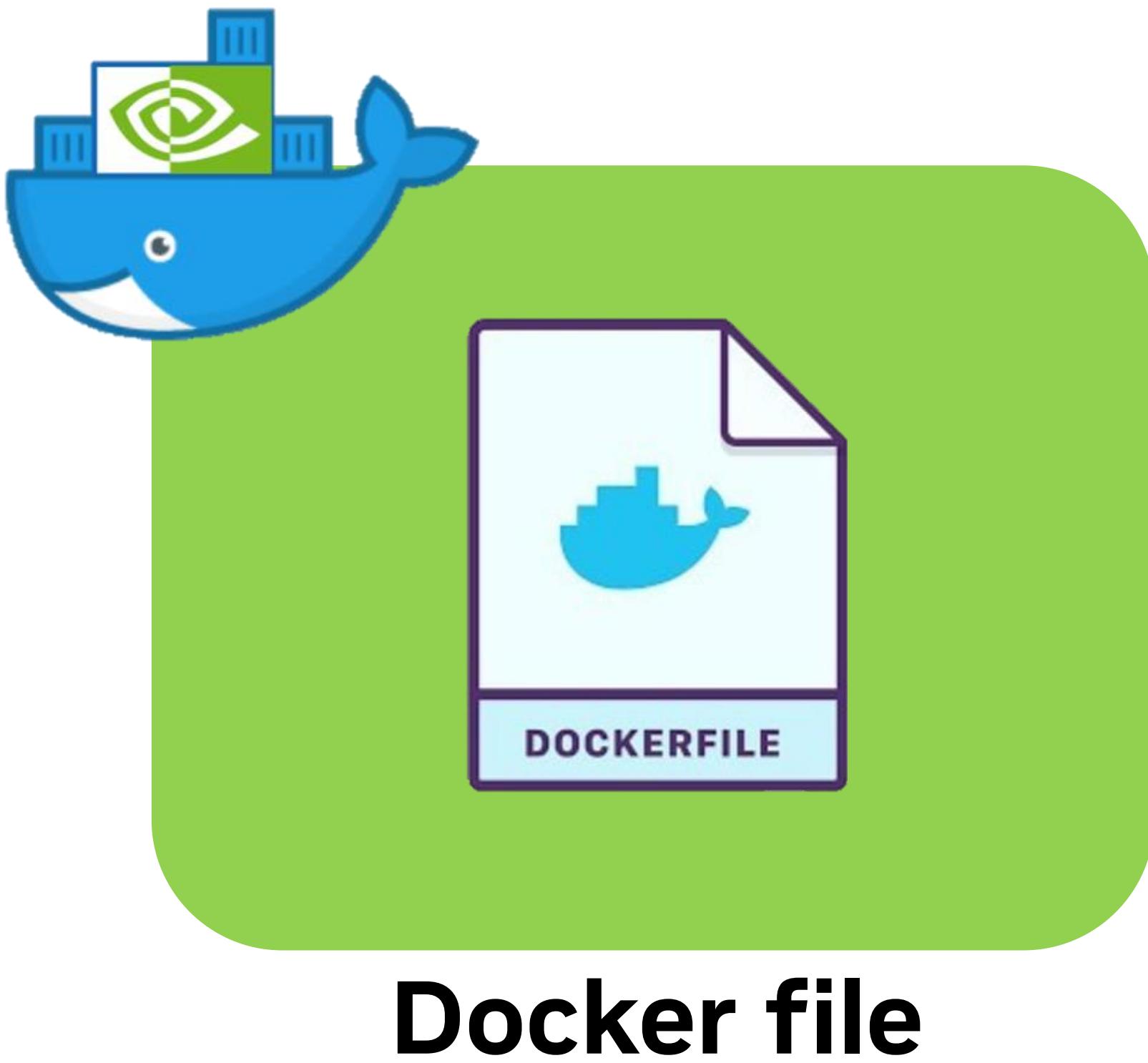


- The Docker Engine is the runtime that allows you to build, run, and manage containers. It takes care of the low-level details of container creation, networking, and storage, allowing you to focus on writing and deploying your applications.
- The Docker Engine can run on a variety of platforms, including laptops, servers, and cloud environments, making it a flexible and portable solution for deploying and running containers.

Docker file, Docker Image and Docker Container



What is a Dockerfile



- Dockerfile is a script for building Docker images.
- Specifies base image, application code, and dependencies.
- Written in simple syntax, each line represents a command.
- Can be built using docker build command.
- Resulting image can be run as a container using docker run.
- Helps automate image building and provides version control.
- Easy to share images and deploy same image consistently.

How to run a Docker image

1. Pull the database image from a Docker repository:

```
docker pull database-image:version
```

2. Start a container from the database image:

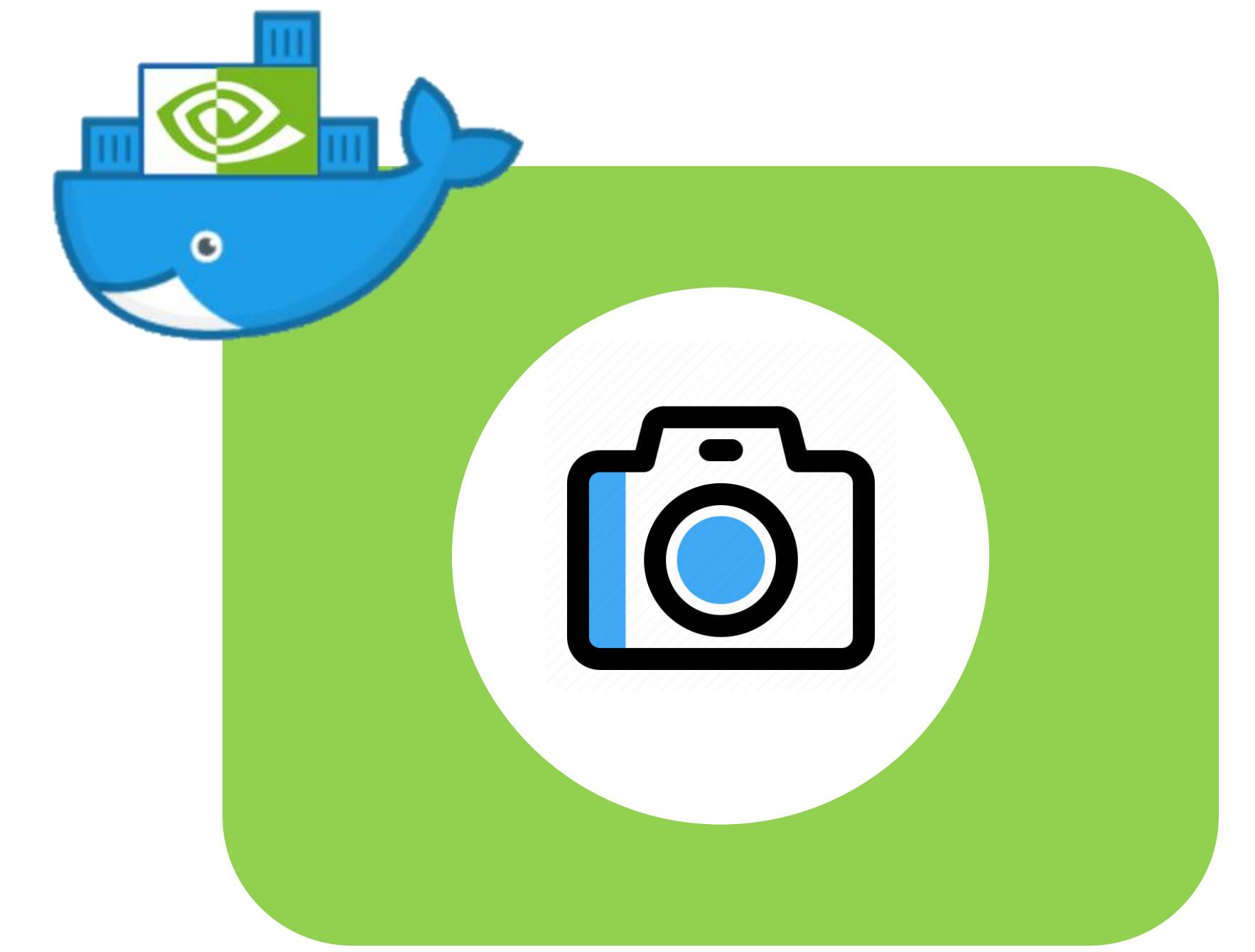
```
docker run --name database -d database-image:version
```

3. Pull the web application image:

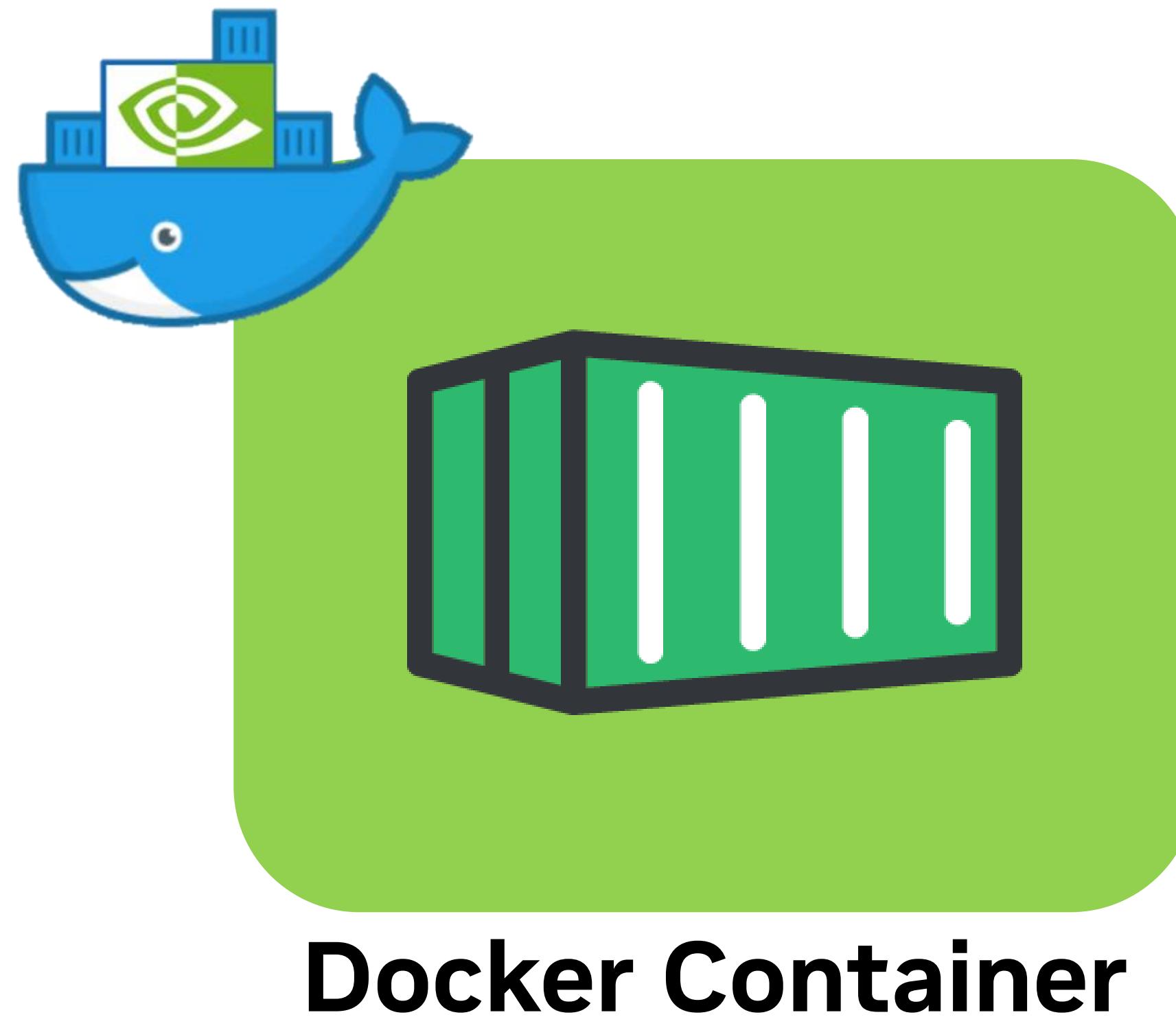
```
docker pull web-app-image:version
```

4. Start a container from the web application image and link it to the database container:

```
docker run --name web-app --link database:db -d web-app-image:version
```

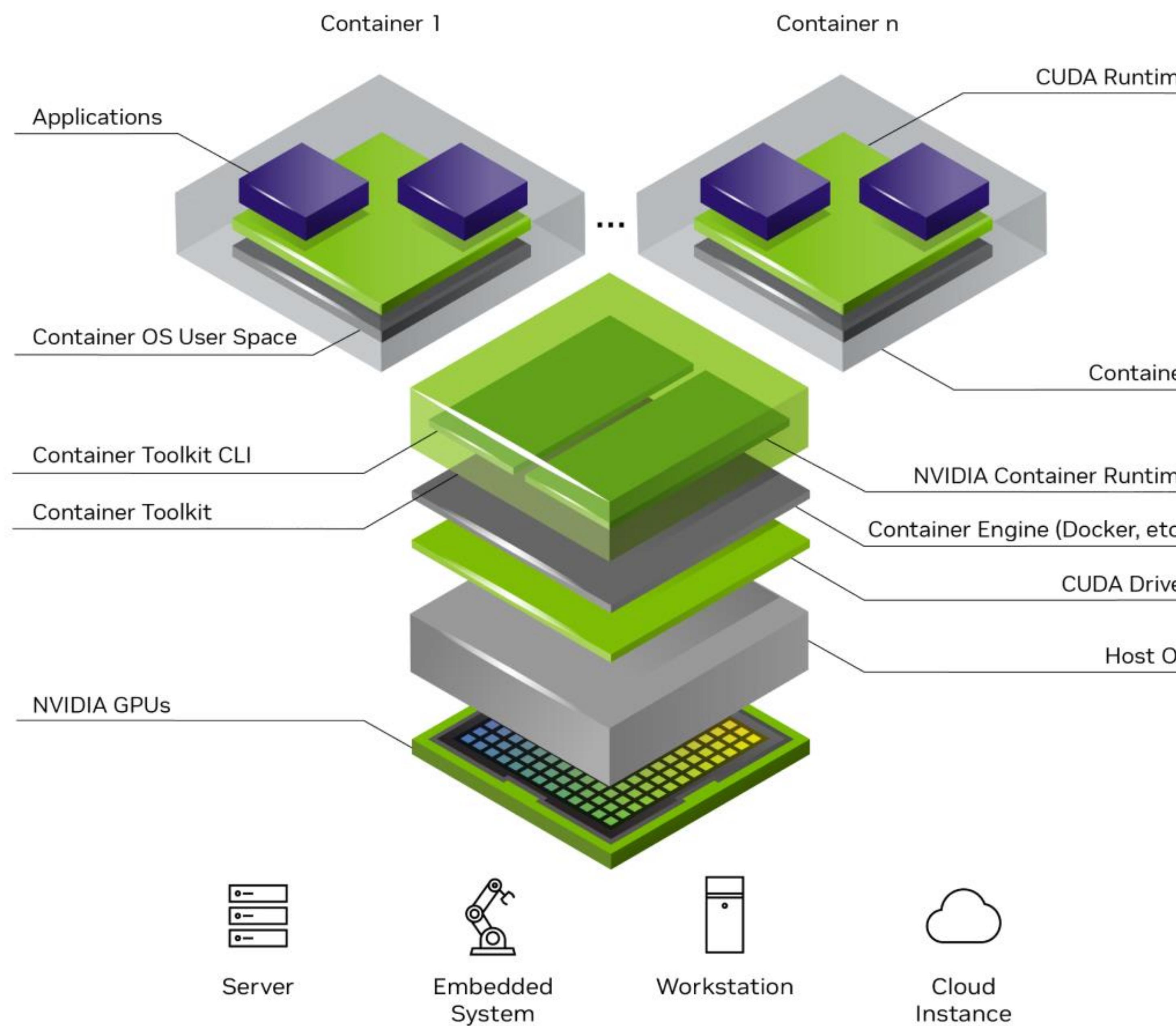


Docker container



- Docker containers are standalone executable packages.
- Include code, runtime, libraries, environment variables, and config files.
- Isolated and provide predictable runtime environment.
- Created from Docker images and run on the Docker Engine.
- Lightweight, efficient, and allow easy deployment of applications.
- Ensure consistent behavior across environments.

