1、Depth camera usage

Raspberry Pi PI5 master needs to enter the docker container first. Orin motherboard does not need to enter, and the device number has been bound

Before driving a depth camera, it is necessary for the host to be able to recognize the Astra camera device; When entering the Docker container, you need to mount this Astra device to recognize the camera in the Docker container. The supporting host has already been built in an environment and does not require additional configuration. If it is on a new host, a rule file needs to be added. The addition method is very simple. Copy the "/etc/udev rules. d/56 orbbec-usb rules" file from the host computer to the "/etc/udev rules. d" directory in the new environment, and then restart system

1、Program Function Description

After the program runs, drive the Astra camera to obtain color RGB, depth, infrared IR image information, and depth point cloud information.

2、Program code reference path

After entering the Docker container, the source code for this function is located at,

```
~/yahboomcar_ros2_ws/software/library_ws/src/ros2_astra_camera/astra_camera/launch/
```

3、Program startup

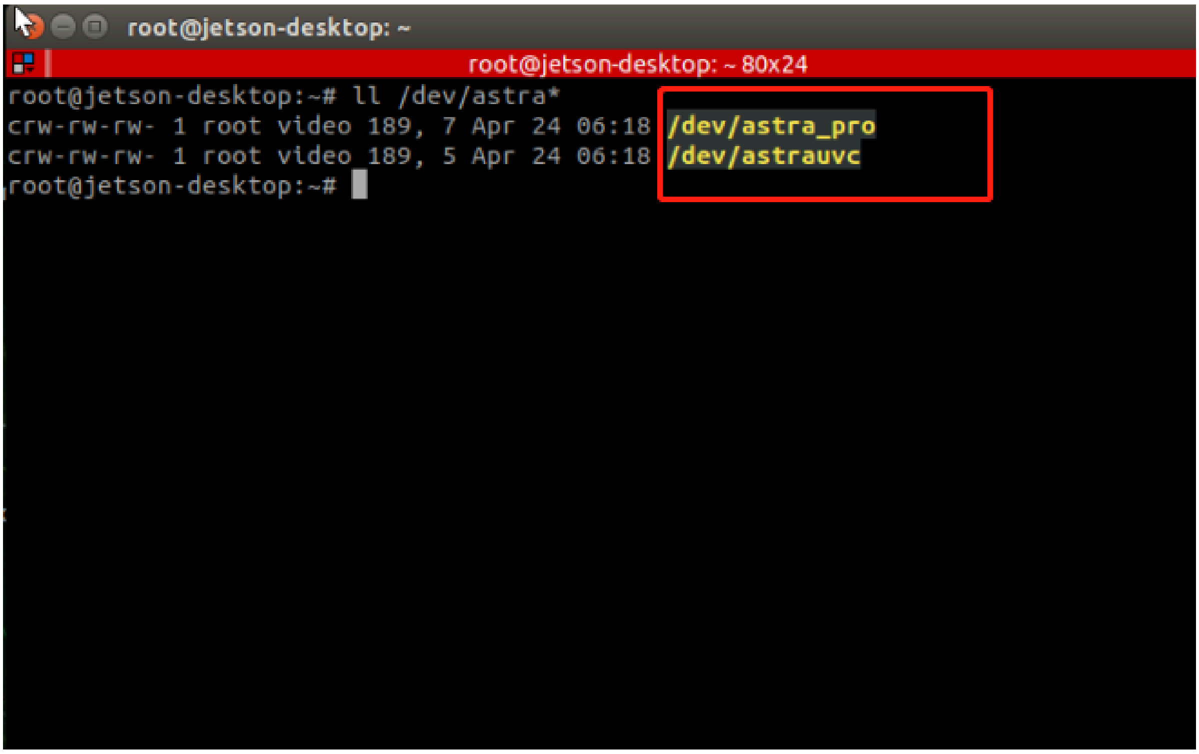
Launch startup command

Launch file	Camera model
ros2 launch astra_camera astra_pro.launch.xml	Astrapro
ros2 launch astra_camera astro_pro_plus.launch.xml	Astraproplus
ros2 launch astra_camera astra.launch.xml	Astramini

Taking starting the Astrapro camera as an example, after entering the Docker container, the terminal enters the following command：

```
ll /dev/as*
```

