

# Running AI Applications Snap on RZ/V2H with Ubuntu 26.04

Prepared by: Devrim Ayyildiz

Reviewed by:

Prepared on: 19 Jun 2026

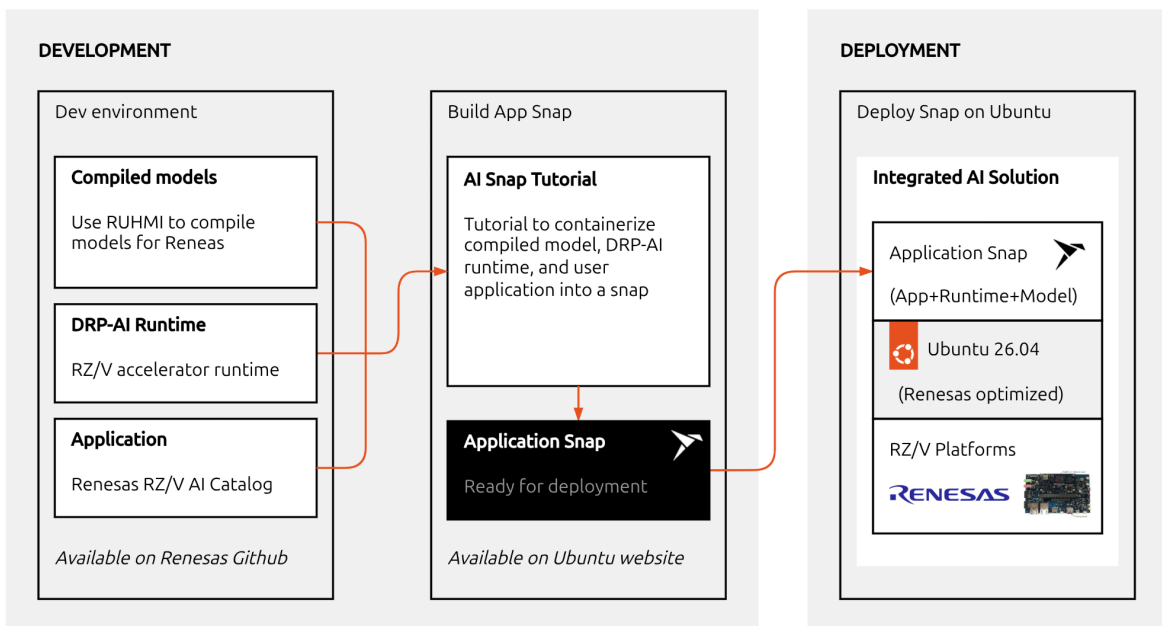
Reviewed on:

Version: 1.0

- [1. Purpose](#)
- [2. Prerequisites](#)
- [3. General Information](#)
- [4. Running the AI Application Snap](#)
  - [Step 1: Install Required Snaps](#)
  - [Step 2: Run Ubuntu Frame](#)
  - [Step 3: Execute the Applications](#)
- [5. Building the AI Application Snap \(optional\)](#)

# 1. Purpose

This document describes how to install and run Renesas AI application snap on the RZ/V2H board. We also demonstrate how the snap can also be built locally via snapcraft.



# 2. Prerequisites

To get started, RZ/V2H board should be provisioned with the Ubuntu image that supports this platform. For setup instructions, refer to the Quick Start Guide available at <https://ubuntu.com/download/renesas-iot>.

In this tutorial we have provisioned the RZ/V2H with an **Ubuntu Server 26.04** image.

To run the snap on the RZ/V2H board, you need:

- Ubuntu 26.04 LTS Server image installed
- An HDMI monitor connected to the RZ/V2H
- A USB camera connected to the RZ/V2H
- Serial connection (UART) to run snap commands
- Ethernet connection to internet (to install Ubuntu Frame and AI Application snap)

## 3. General Information

The AI application snap is based on the AI applications presented in Renesas RZ/V AI Applications repository: [https://github.com/renesas-rz/rzv\\_ai\\_sdk](https://github.com/renesas-rz/rzv_ai_sdk).

The snap source used in this guide is in: [https://github.com/canonical/rzv\\_ai\\_applications](https://github.com/canonical/rzv_ai_applications)

For demonstration purposes, we provide the AI Application Snap: **rzv-ai-applications**. It contains 2 applications:

- **object-tracker**
- **object-counter**

The object-tracker application tracks and counts the number of people entering a designated boundary line, while simultaneously reducing the count for those exiting the boundary line. Details of the application can be found [here](#).

The object-counter counts different types of objects: vehicles (typically for traffic monitoring), animals (for zoo or farm monitoring) and general objects. Therefore it contains three different models. Details of the application can be found [here](#).

The snap is published in the Snap Store: <https://snapcraft.io/rzv-ai-applications>. Therefore we will simply install the snap directly from the store to the RZ/V2H via network connection.

## 4. Running the AI Application Snap

### Step 1: Install Required Snaps

On the RZ/V2H board, install Ubuntu Frame, which serves as a reliable and secure display server for embedded Linux devices.

```
Shell
```

```
$ sudo snap install ubuntu-frame
```

Then, install the AI Applications snap:

```
Shell
```

```
$ sudo snap install --devmode rzv-ai-applications --channel=beta
```

## Step 2: Run Ubuntu Frame

Run Ubuntu Frame on the device. You will notice the Ubuntu Frame interface displayed on your HDMI-connected monitor.

```
Shell
```

```
$ ubuntu-frame &
```

## Step 3: Execute the Applications

You can now run the desired AI applications. The command arguments are identical to those defined in the upstream AI applications provided in Renesas [rzv-ai-sdk](#).

```
Shell
```

```
$ rzv-ai-applications.object-tracker USB
```

```
$ rzv-ai-applications.object-counter <COCO|animal|vehicle> USB
```

## 5. Building the AI Application Snap (optional)

In case you want to build the snap locally, it only takes cloning the rzv-ai-applications repository and build the snap with the snapcraft tool:

In case snapcraft is not installed on your system, you can install it via:

Shell

```
$ snap install snapcraft --classic
```

Refer to [Snapcraft documentation](#) for details. Once snapcraft is installed, clone the repository and build the snap:

Shell

```
$ git clone git@github.com:canonical/rzv\_ai\_applications.git
$ cd rzv_ai_applications
$ snapcraft pack
```

This will create the snap file **rzv-ai-applications\_1.0\_arm64.snap** locally. Refer to snapcraft.yaml file on how the snap build configuration is set up. There is excellent [documentation](#) and [tutorials](#) on how snapcraft works and how to build your own snaps.