The NVIDIA Container Toolkit



- The NVIDIA Container Toolkit **enables** the use of NVIDIA GPUs in a variety of container engines including Docker and Podman.
 - The NVIDIA Container Toolkit takes care of **making the local GPUs and drivers available to containers**. This allows container images for GPU-accelerated applications to be handled in the same way as any other type of application making it easier to develop, test, and deploy GPU-accelerated applications in a **consistent and portable way**.

Example: NVIDIA Dockerfile

My NVIDIA/CUDA base image

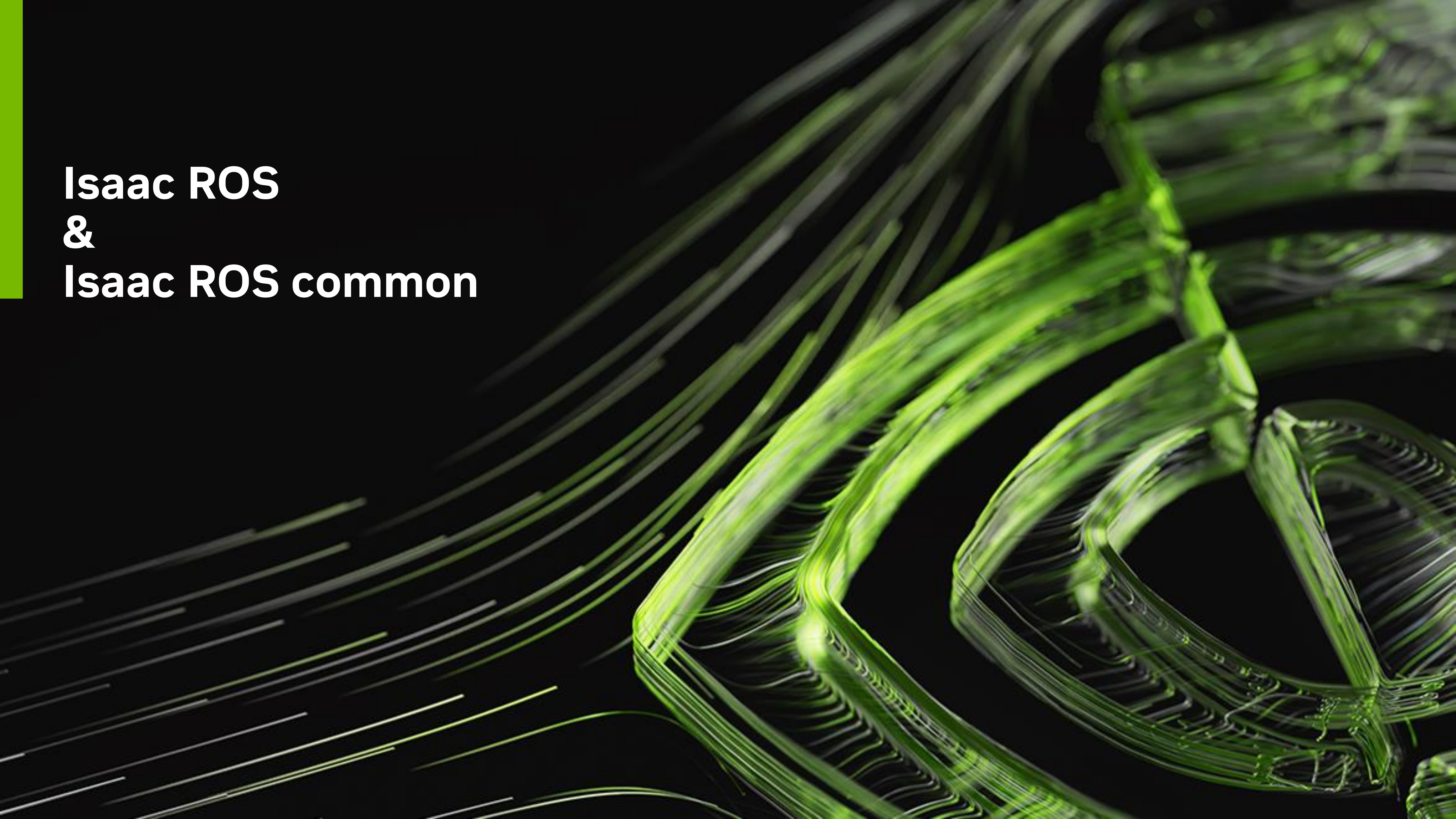
Where I install and run my code

Where I install all my dependecies

Where my code start when I start a container

```
1 # Use a CUDA-enabled base image
2 FROM nvidia/cuda:11.4-base-ubuntu20.04
3
4 # Install the required dependencies
5 RUN apt-get update && apt-get install -y \
6     python3 \
7     python3-pip
8
9 # Copy the application code into the image
10 COPY . /app
11
12 # Set the working directory
13 WORKDIR /app
14
15 # Install the application's dependencies
16 RUN pip3 install -r requirements.txt
17
18 # Set the entrypoint command
19 ENTRYPOINT ["python3", "main.py"]
```

Isaac ROS
&
Isaac ROS common



NVIDIA Isaac ROS Resources

<https://developer.nvidia.com/isaac-ros-gems>

NVIDIA Isaac ROS GEMs

The NVIDIA® Isaac ROS GEMs are **hardware accelerated packages** that make it easier for **ROS developers** to build high-performance solutions on NVIDIA hardware.

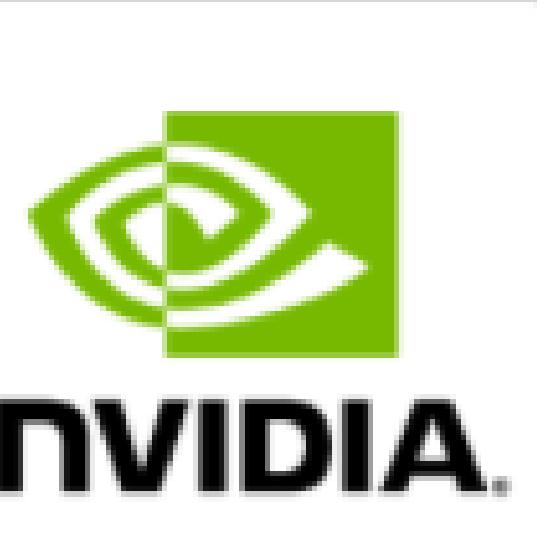
GET STARTED



Composite Image from 3 Isaac ROS GEMs -- DNN (PeopleSemSegnet)/AprilTags/Disparity(Depth)

NVIDIA Isaac ROS Resources

<https://github.com/NVIDIA-ISAAC-ROS>



NVIDIA Isaac ROS

High-performance computing for robotics

📍 United States of America [🔗 https://developer.nvidia.com/isaac-ros...](https://developer.nvidia.com/isaac-ros)

🏠 Overview 🚦 Repositories 8 🏷 Packages 🔎 People 📁 Projects

Pinned

💻 [isaac_ros_apriltag](#) Public

CUDA-accelerated Apriltag detection and pose estimation.

● C++ ⭐ 21 🏷 3

💻 [isaac_ros_visual_odometry](#) Public

Visual odometry package based on hardware-accelerated NVIDIA Elbrus library with world class quality and performance.

● C++ ⭐ 88 🏷 7

💻 [isaac_ros_image_pipeline](#) Public

Hardware-accelerated ROS2 packages for camera image processing.

● Python ⭐ 19 🏷 2

💻 [isaac_ros_argus_camera](#) Public

ROS2 packages based on NVIDIA libArgus library for hardware-accelerated CSI camera support.

● C++ ⭐ 15 🏷 3

💻 [isaac_ros_image_segmentation](#) Public

Hardware-accelerated, deep learned semantic image segmentation

● C++ ⭐ 6

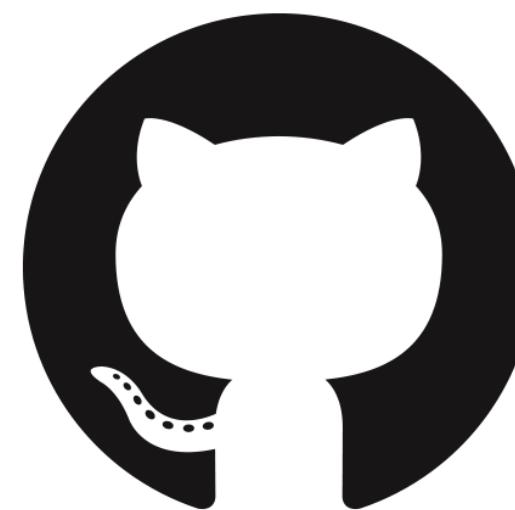
💻 [isaac_ros_pose_estimation](#) Public

Deep learned, hardware-accelerated 3D object pose estimation

● Python ⭐ 16 🏷 3

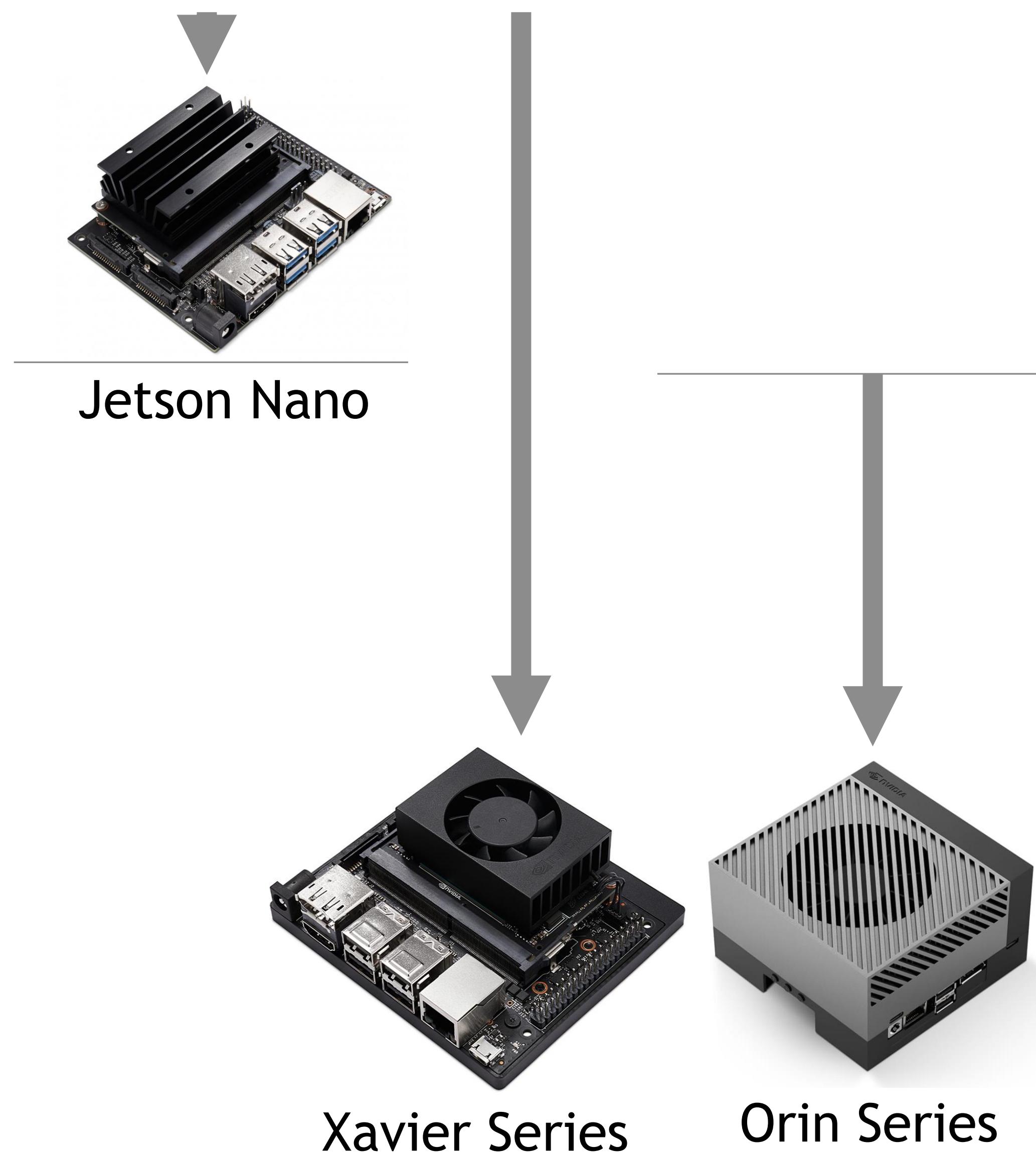
Isaac ROS versions

Some hints when you work with Isaac ROS



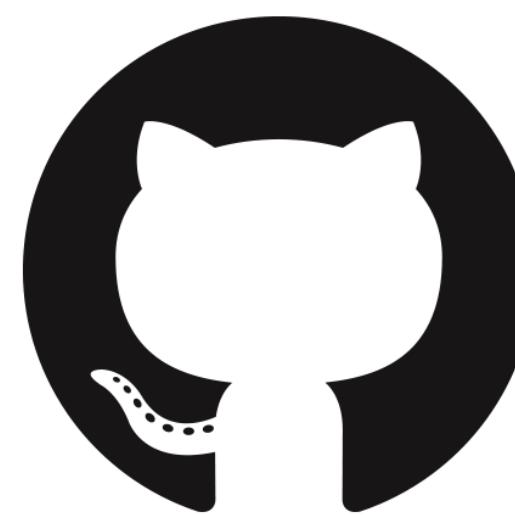
<https://github.com/NVIDIA-ISAAC-ROS/.github/blob/main/profile/release-notes.md>

Version	Release date	NVIDIA Jetpack	ROS
EA1	Aug 11, 2021	4.5	Foxy
EA2	Oct 20, 2021	4.6	Foxy
0.9.1 EA2.1	Nov 22, 2021	4.6	Foxy
0.93 EA3	Mar 23, 2022	4.6.1	Foxy
0.10.0 DP	June 30, 2022	5.0.1 DP	Humble
0.10.1 DP	July 12, 2022	5.0.1 DP	Humble
0.11.0 DP	Sep 1, 2022	5.0.2	Humble
0.20.0 DP	Oct 19, 2022	5.0.2	Humble