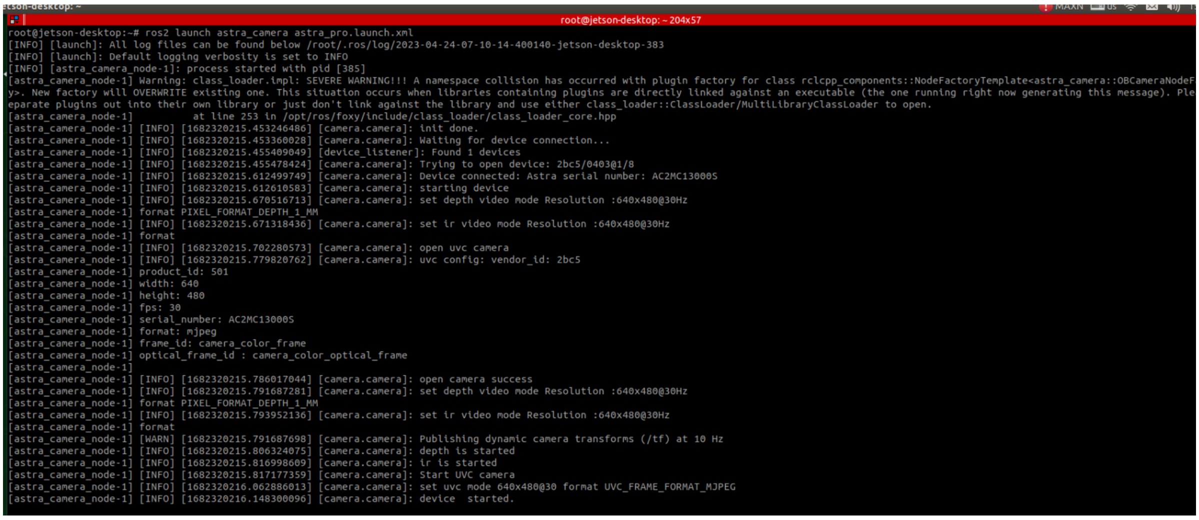
Taking the Astrapro camera as an example, after entering the Docker container, the terminal inputs the camera model,



```
root@jetson-desktop:~# ll /dev/astra*
crw-rw-rw- 1 root video 189, 7 Apr 24 06:18 /dev/astra_pro
crw-rw-rw- 1 root video 189, 5 Apr 24 06:18 /dev/astrauvc
root@jetson-desktop:~# 
```