NVIDIA Isaac ROS common

How it is work Isaac ROS Common



https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_common

NVIDIA-ISAAC-ROS / isaac_ros_common Public Watch 3 Fork 35 Starred 68

Code Issues 3 Pull requests 1 Actions Security Insights

main Go to file Add file Code About

hemalshahNV Merge pull request #54 from NVIDIA-ISAAC-ROS/h... last week 33

docker Revert Isaac ROS Dev base image for aarch64 last week

docs Update troubleshooting to address common G... 2 weeks ago

isaac_ros_apriltag_inte... Isaac ROS 0.20.0 (DP2) 4 months ago

isaac_ros_biol3d_interfa... Isaac ROS 0.20.0 (DP2) 4 months ago

isaac_ros_common Isaac ROS 0.20.0 (DP2) 4 months ago

isaac_ros_tensor_list_i... Isaac ROS 0.20.0 (DP2) 4 months ago

isaac_ros_test Isaac ROS 0.20.0 (DP2) 4 months ago

resources Isaac ROS 0.10.0 (DP) 8 months ago

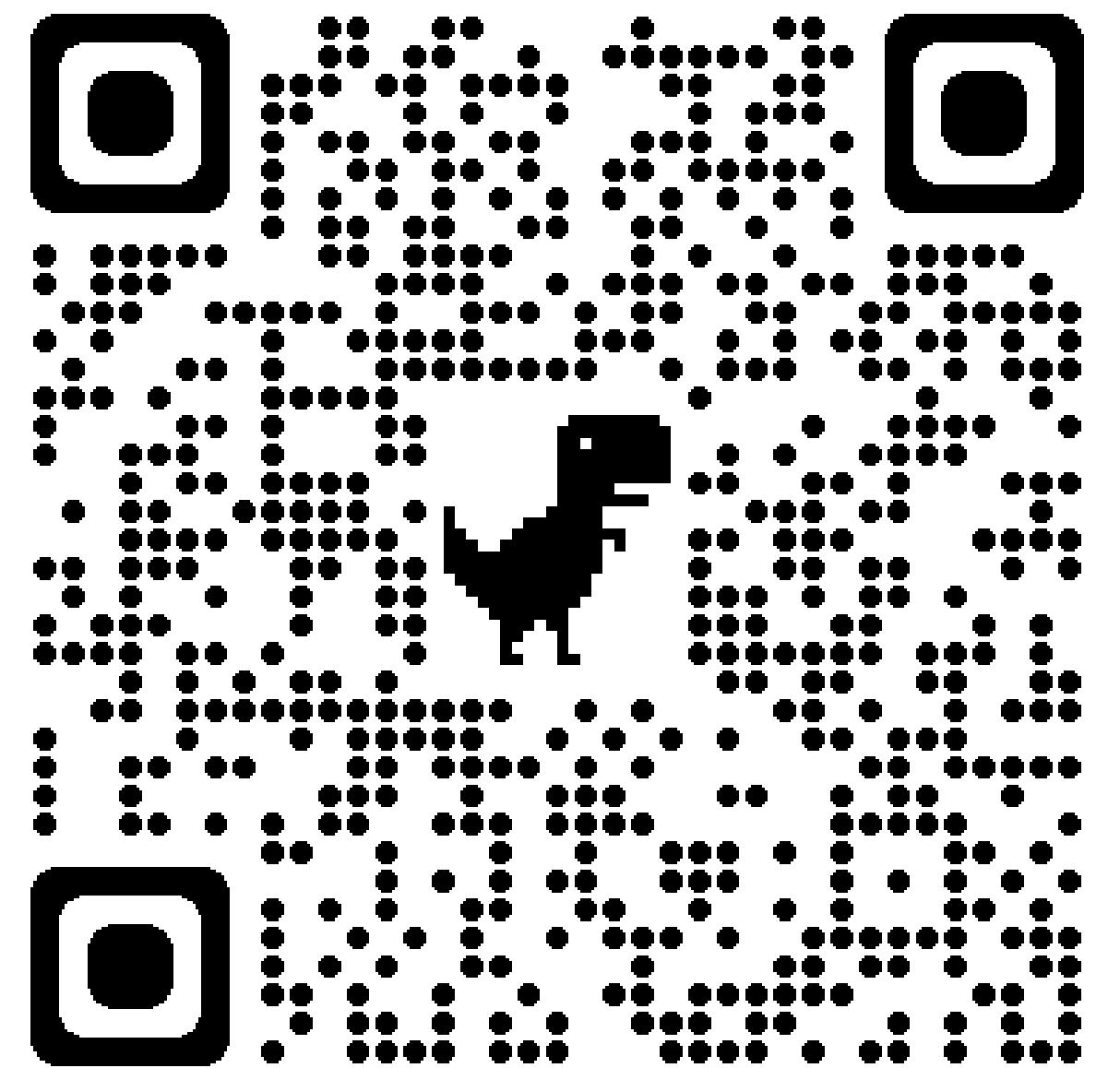
scripts Isaac ROS 0.20.0 (DP2) 4 months ago

Common utilities, packages, scripts, Dockerfiles, and testing infrastructure for Isaac ROS packages.

developer.nvidia.com/isaac-ros-ge...

docker ubuntu gpu ros nvidia jetson ros2 ros2-humble

Readme View license 68 stars 3 watching 35 forks



Isaac ROS

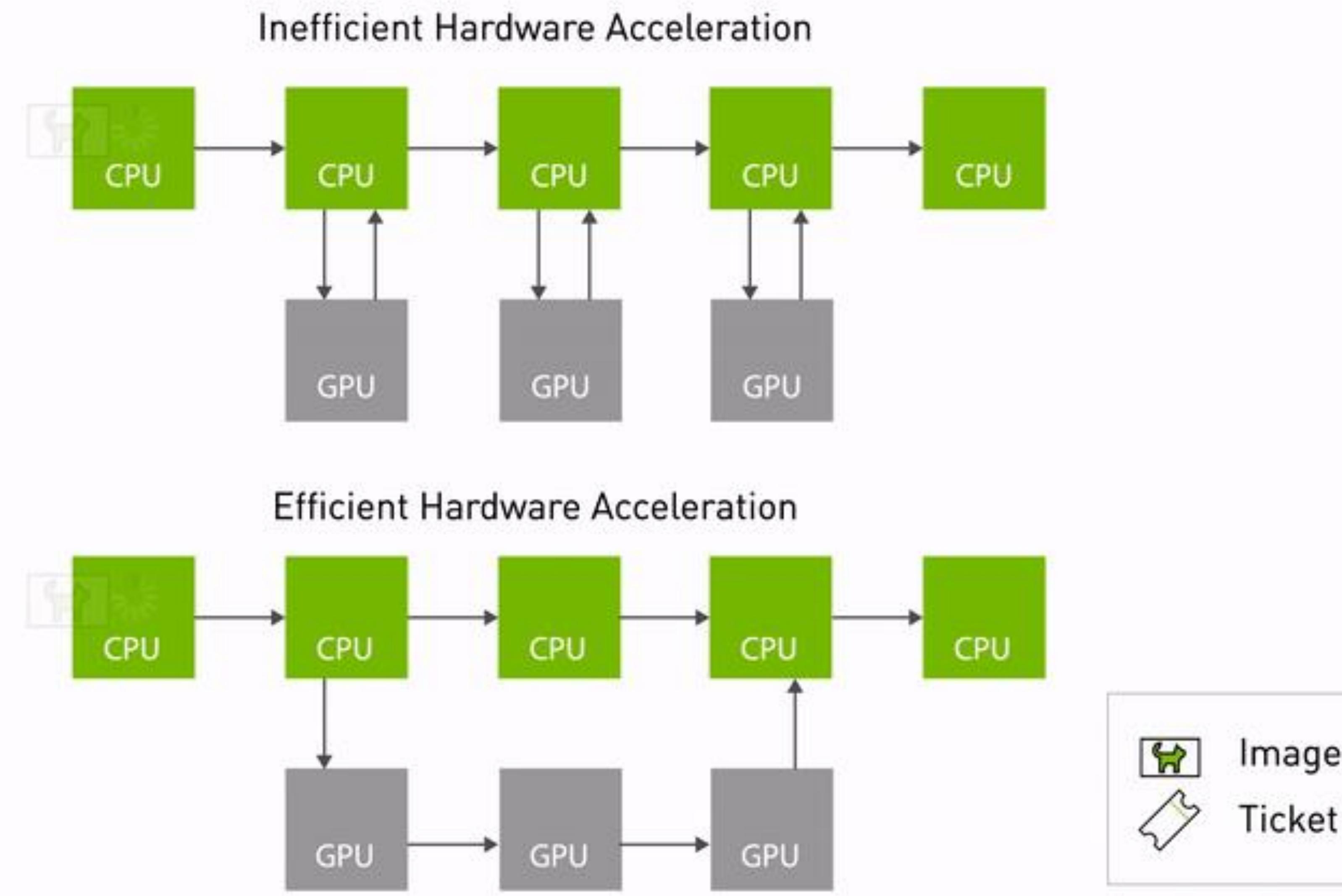
Specifications for 0.20.0 DP

	Desktop	Jetson
Machine	X86/64 with discrete NVIDIA	NVIDIA Jetson Orin/Xavier
Software	Ubuntu 20.04	Jetpack 5.0.2 (Ubuntu 20.04)
ROS2		 Humble
CUDA		11.4
VPI		2.16
Other libraries		Tensorrt 8.4
		Tritonserver 2.24
		Tao-converter
		mqtt

* = optional

NVIDIA Isaac ROS for Transport (NITROS)

Hardware Acceleration for Graphs of Nodes



2.8X
faster pixel
processing* on AGX
Xavier

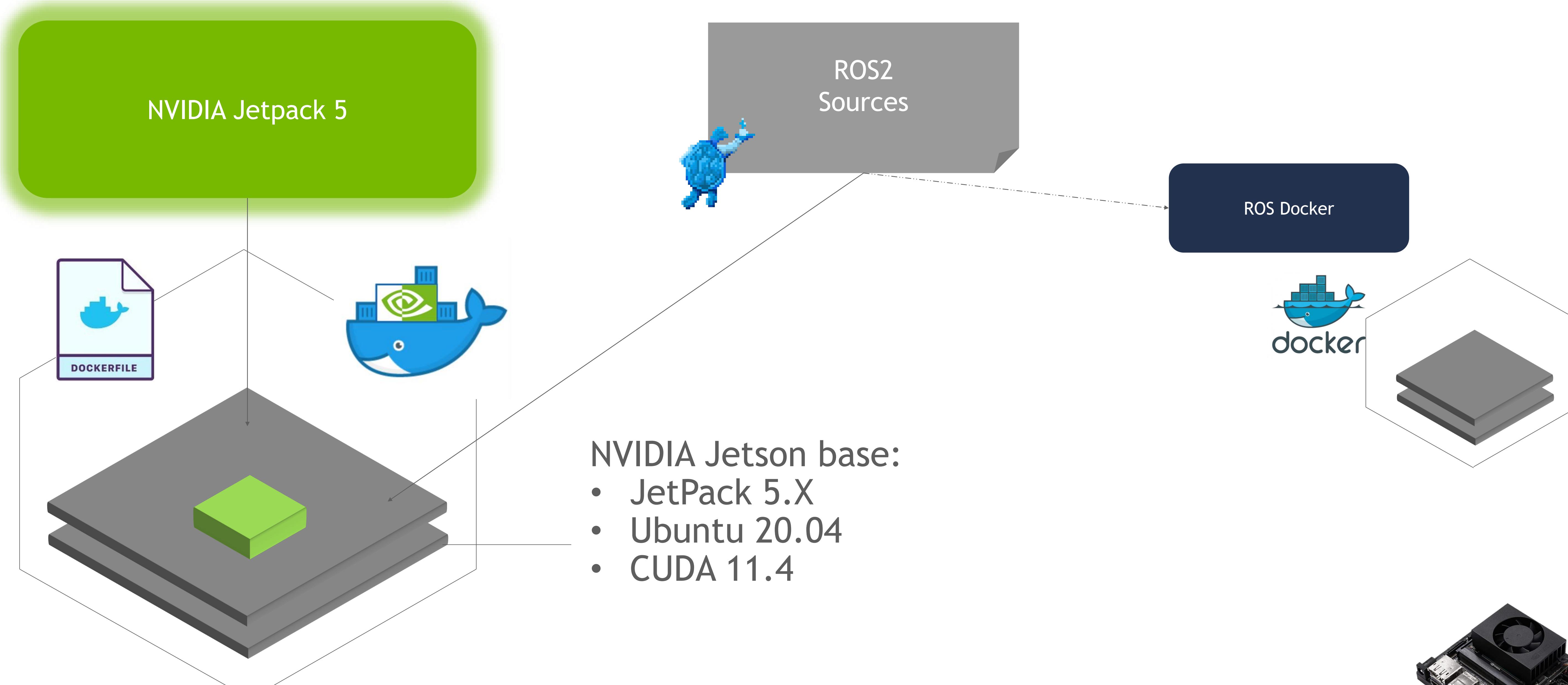
7X
faster pixel processing'
on AGX Orin

* ROS2 node graph operating in sequence on 1080p CUDA buffers in Foxy vs the same node graph in Humble with Type Adaptation; results measured in Hz on Jetpack 5.0 developer preview, Ubuntu 20.04 with Jetson AGX Orin and Xavier. Graph of nodes is designed to test framework performance by minimizing compute workload, bringing focus to overhead in the ROS Client Library

ROS on NVIDIA Jetson

How ROS2 container is built on NVIDIA Jetson

GPU accelerated



Both docker distributions runs on NVIDIA Jetson



Agenda

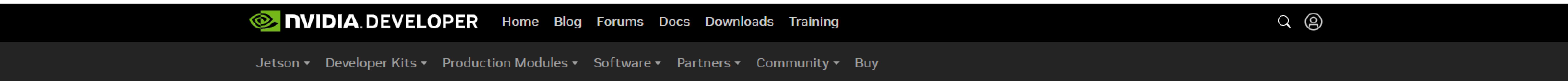
- ✓ Introduction
- ✓ Docker/NVIDIA Docker
- ✓ Isaac ROS common
 - Build my Docker
 - Recap



Build my Isaac ROS docker

NVIDIA Jetpack 5

<https://developer.nvidia.com/embedded/jetpack>



The image shows the top navigation bar of the NVIDIA Developer website. It features the NVIDIA logo and the word "DEVELOPER" in white on a black background. To the right are links for "Home", "Blog", "Forums", "Docs", "Downloads", and "Training". On the far right are a search icon and a user profile icon. Below this is a secondary navigation bar with dropdown menus for "Jetson", "Developer Kits", "Production Modules", "Software", "Partners", "Community", and "Buy".

[Home](#) > [Autonomous Machines](#) > [Develop](#) > [Tools](#) > [JetPack SDK](#)

JetPack SDK

NVIDIA JetPack SDK is the most comprehensive solution for building end-to-end accelerated AI applications. JetPack provides a full development environment for hardware-accelerated AI-at-the-edge development on Nvidia Jetson modules.

JetPack includes [Jetson Linux](#) with bootloader, Linux kernel, Ubuntu desktop environment, and a complete set of libraries for acceleration of GPU computing, multimedia, graphics, and computer vision. It also includes samples, documentation, and developer tools for both host computer and developer kit, and supports higher level SDKs such as DeepStream for streaming video analytics, Isaac for robotics, and Riva for conversational AI.

JetPack 5.1

JetPack 5.1 is a production quality release and brings support for Jetson Orin NX 16GB module. It includes [Jetson Linux 35.2.1 BSP](#) with Linux Kernel 5.10, an Ubuntu 20.04 based root file system, a UEFI based bootloader, and OP-TEE as Trusted Execution Environment. JetPack 5.1 packages CUDA 11.4, TensorRT 8.5.2, cuDNN 8.6.0 and VPI 2.2, along with other updates. See highlights below for the full list of features.