Enter in the Docker terminal,

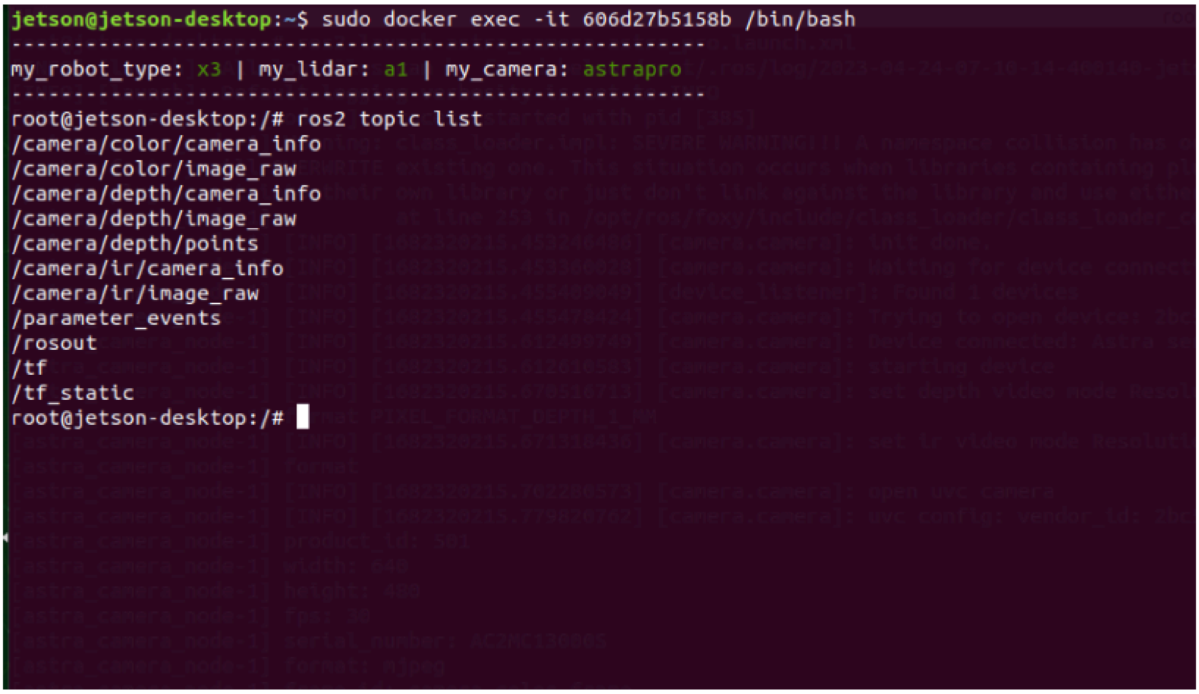
```
ros2 launch astra_camera astra_pro.launch.xml
```



```
root@jetson-desktop:~# ros2 launch astra_camera astra_pro.launch.xml
[INFO] [launch]: All log files can be found below /root/.ros/log/2023-04-24-07-10-14-400140-jetson-desktop-383
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [astra_camera_node-1]: process started with pid [385]
[astra_camera_node-1] Warning: class_loader.impl: SEVERE WARNING!!! A namespace collision has occurred with plugin factory for class rclcpp_components::NodeFactoryTemplate<astra_camera::OBCameraNodeF
y>. New factory will OVERWRITE existing one. This situation occurs when libraries containing plugins are directly linked against an executable (the one running right now generating this message). Ple
eparate plugins out into their own library or just don't link against the library and use either class_loader::ClassLoader/MultiLibraryClassLoader to open.
[astra_camera_node-1]          at line 253 in /opt/ros/foxy/include/class_loader/class_loader_core.hpp
[astra_camera_node-1] [INFO] [1682320215.453246486] [camera.camera]: init done.
[astra_camera_node-1] [INFO] [1682320215.453360028] [camera.camera]: Waiting for device connection...
[astra_camera_node-1] [INFO] [1682320215.455409049] [device_listener]: Found 1 devices
[astra_camera_node-1] [INFO] [1682320215.455478424] [camera.camera]: Trying to open device: 2bc5/0403@1/8
[astra_camera_node-1] [INFO] [1682320215.612499749] [camera.camera]: Device connected: Astra serial number: AC2MC13000S
[astra_camera_node-1] [INFO] [1682320215.612610583] [camera.camera]: starting device
[astra_camera_node-1] [INFO] [1682320215.670516713] [camera.camera]: set depth video mode Resolution :640x480@30Hz
[astra_camera_node-1] format PIXEL_FORMAT_DEPTH_1_MM
[astra_camera_node-1] [INFO] [1682320215.671318436] [camera.camera]: set ir video mode Resolution :640x480@30Hz
[astra_camera_node-1] format
[astra_camera_node-1] [INFO] [1682320215.702280573] [camera.camera]: open uvc camera
[astra_camera_node-1] [INFO] [1682320215.779820762] [camera.camera]: uvc config: vendor_id: 2bc5
[astra_camera_node-1] product_id: 501
[astra_camera_node-1] width: 640
[astra_camera_node-1] height: 480
[astra_camera_node-1] fps: 30
[astra_camera_node-1] serial_number: AC2MC13000S
[astra_camera_node-1] format: mjpeg
[astra_camera_node-1] frame_id: camera_color_frame
[astra_camera_node-1] optical_frame_id : camera_color_optical_frame
[astra_camera_node-1]
[astra_camera_node-1] [INFO] [1682320215.786017044] [camera.camera]: open camera success
[astra_camera_node-1] [INFO] [1682320215.791687281] [camera.camera]: set depth video mode Resolution :640x480@30Hz
[astra_camera_node-1] format PIXEL_FORMAT_DEPTH_1_MM
[astra_camera_node-1] [INFO] [1682320215.793952136] [camera.camera]: set ir video mode Resolution :640x480@30Hz
[astra_camera_node-1] format
[astra_camera_node-1] [WARN] [1682320215.791687698] [camera.camera]: Publishing dynamic camera transforms (/tf) at 10 Hz
[astra_camera_node-1] [INFO] [1682320215.806324075] [camera.camera]: depth is started
[astra_camera_node-1] [INFO] [1682320215.816998609] [camera.camera]: ir is started
[astra_camera_node-1] [INFO] [1682320215.817177359] [camera.camera]: Start UVC camera
[astra_camera_node-1] [INFO] [1682320216.062886013] [camera.camera]: set uvc mode 640x480@30 format UVC_FRAME_FORMAT_MJPEG
[astra_camera_node-1] [INFO] [1682320216.148300096] [camera.camera]: device  started.
```

You can use the following command to view topics, enter in the Docker terminal,

```
ros2 topic list
```



```
jetson@jetson-desktop:~$ sudo docker exec -it 606d27b5158b /bin/bash
-------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
-------------------------------------------------------
root@jetson-desktop:/# ros2 topic list
/camera/color/camera_info
/camera/color/image_raw
/camera/depth/camera_info
/camera/depth/image_raw
/camera/depth/points
/camera/ir/camera_info
/camera/ir/image_raw
/parameter_events
/rosout
/tf
/tf_static
root@jetson-desktop:/# 
```

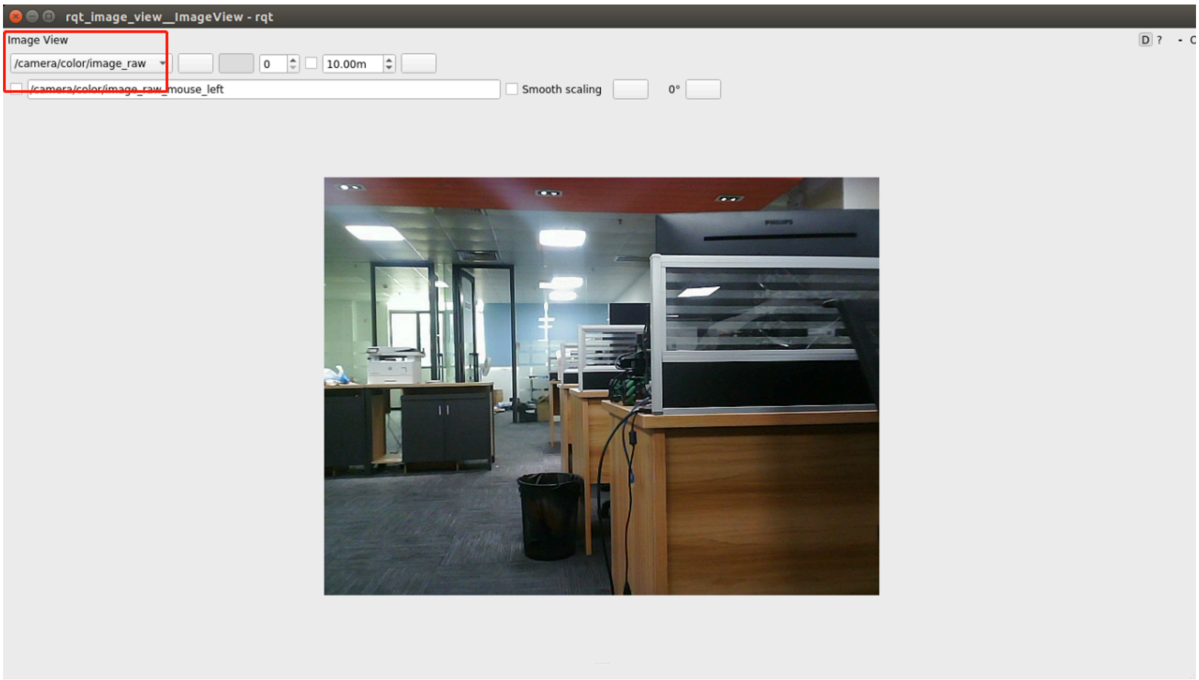
The main topics are as follows:

Topic Name	topical contents
/camera/color/image_raw	RGB color image data
/camera/depth/image_raw	Depth image data
/camera/depth/points	Depth point cloud data
/camera/ir/image_raw	IR infrared image data

Using rqt_ Image_ View tool to view image data, input in Docker terminal,

```
ros2 run rqt_image_view rqt_image_view
```