This release supports Jetson AGX Orin 32 GB production module, Jetson AGX Orin Developer Kit, Jetson Orin NX 16GB production module and also supports Jetson AGX Xavier series and Jetson Xavier NX series modules, as well as Jetson AGX Xavier Developer Kit and Jetson Xavier NX Developer Kit.

NVIDIA Jetpack 5

CUDA

¹Signing is supported. Encryption will be supported in the next release.

²Jetson Linux 35.2.1 will not require any change to use the OTA tools released in the next release. Jetson Orin based products running Jetson Linux 35.2.1 should be able to upgrade using the OTA tools released in the next release.

TensorRT

TensorRT is a high performance deep learning inference runtime for image classification, segmentation, and object detection neural networks. TensorRT is built on CUDA, NVIDIA's parallel programming model, and enables you to optimize inference for all deep learning frameworks. It includes a deep learning inference optimizer and runtime that delivers low latency and high-throughput for deep learning inference applications.

JetPack 5.1 includes TensorRT 8.5.2

cuDNN

CUDA Deep Neural Network library provides high-performance primitives for deep learning frameworks. It provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers.

JetPack 5.1 includes cuDNN 8.6.0

CUDA

CUDA Toolkit provides a comprehensive development environment for C and C++ developers building GPU-accelerated applications. The toolkit includes a compiler for NVIDIA GPUs, math libraries, and tools for debugging and optimizing the performance of your applications.

JetPack 5.1 includes CUDA 11.4.15

Starting with JetPack 5.0.2, upgrade to latest and greatest CUDA releases from CUDA 11.8 onwards without the need to update Jetson Linux other JetPack components. Refer to instructions in the **CUDA documentation** on how to get the latest CUDA on JetPack.

Multimedia API

The **Jetson Multimedia API** package provides low level APIs for flexible application development.

Camera application API: libargus offers a low-level frame-synchronous API for camera applications, with per frame camera parameter control, multiple (including synchronized) camera support, and EGL stream outputs. RAW output CSI cameras needing ISP can be used with either libargus or GStreamer plugin. In either case, the V4L2 media-controller sensor driver API is used.

Sensor driver API: V4L2 API enables video decode, encode, format conversion and scaling functionality. V4L2 for encode opens up many features like bit rate control, quality presets, low latency encode, temporal tradeoff, motion vector maps, and more.

JetPack 5.1 Camera highlights include:

- Support for AR1335 YUV camera
- Enhanced support for simultaneous V4L2 and Argus

NVIDIA docker - CUDA

How is made nvidia/cuda docker image

<https://catalog.ngc.nvidia.com/orgs/nvidia/containers/cuda>

NVIDIA NGC | CATALOG Welcome Guest

Catalog > Containers > CUDA

CUDA

[Copy Image Path](#)

[Overview](#) Tags Layers Security Scanning Related Collections

NVIDIA CUDA

CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units (GPUs). With CUDA, developers can dramatically speed up computing applications by harnessing the power of GPUs.

The CUDA Toolkit from NVIDIA provides everything you need to develop GPU-accelerated applications. The CUDA Toolkit includes GPU-accelerated libraries, a compiler, development tools and the CUDA runtime.

The CUDA container images provide an easy-to-use distribution for CUDA supported platforms and architectures.

End User License Agreements

The images are governed by the following NVIDIA End User License Agreements. By pulling and using the CUDA images, you accept the terms and conditions of these licenses. Since the images may include components licensed under open-source licenses such as GPL, the sources for these components are archived [here](#).

NVIDIA Deep learning Container License

To view the NVIDIA Deep Learning Container license, click [here](#)

Documentation

For more information on CUDA, including the release notes, programming model, APIs and developer tools, visit the [CUDA documentation site](#).

Announcement

Cuda 12 images are now LIVE

Current version of CUDA is 12.0.1

Entrypoint scripts are added to all images, which include deprecation notices for image sets that have reached End-of-life.

Please read our [Container Tag Support Policy](#) for more information.

Description
CUDA is a parallel computing platform and programming model that enables dramatic increases in computing performance by harnessing the power of the NVIDIA GPUs.

Publisher
NVIDIA

Latest Tag
11.8.0-devel-ubuntu22.04

Modified
February 2, 2023

Compressed Size
2.57 GB

Multinode Support
No

Multi-Arch Support
Yes

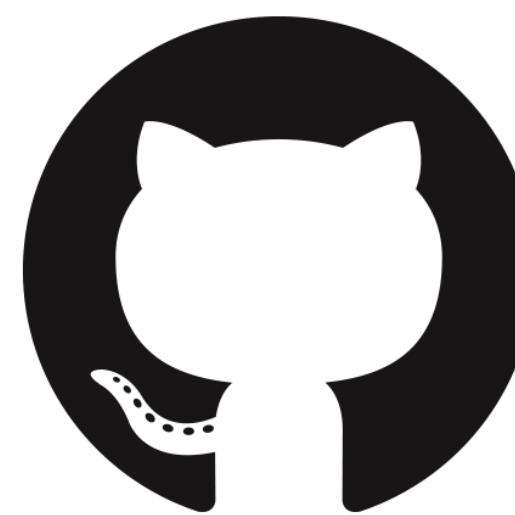
11.8.0-devel-ubuntu22.04 (Latest) Scan Results
No results available.

>>



Isaac ROS base

Example of Isaac ROS package



https://github.com/rbonghi/isaac_ros_base

rbonghi / isaac_ros_base Public

Pin Unwatch 1 Fork 1 Star 4

Code Issues Pull requests Actions Projects Wiki Security Insights ...

main

Go to file

Add file

Code

About



rbonghi	Fix name folder	...	2 days ago	91
.github	Bump dependabot/fetch-metadata fro...		2 weeks ago	
example	Added Isaac package and clean ros-cor...		4 months ago	
isaac_ros_runner	Fix docker compose and installer		4 months ago	
scripts	Update to JP5.1 and removed nav2		last week	
tao	Improve Devel and fix Humble for x86		4 months ago	
vpi	Fix name folder		2 days ago	
.dockerignore	Improve Docker devel and minor fixes		4 months ago	

Multi architecture Isaac ROS
for NVIDIA Jetson and x86 - CI
based

docker

nvidia-docker

ros2

humble

isaac-ros

ros-humble

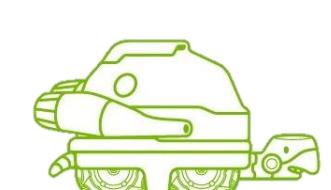
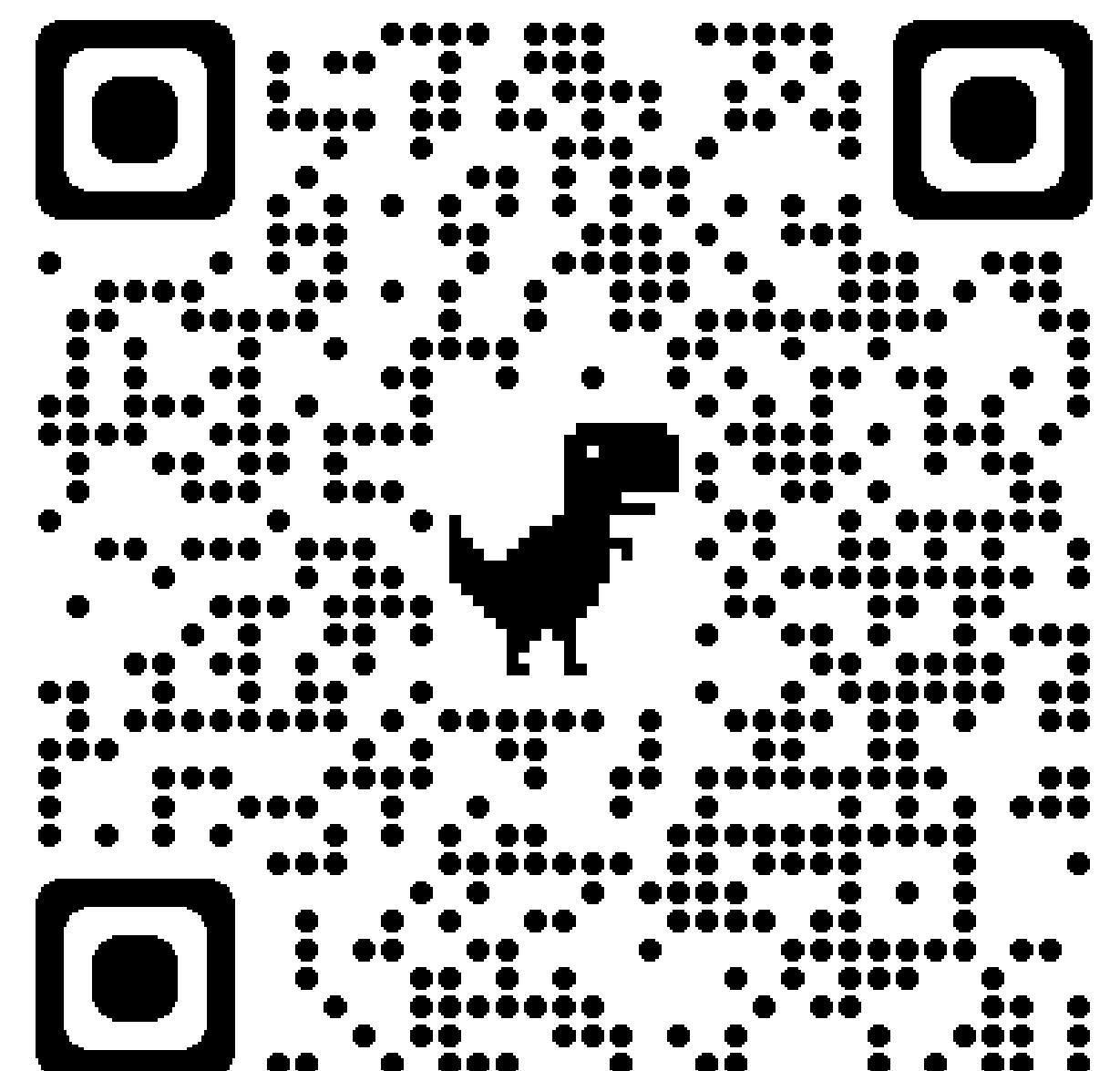
Readme

MIT license

4 stars

1 watching

1 fork



Build our Isaac ROS docker file

Before to start another tool

<https://github.com/dirk-thomas/vcstool>

```
workspace_folder/
  src/
    package_1/
      CMakeLists.txt
      package.xml

    package_2/
      setup.py
      package.xml
      resource/package_2
    ...
    package_n/
      CMakeLists.txt
      package.xml
```

- Supports multiple version control systems (VCS), including Git and Subversion
- Centralized management of packages stored in version control repositories
- Automation of common tasks, such as cloning and updating repositories, checking out specific versions, and more
- Supports the management of dependencies in software development environments
- Can be used in the ROS (Robot Operating System) ecosystem

`sudo apt-get install python3-vcstool`

Write my Isaac ROS workspace

The faster and cool way

my_isaac_ros.rosinstall

```
- git:  
  local-name: isaac_ros_common  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_common.git  
- git:  
  local-name: isaac_ros_nitros  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_nitros.git  
- git:  
  local-name: isaac_ros_image_pipeline  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_image_pipeline.git  
- git:  
  local-name: isaac_ros_apriltag  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_apriltag.git  
- git:  
  local-name: isaac_ros_visual_slam  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_visual_slam.git
```

```
1  FROM rbonghi/isaac_ros_base:humble-base-devel  
2  
3  # Copy wstool isaac_ros.rosinstall  
4  COPY my_isaac_ros.rosinstall /my_isaac_ros.rosinstall  
5  # Import also all Isaac ROS packages  
6  RUN mkdir -p /ws_ros/src \  
7      && cd /ws_ros \  
8      && vcs import $/ws_ros/src < /my_isaac_ros.rosinstall \  
9      && rm /my_isaac_ros.rosinstall \  
10     # Install dependencies using rosdep  
11     # Load variable ROS2  
12     && . /opt/ros/$ROS_DISTRO/install/setup.sh \  
13     && apt-get update \  
14     && rosdep install -y \  
15         --ignore-src \  
16         --from-paths src \  
17         --rosdistro ${ROS_DISTRO} \  
18     && rm -Rf /var/lib/apt/lists/* \  
19     && apt-get clean \  
20     # Build Isaac ROS and clean resources  
21     && colcon build --merge-install --cmake-args -DCMAKE_BUILD_TYPE=Release \  
22     && rm -Rf src build log  
23  
24  CMD ["ros2", "launch", "my_cool_package", "isaac_ros.forever.launch.py"]
```

Write my Isaac ROS workspace

The faster and cool way

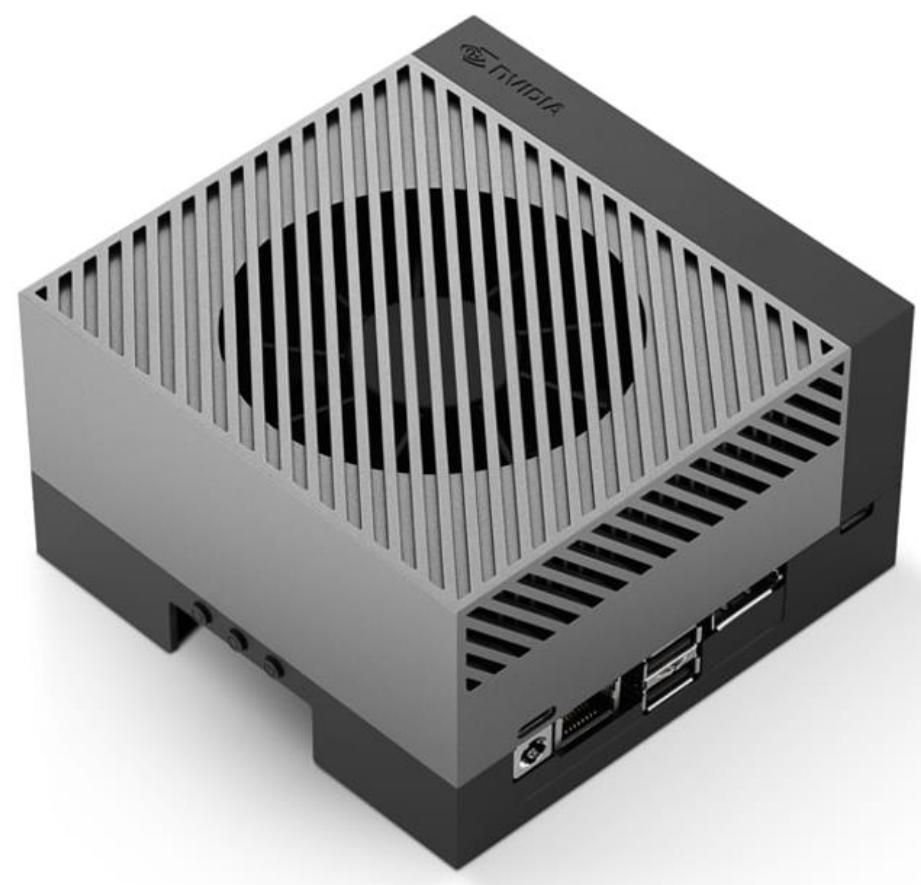
```
1 FROM rbonghi/isaac_ros_base:humble-base-devel
2
3 # Copy wstool isaac_ros.rosinstall
4 COPY my_isaac_ros.rosinstall /my_isaac_ros.rosinstall
5 # Import also all Isaac ROS packages
6 RUN mkdir -p /ws_ros/src \
7     && cd /ws_ros \
8     && vcs import $/ws_ros/src < /my_isaac_ros.rosinstall \
9     && rm /my_isaac_ros.rosinstall \
10    # Install dependencies using rosdep
11    # Load variable ROS2
12    && . /opt/ros/$ROS_DISTRO/install/setup.sh \
13    && apt-get update \
14    && rosdep install -y \
15        --ignore-src \
16        --from-paths src \
17        --rosdistro ${ROS_DISTRO} \
18    && rm -Rf /var/lib/apt/lists/* \
19    && apt-get clean \
20    # Build Isaac ROS and clean resources
21    && colcon build --merge-install --cmake-args -DCMAKE_BUILD_TYPE=Release \
22    && rm -Rf src build log
23
24 CMD ["ros2", "launch", "my_cool_package", "isaac_ros.forever.launch.py"]
```