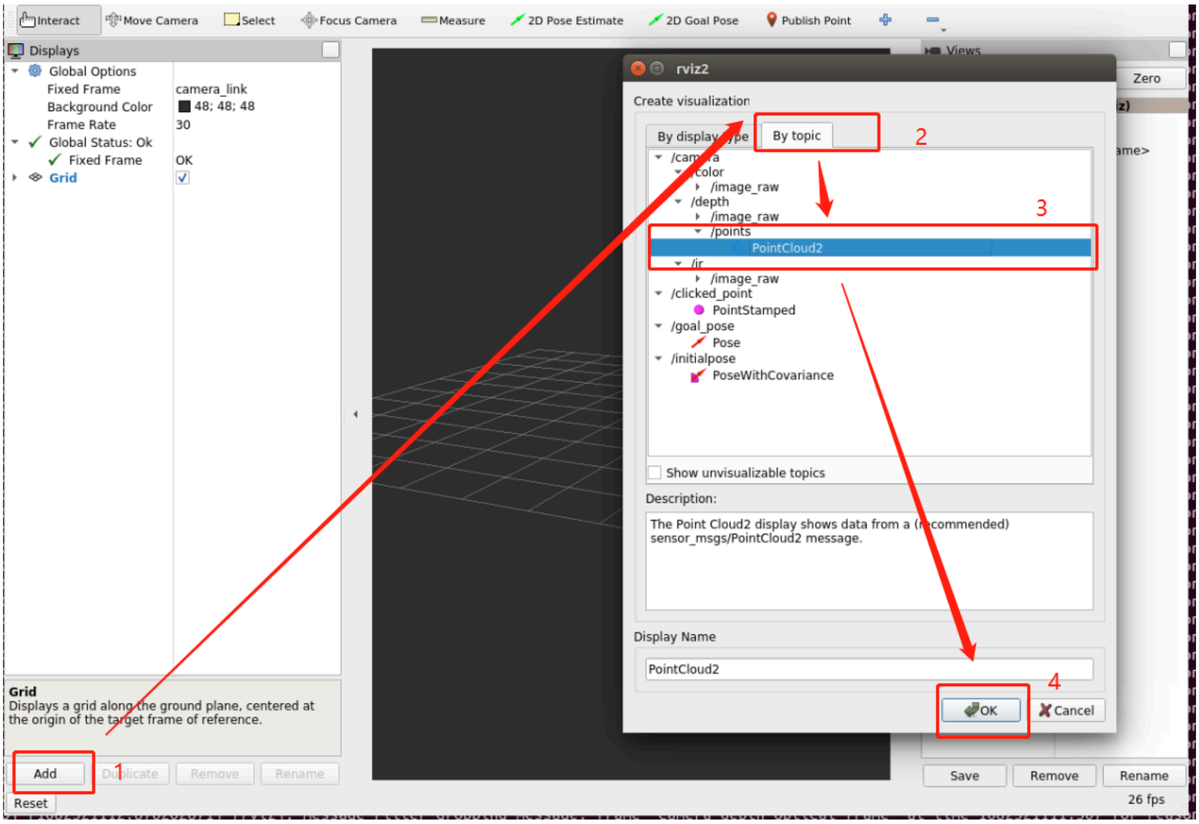
Then select the corresponding image topic to be displayed in the upper left corner, taking displaying RGB images as an example,


rqt_image_view__ImageView - rqt
Image View
/camera/color/image_raw
0
10.00m
/camera/color/image_raw_mouse_left
Smooth scaling
0°