Where we launch our
ros2 launch.py

Where we build all
Isaac ROS
ROS2
packages

Build and run my Isaac ROS Dockerfile

Same file that you can build on your device



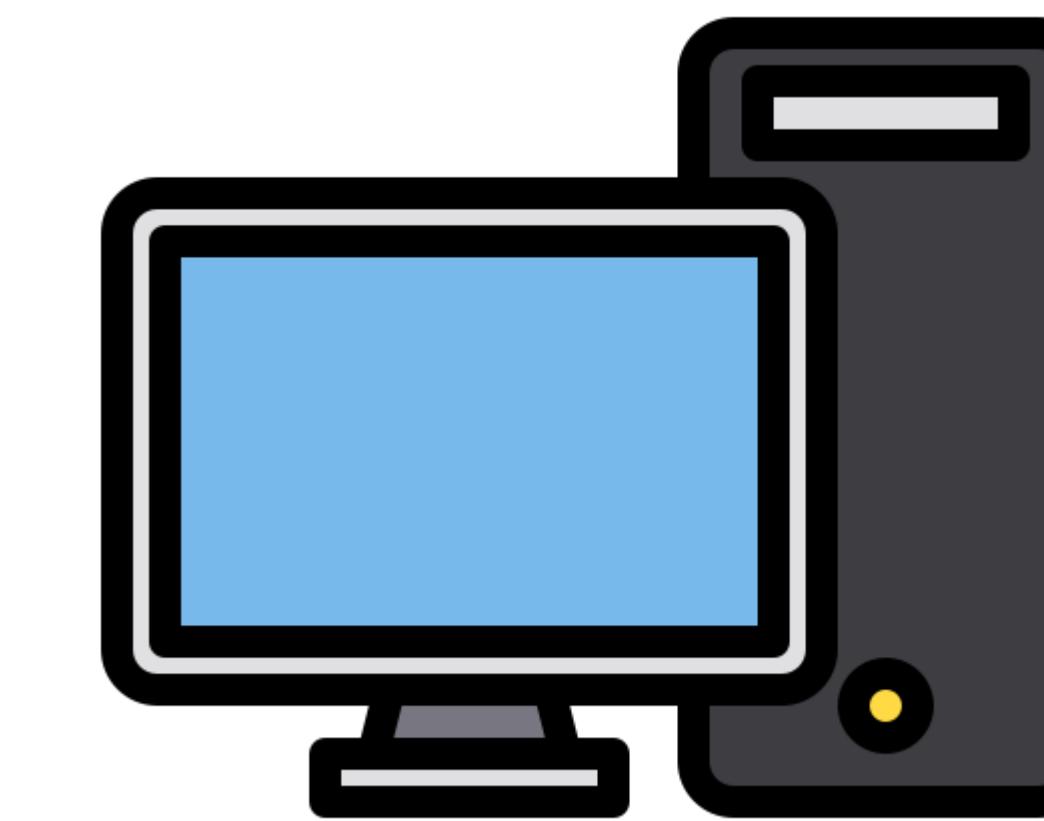
BUILD

From my NVIDIA
Jetson

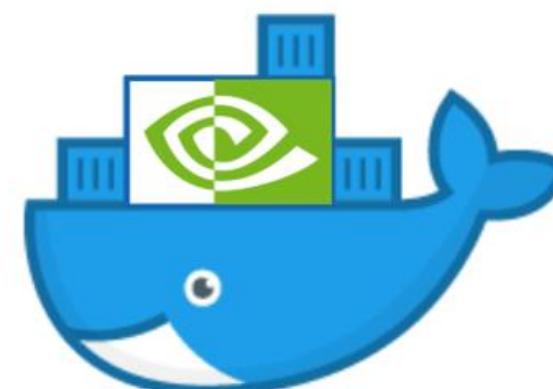
```
docker build -t my_cool_docker .
```

BUILD

From my Desktop



```
docker build -t my_cool_docker .
```



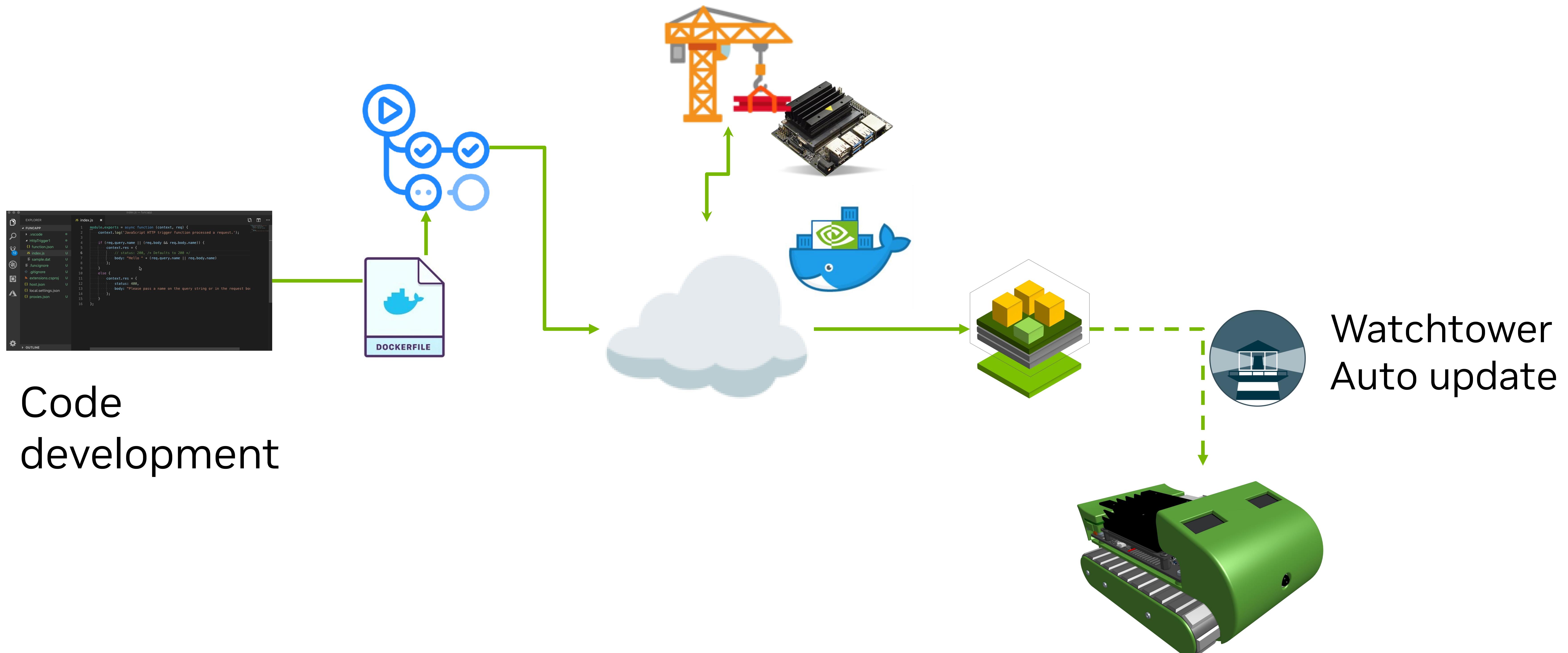
RUN

```
docker run --runtime=nvidia my_cool_docker
```

Nanosaur CI/CD

How a Nanosaur build works

CI - Remote building





Isaac ROS for Jetpack 4.6

Isaac ROS - Tutorial

https://github.com/rbonghi/isaac_ros_tutorial

rbonghi / isaac_ros_tutorial Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main ▾ 2 branches 0 tags Go to file Add file ▾ Code ▾

rbonghi Fix always pull base image	✓ 1ac4fed 5 hours ago	53 commits
.github	Fix always pull base image	5 hours ago
00-isaac_ros_base	Fix ZED launcher + fix image_common stereolabs/zed-ros2-wrapper#66	3 days ago
01-argus_camera	Improvements for ZED camera docker	20 days ago
02-realsense_camera	First draft OAK-D camera	2 days ago
03-zed_camera	First draft OAK-D camera	2 days ago
04-oakd_camera	First draft OAK-D camera	2 days ago
isaac_ros_runner	Add tutorial package for ZED	21 days ago
scripts	Fix always pull base image	5 hours ago
.gitignore	Improve builder script, CI and readme	24 days ago
LICENSE	Add tutorial package for ZED	21 days ago
README.md	First draft OAK-D camera	2 days ago

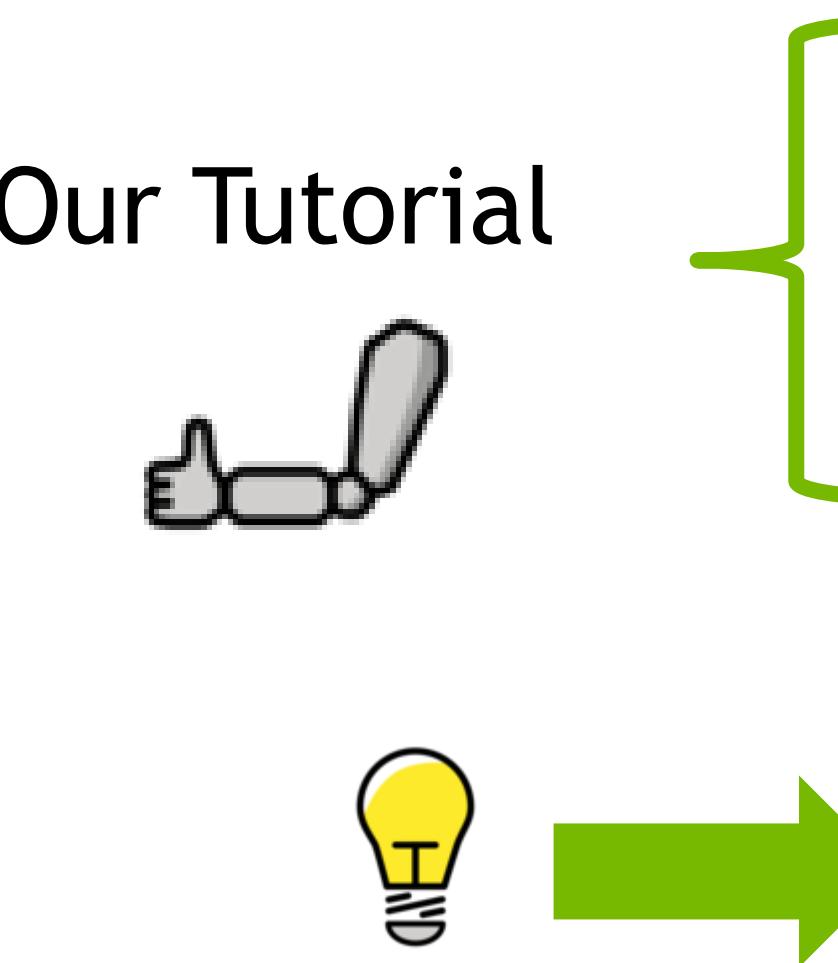
tutorial

💡 ➡

README.md

isaac_ros_tutorial

A set of tutorial to define and setup your docker for each package



Minimal Isaac ROS Dockerfile

ROS2 on NVIDIA Jetson

Docker containers to develop and run Isaac ROS packages

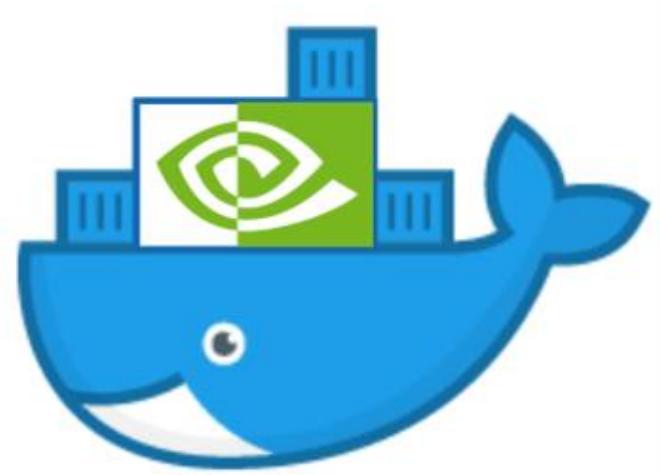
ROS2 Foxy supported on Ubuntu 20.04

However, JetPack 4.6 L4T based on Ubuntu 18.04

- **Solution 1:** Compile ROS2 Foxy from source yourself on JetPack 4.6 and install dependencies manually

- **Solution 2:** Docker image with ROS2 Foxy and useful dependencies compiled from source on L4T

Run container with NVIDIA Container Runtime to access CUDA-X layers on Jetson



ROS2 nodes



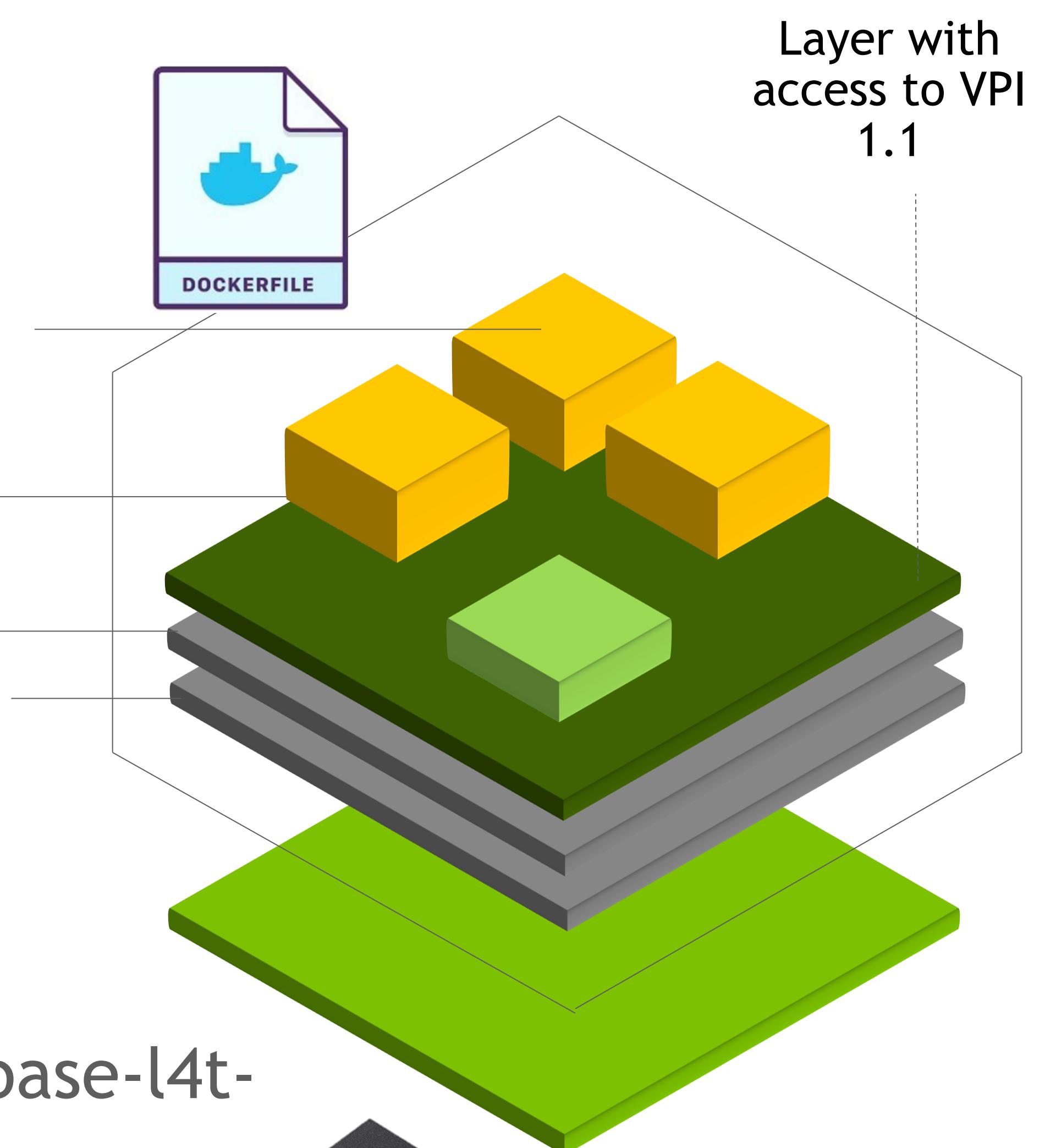
Isaac ROS Common

ROS2 Foxy

NVIDIA Jetson base:

- JetPack 4.6
- Ubuntu 18.04
- CUDA 10.2

dustynv/ros:foxy-ros-base-l4t-4.6.1



Setup your NVIDIA JETSON

Enabling Docker build

1

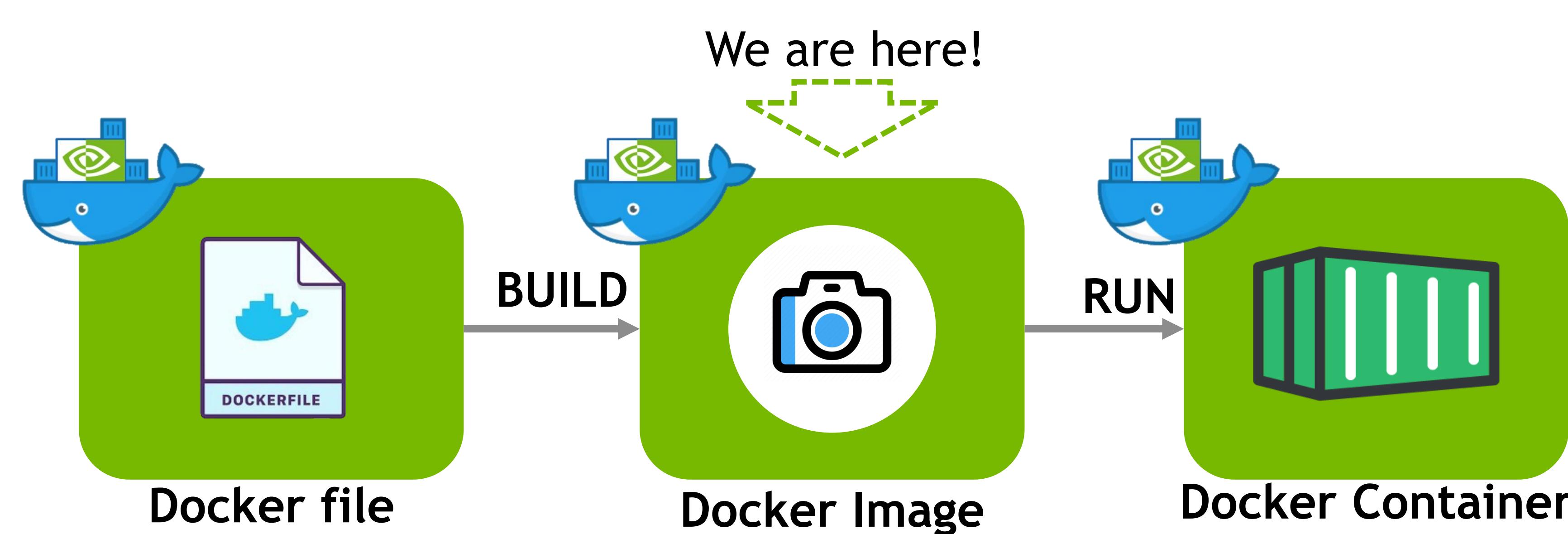
To enable access to the [CUDA compiler](#) (nvcc) during [docker build](#) operations, add "default-runtime": "nvidia" to your [/etc/docker/daemon.json](#) configuration file before attempting to build the containers:

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

2

Enable jetson_multimedia_api for [docker build](#) adding in [/etc/nvidia-container-runtime/host-files-for-container.d](#)

1	dir	/usr/src/jetson_multimedia_api
---	-----	--------------------------------

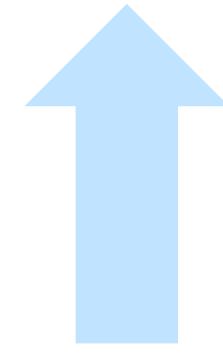


There is a script for that...
let's wait!

How to build a minimal Isaac ROS docker image

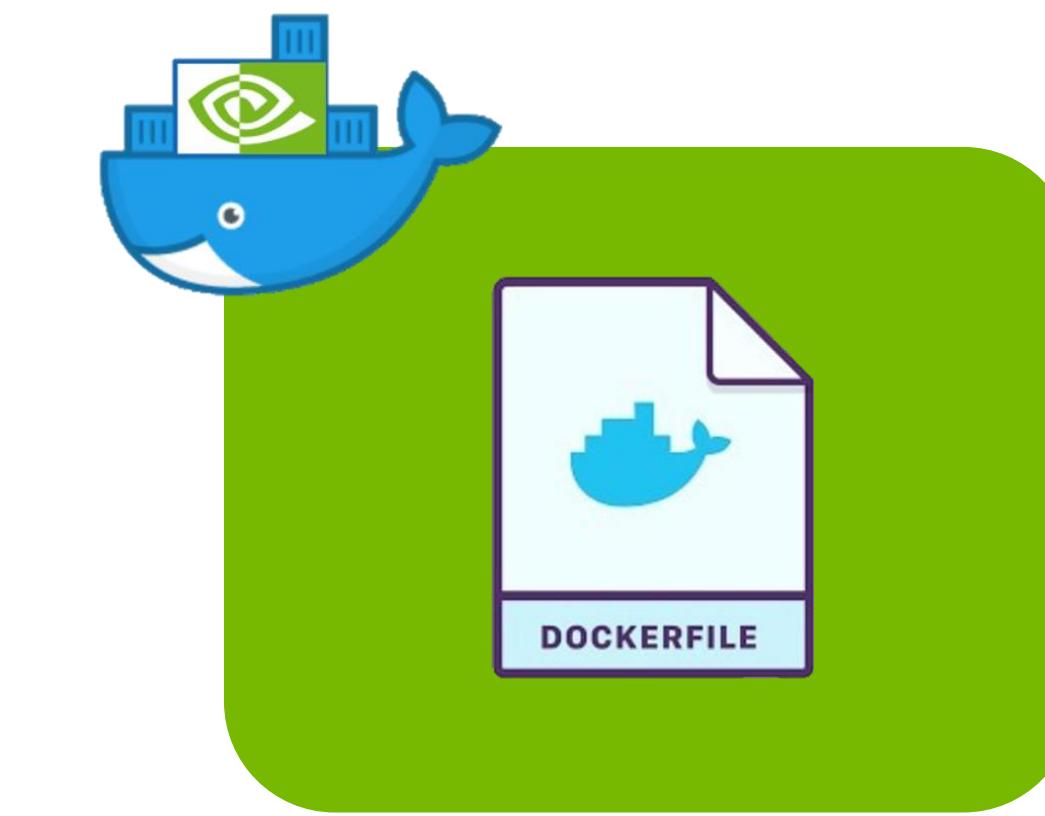
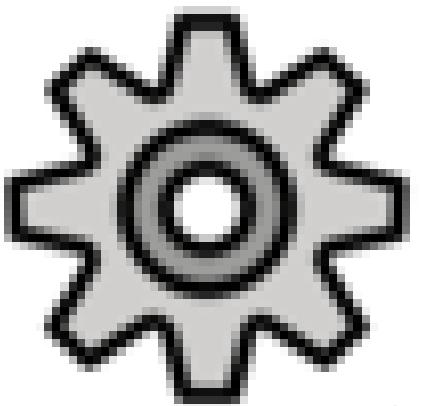
All steps to build a Docker Image

```
main rbonghi
isaac_ros_tutorial / 00-isaac_ros_base /
Dockerfile
isaac_ros_base.rosinstall
```

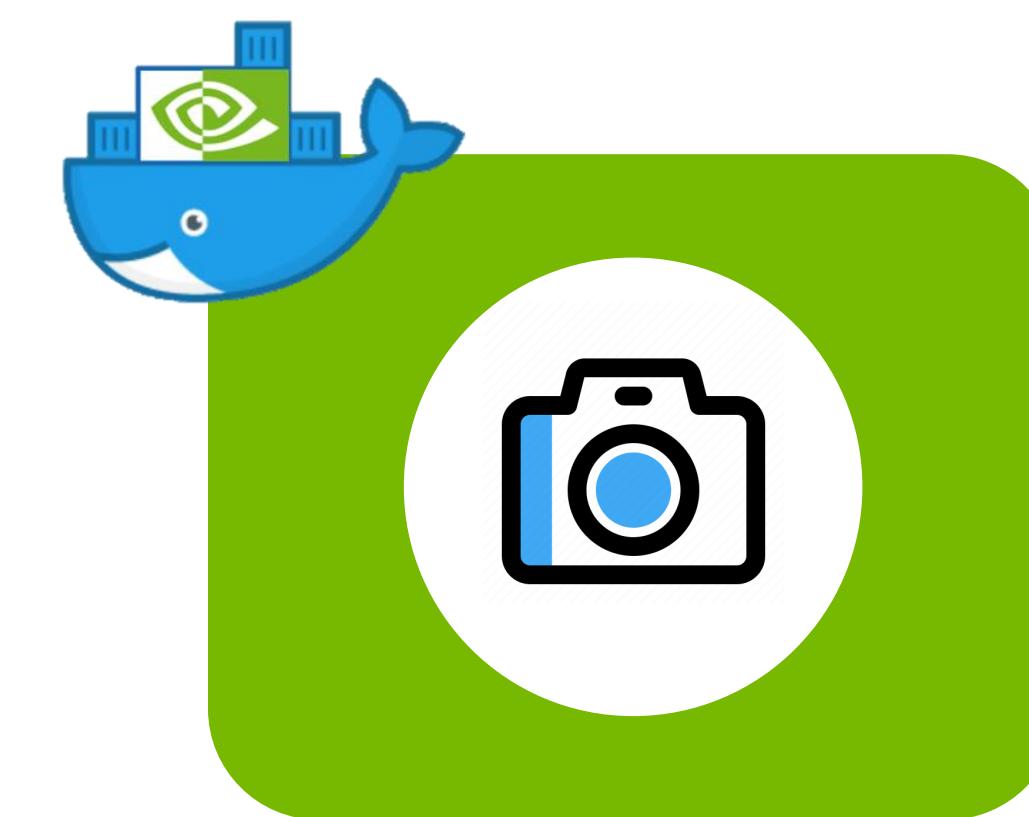


vcs importer

bash script/build_docker.sh <FOLDER_NAME>



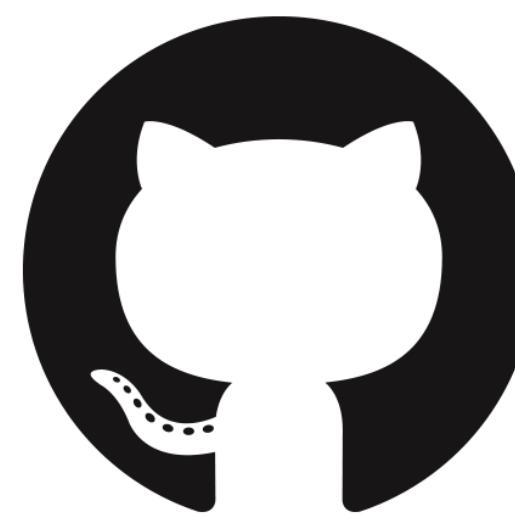
Docker file



Docker Image

```
21 # Jetpack 4.6
22 # Docker file for aarch64 based Jetson device
23 ARG BASE_IMAGE="dustynv/ros:foxy-ros-base-14t-r32.6.1"
24 FROM ${BASE_IMAGE}
25 # Environment variables
26 ENV L4T_MINOR_VERSION=6
27 ENV JETPACK_MAJOR=4
28 ENV JETPACK_MINOR=6
29 # Configuration CUDA
30 ARG CUDA=10.2
31 ARG L4T=r32.${L4T_MINOR_VERSION}
32
33 # Disable terminal interaction for apt
34 ENV DEBIAN_FRONTEND=noninteractive
35
36 # Install OpenCV dependencies
37 RUN apt-get update && apt-get install -y \
38     libavformat-dev \
39     libjpeg-dev \
40     libopenjp2-7-dev \
41     libpng-dev \
42     libpq-dev \
43     libswscale-dev \
44     libtbb2 \
45     libtbb-dev \
46     libtiff-dev \
47     pkg-config \
48     yasm && \
49     rm -rf /var/lib/apt/lists/*
```

NVIDIA Isaac ROS - Template for JP4.6



https://github.com/rbonghi/isaac_ros_template

rbonghi / isaac_ros_template Public template

Pin Unwatch 1 Fork 0 Star 6

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

foxy 1 branch 0 tags Go to file Add file Use this template

rbonghi Update README.md d0386b4 on May 4, 2022 23 commits

.images	Add last image readme	10 months ago
scripts	Fix build image with L4T check and docker runtime nvidia check	10 months ago
01_isaac_ros.rosinstall	errange files	10 months ago
02_your_ros2_pkgs.rosinstall	errange files	10 months ago
Dockerfile	Add last image readme	10 months ago
LICENSE	First commit	10 months ago
README.md	Update README.md	10 months ago
build_image.bash	Improve readme	10 months ago

Use this template

About

This template repository is made for build your personal Isaac ROS container

nvidia nvidia-docker nvidia-jetson
isaac-ros

Readme MIT license 6 stars 1 watching 0 forks

Releases

No releases published Create a new release

NVIDIA

NVIDIA Isaac ROS – Template for JP4.6

Example

 [rbonghi / isaac_ros_realsense_example](#) Public

generated from [rbonghi/isaac_ros_template](#)

[Pin](#) [Unwatch 1](#) [Fork 0](#) [Star 1](#)

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[main](#) [1 branch](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

 [rbonghi](#) Clean and example with Realsense 44a2e80 on May 4, 2022 2 commits

 scripts	Initial commit	10 months ago
 01_isaac_ros.rosinstall	Clean and example with Realsense	10 months ago
 02_your_ros2_pkgs.rosinstall	Initial commit	10 months ago
 Dockerfile	Clean and example with Realsense	10 months ago
 LICENSE	Initial commit	10 months ago
 README.md	Clean and example with Realsense	10 months ago
 build_image.bash	Initial commit	10 months ago