Use rviz2 to display deep point clouds, input in Docker terminal,

```
rviz2
```

Then add deep point cloud information in rviz,



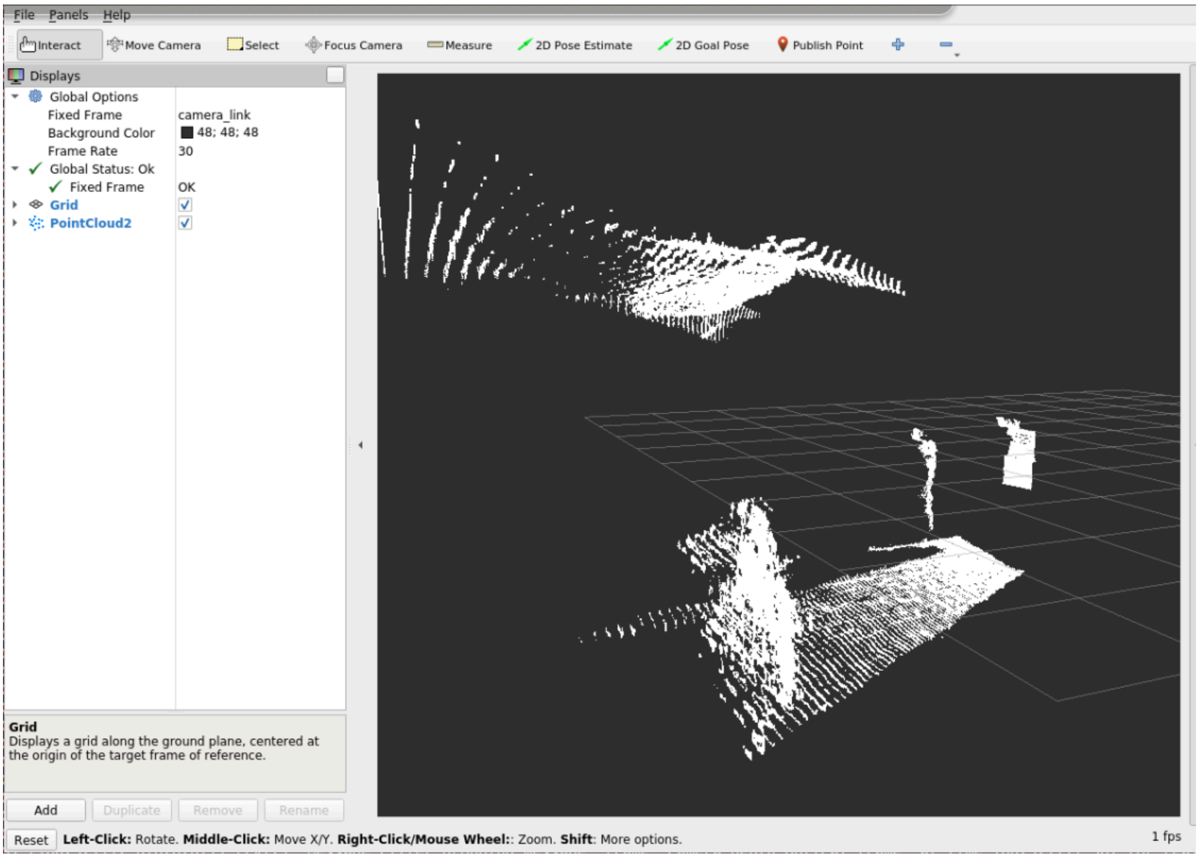
Global Options
Fixed Frame camera_link
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Fixed Frame OK
Grid ☑

Description:
The Point Cloud2 display shows data from a (recommended) sensor_msgs/PointCloud2 message.

Display Name
PointCloud2

Grid
Displays a grid along the ground plane, centered at the origin of the target frame of reference.

Finally, modify the 【 Fixed Frame】 to 【camera_ Link 】 to view point cloud information,



Global Options
Fixed Frame camera_link
Background Color 48; 48; 48
Frame Rate 30
Global Status: Ok
Fixed Frame OK
Grid ☑
PointCloud2 ☑

Grid
Displays a grid along the ground plane, centered at the origin of the target frame of reference.

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

If only depth information is needed, Astrapro and Astraproplus only need to start the following command,

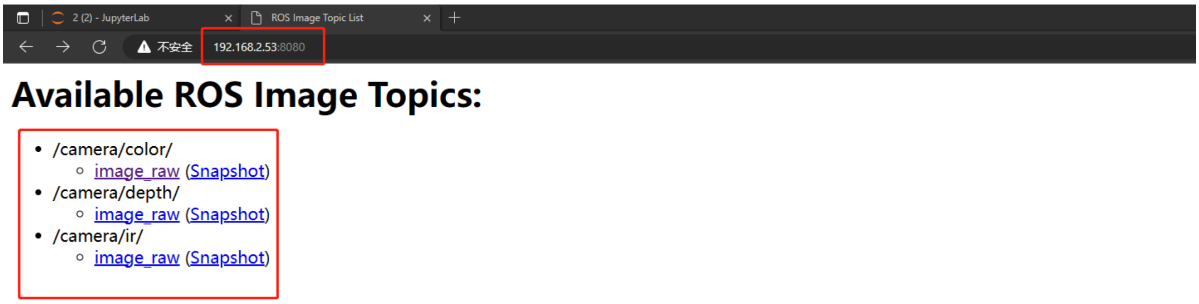
```
ros2 launch astra_camera astra.launch.xml
```

Viewing image data on web pages,

```
ros2 run web_video_server web_video_server
```

Then open the browser, and the computer and host network must be on the same local area network. Enter the URL: your host IP+8080, for example, my host network IP is 192.168.2.53, and my dock container also uses the host network. Therefore, the network IPs of the two are the same, so enter the URL in the browser:

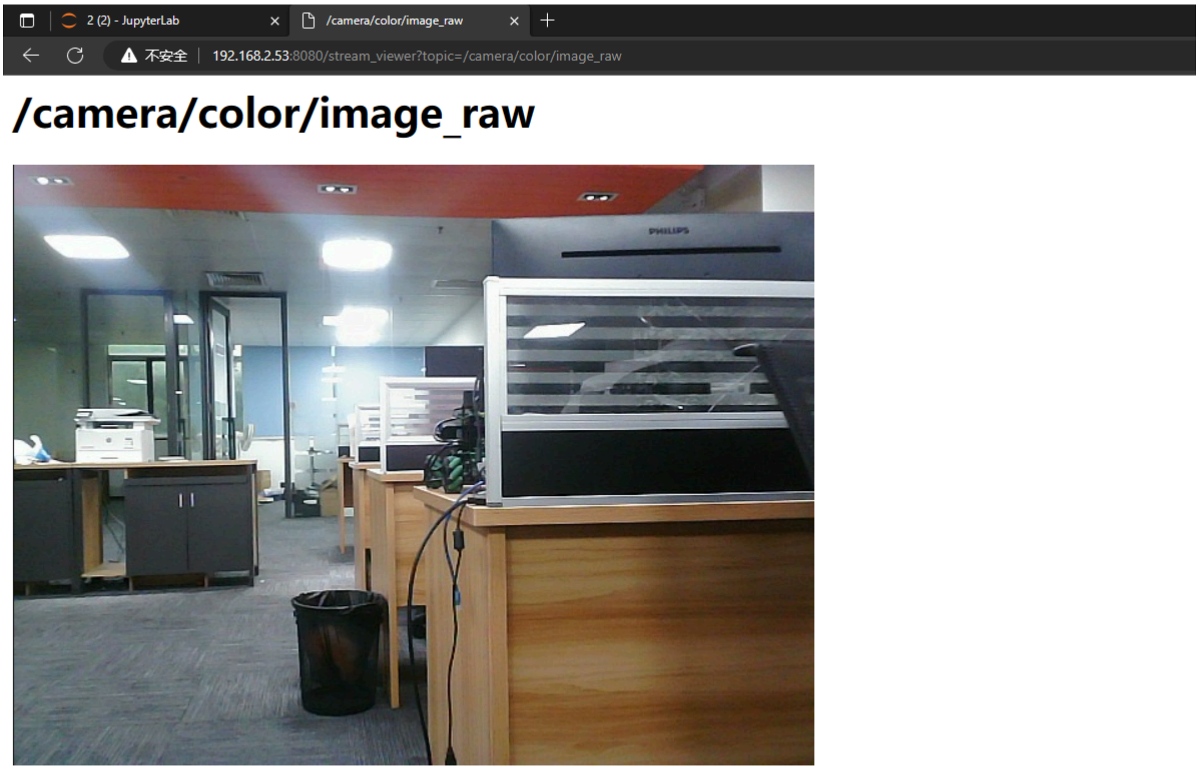
```
http://192.168.2.53:8080/
```



Available ROS Image Topics:

- /camera/color/
  - image_raw (Snapshot)
- /camera/depth/
  - image_raw (Snapshot)
- /camera/ir/
  - image_raw (Snapshot)

Then, select one of the image topic data, such as displaying an rgb image, and select "image_raw" under "/camera/color/".


2 (2) - JupyterLab
/camera/color/image_raw
不安全
192.168.2.53:8080/stream_viewer?topic=/camera/color/image_raw
PHILIPS

/camera/color/image_raw