 README.md

 **Isaac ROS Template**

About

Example use case from [isaac_ros_template](#)

[Readme](#) [MIT license](#) [1 star](#) [1 watching](#) [0 forks](#)

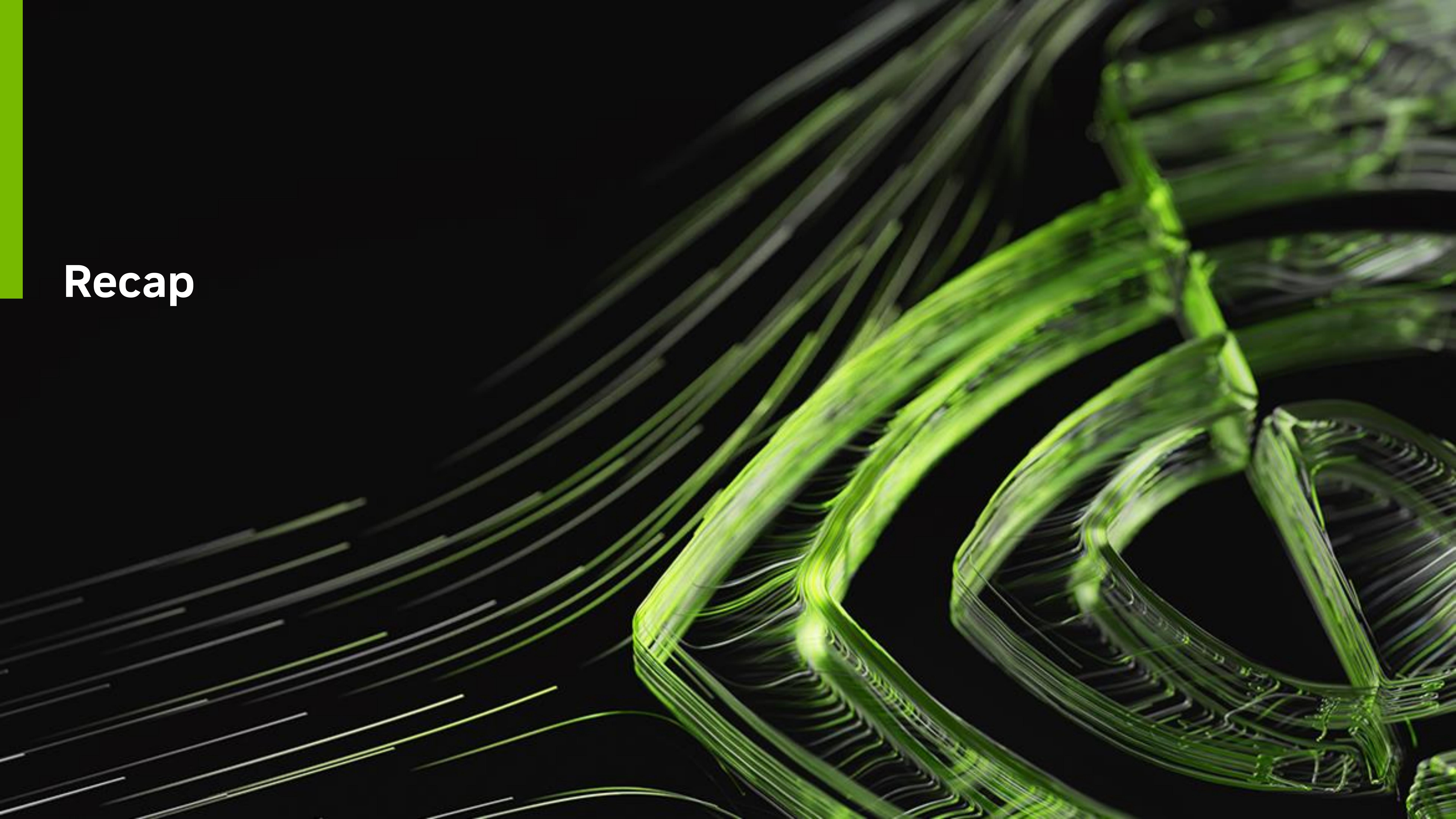
Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)



The background of the slide features a dense, abstract pattern of glowing green lines against a black background. These lines vary in thickness and intensity, creating a sense of depth and motion. Some lines are straight, while others curve or twist, forming a complex web-like structure.

Recap

NVIDIA Isaac ROS Resources

<https://github.com/NVIDIA-ISAAC-ROS>



NVIDIA Isaac ROS

High-performance computing for robotics

📍 United States of America [🔗 https://developer.nvidia.com/isaac-ros...](https://developer.nvidia.com/isaac-ros)

🏠 Overview 🚦 Repositories 8 🏷 Packages 🔎 People 📁 Projects

Pinned

💻 [isaac_ros_apriltag](#) Public

CUDA-accelerated Apriltag detection and pose estimation.

● C++ ⭐ 21 🏷 3

💻 [isaac_ros_visual_odometry](#) Public

Visual odometry package based on hardware-accelerated NVIDIA Elbrus library with world class quality and performance.

● C++ ⭐ 88 🏷 7

💻 [isaac_ros_image_pipeline](#) Public

Hardware-accelerated ROS2 packages for camera image processing.

● Python ⭐ 19 🏷 2

💻 [isaac_ros_argus_camera](#) Public

ROS2 packages based on NVIDIA libArgus library for hardware-accelerated CSI camera support.

● C++ ⭐ 15 🏷 3

💻 [isaac_ros_image_segmentation](#) Public

Hardware-accelerated, deep learned semantic image segmentation

● C++ ⭐ 6

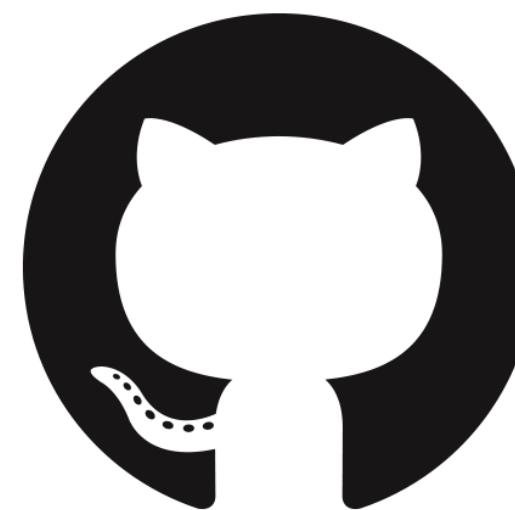
💻 [isaac_ros_pose_estimation](#) Public

Deep learned, hardware-accelerated 3D object pose estimation

● Python ⭐ 16 🏷 3

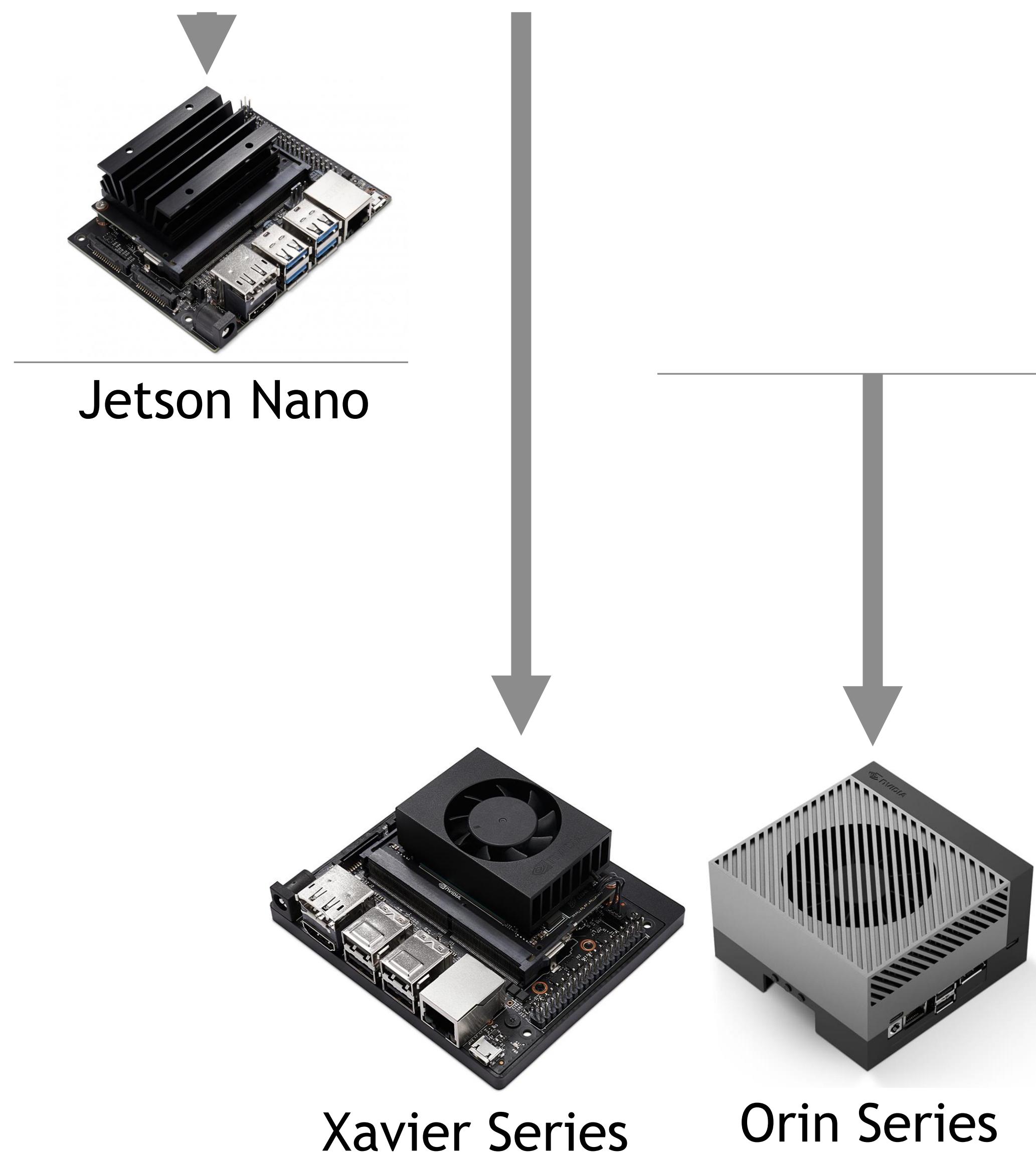
Isaac ROS versions

Some hints when you work with Isaac ROS



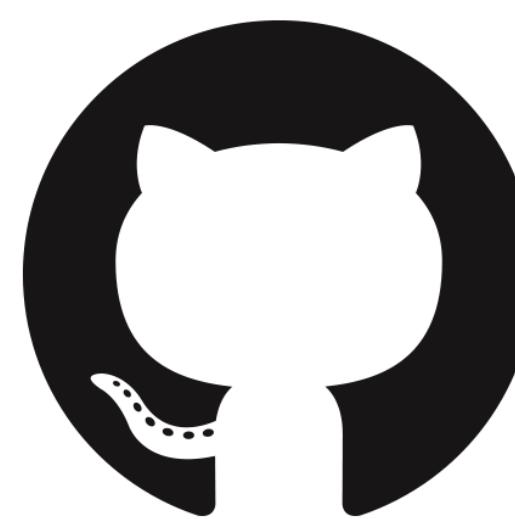
<https://github.com/NVIDIA-ISAAC-ROS/.github/blob/main/profile/release-notes.md>

Version	Release date	NVIDIA Jetpack	ROS
EA1	Aug 11, 2021	4.5	Foxy
EA2	Oct 20, 2021	4.6	Foxy
0.9.1 EA2.1	Nov 22, 2021	4.6	Foxy
0.93 EA3	Mar 23, 2022	4.6.1	Foxy
0.10.0 DP	June 30, 2022	5.0.1 DP	Humble
0.10.1 DP	July 12, 2022	5.0.1 DP	Humble
0.11.0 DP	Sep 1, 2022	5.0.2	Humble
0.20.0 DP	Oct 19, 2022	5.0.2	Humble



Isaac ROS base

Example of Isaac ROS package



https://github.com/rbonghi/isaac_ros_base

rbonghi / isaac_ros_base Public

Pin Unwatch 1 Fork 1 Star 4

Code Issues Pull requests Actions Projects Wiki Security Insights ...

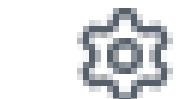
main

Go to file

Add file

Code

About



rbonghi	Fix name folder	...	2 days ago	91
.github	Bump dependabot/fetch-metadata fro...		2 weeks ago	
example	Added Isaac package and clean ros-cor...		4 months ago	
isaac_ros_runner	Fix docker compose and installer		4 months ago	
scripts	Update to JP5.1 and removed nav2		last week	
tao	Improve Devel and fix Humble for x86		4 months ago	
vpi	Fix name folder		2 days ago	
.dockerignore	Improve Docker devel and minor fixes		4 months ago	

Multi architecture Isaac ROS
for NVIDIA Jetson and x86 - CI
based

docker nvidia-docker ros2

humble isaac-ros ros-humble

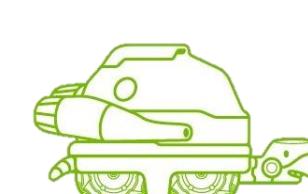
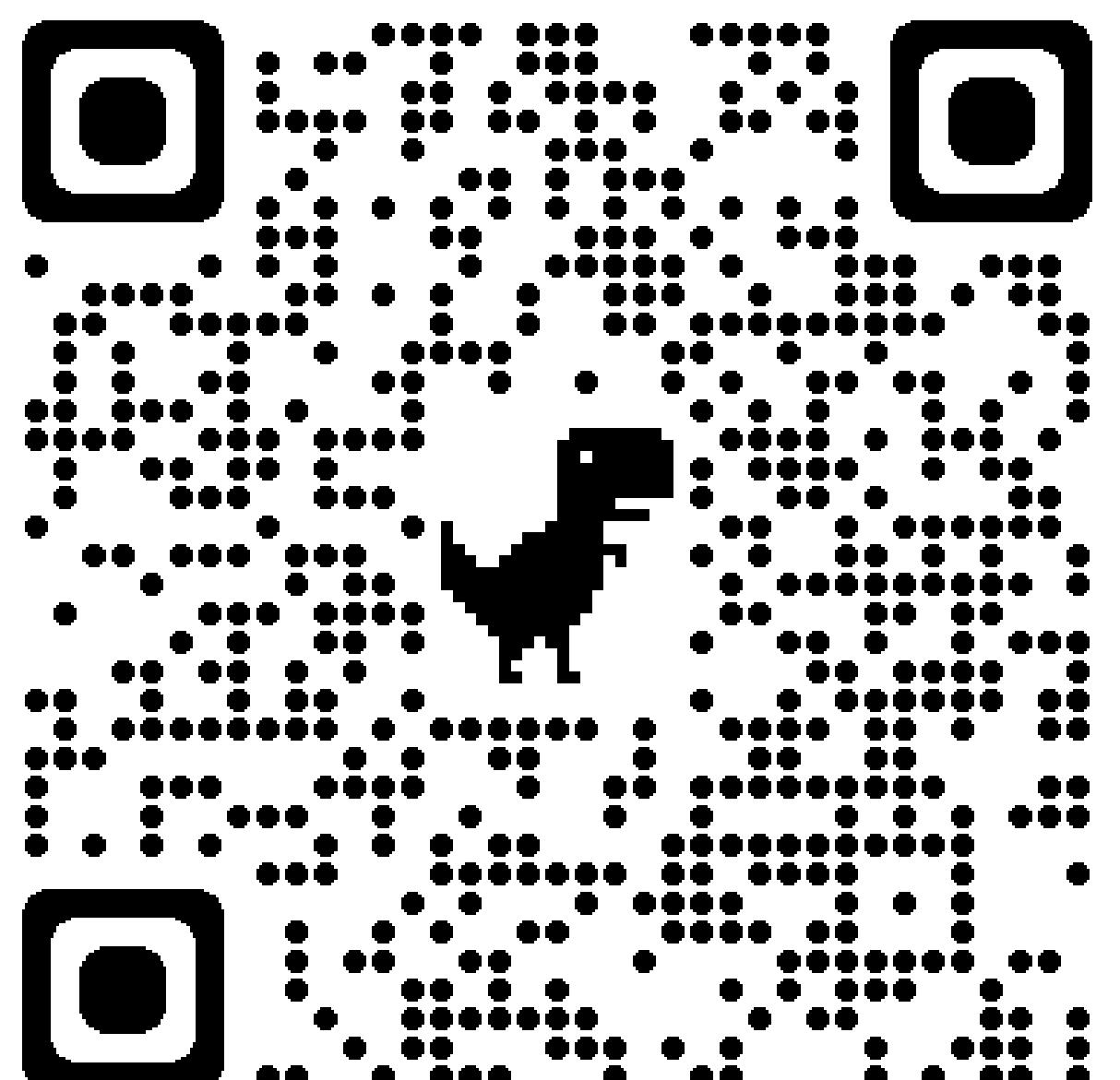
Readme

MIT license

4 stars

1 watching

1 fork



Write my Isaac ROS workspace

The faster and cool way

my_isaac_ros.rosinstall

```
- git:  
  local-name: isaac_ros_common  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_common.git  
- git:  
  local-name: isaac_ros_nitros  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_nitros.git  
- git:  
  local-name: isaac_ros_image_pipeline  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_image_pipeline.git  
- git:  
  local-name: isaac_ros_apriltag  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_apriltag.git  
- git:  
  local-name: isaac_ros_visual_slam  
  uri: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_visual_slam.git
```

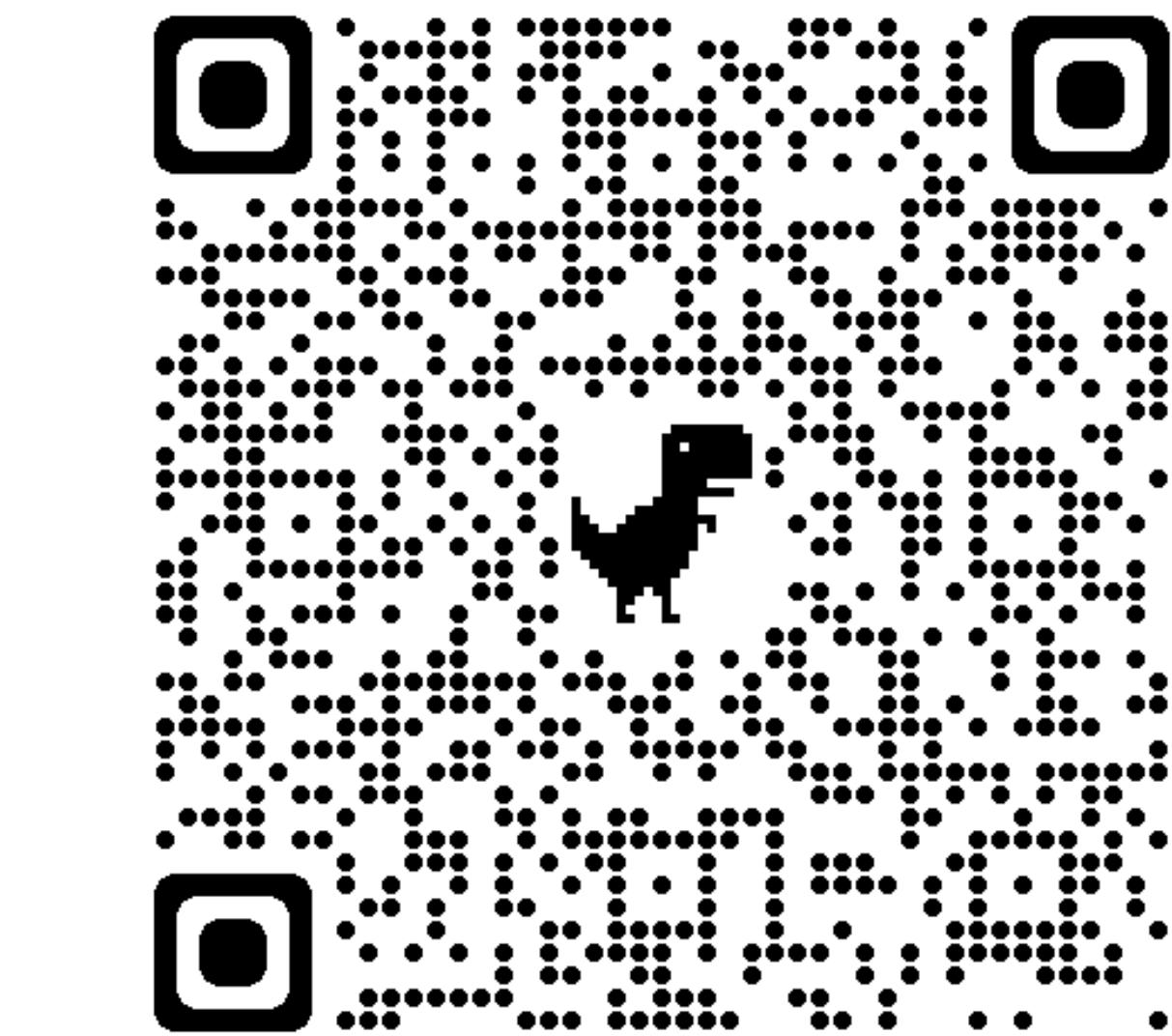
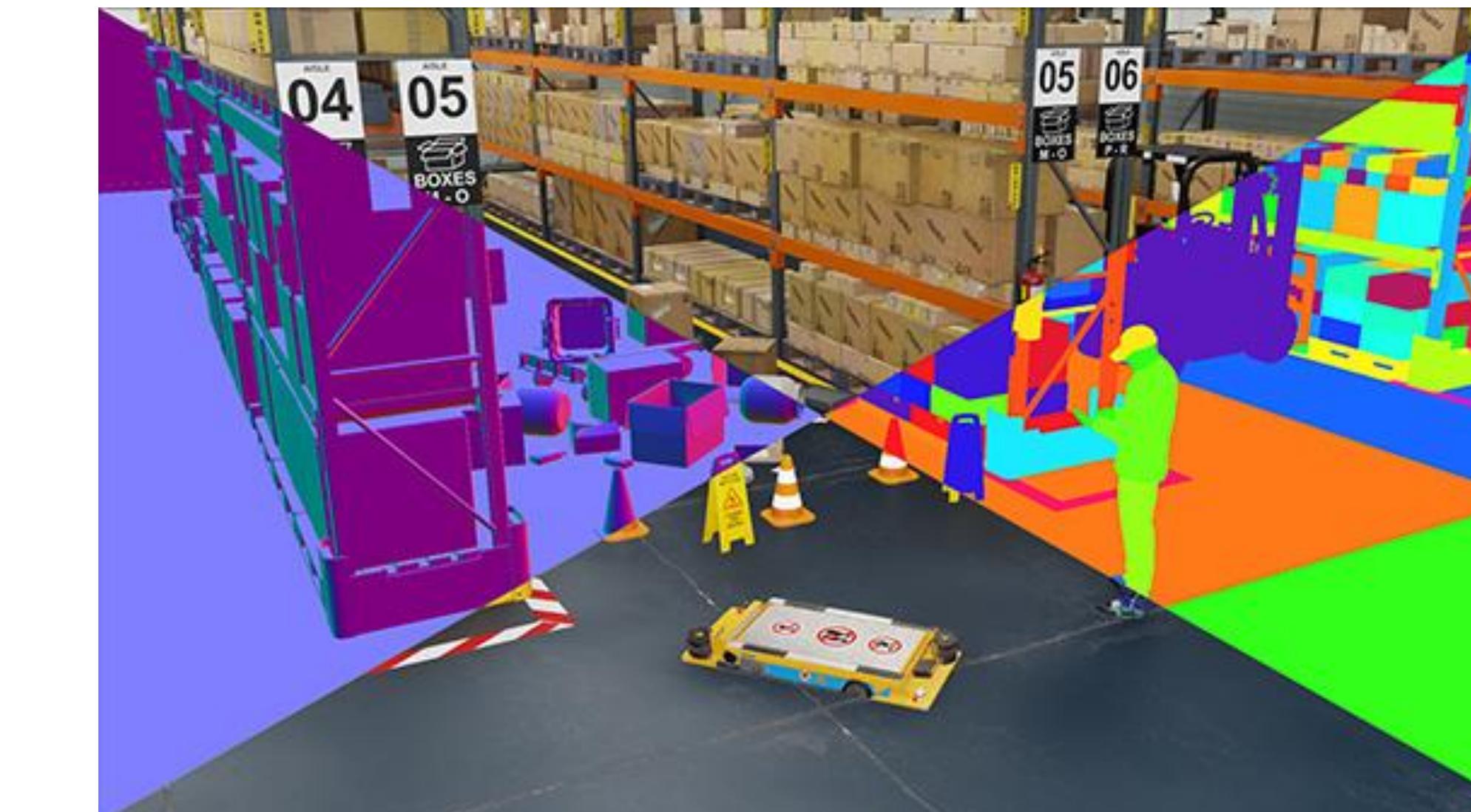
```
1  FROM rbonghi/isaac_ros_base:humble-base-devel  
2  
3  # Copy wstool isaac_ros.rosinstall  
4  COPY my_isaac_ros.rosinstall /my_isaac_ros.rosinstall  
5  # Import also all Isaac ROS packages  
6  RUN mkdir -p /ws_ros/src \  
7      && cd /ws_ros \  
8      && vcs import $/ws_ros/src < /my_isaac_ros.rosinstall \  
9      && rm /my_isaac_ros.rosinstall \  
10     # Install dependencies using rosdep  
11     # Load variable ROS2  
12     && . /opt/ros/$ROS_DISTRO/install/setup.sh \  
13     && apt-get update \  
14     && rosdep install -y \  
15         --ignore-src \  
16         --from-paths src \  
17         --rosdistro ${ROS_DISTRO} \  
18     && rm -Rf /var/lib/apt/lists/* \  
19     && apt-get clean \  
20     # Build Isaac ROS and clean resources  
21     && colcon build --merge-install --cmake-args -DCMAKE_BUILD_TYPE=Release \  
22     && rm -Rf src build log  
23  
24  CMD ["ros2", "launch", "my_cool_package", "isaac_ros.forever.launch.py"]
```



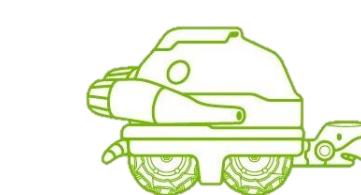
How to Generate Synthetic Data for Training AI Models with Isaac Replicator

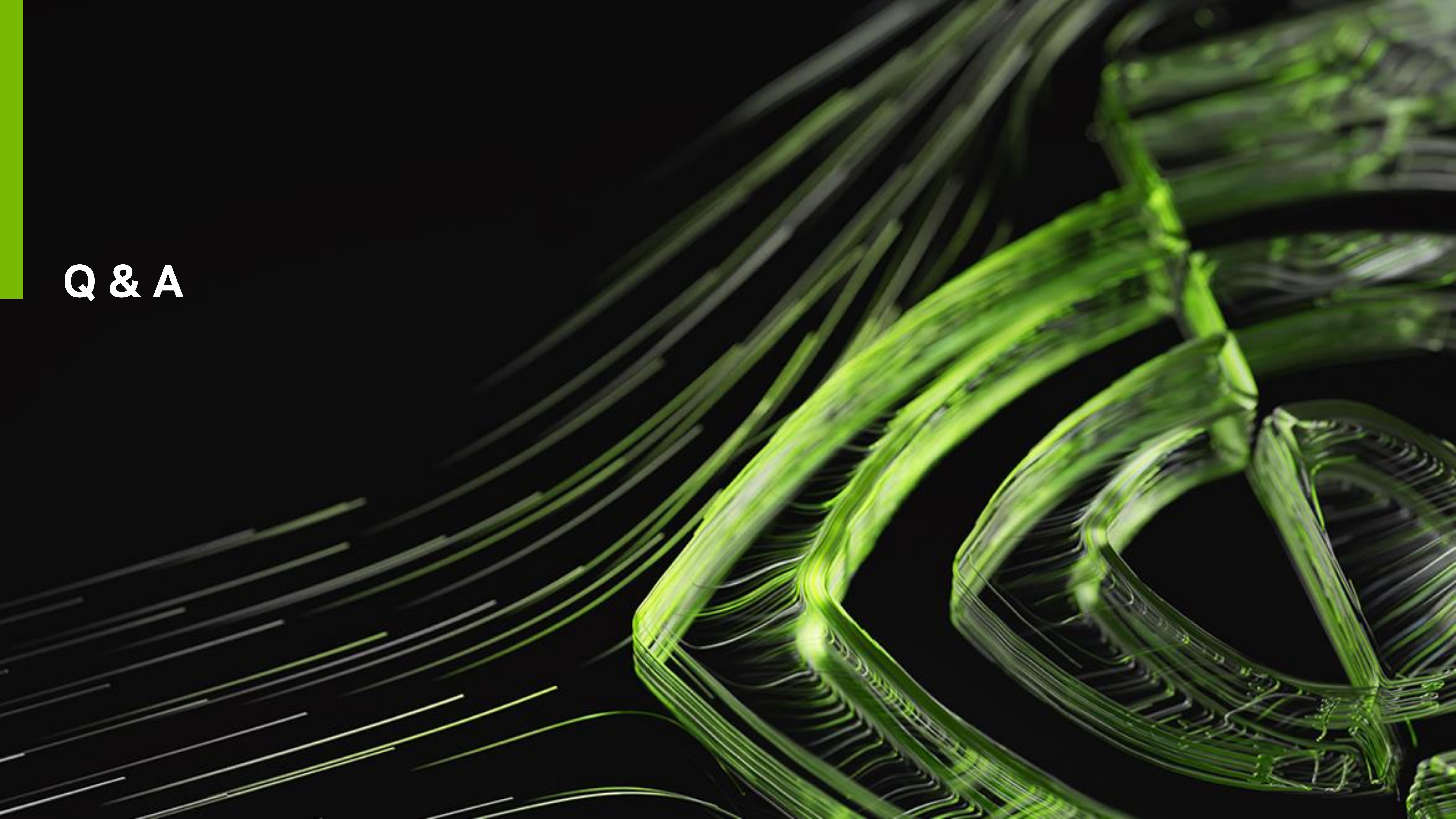
Use Isaac Replicator to build production-quality synthetic datasets to train your deep learning perception models.

23rd February 2023 – 8 am PST / 5 pm CEST



<https://gateway.on24.com/wcc/experience/elitenvidiabril/1407606/4076607/isaac-sim-series>



The background of the slide features a complex, abstract pattern of glowing green lines against a black background. These lines vary in thickness and intensity, creating a sense of depth and motion. Some lines are sharp and straight, while others are blurred and curved, suggesting speed or energy flow. The overall effect is futuristic and dynamic.

Q & A

