

Edge Insights for Vision

Contents

Edge Insights for Vision.. 3

How it Works .. 4

Get Started Guide using Linux*.. 6

Linux* Requirements	6
Install Edge Insights for Vision	7
Introduction to the Edge Software Command Line Interface (CLI)	11
Linux* Troubleshooting	17
Linux* Tutorials.....	17
Single and Multi-Object Detection with Hardware Acceleration on Linux*	18
Single and Multi-Object Detection on CPU on Linux*	18
Single and Multi-Object Detection on GPU on Linux*	20
Single and Multi-Object Detection on Intel® Vision Accelerator on Linux*	23
Multi-Camera Detection of Social Distancing on Linux*	26

Release Notes .. 33

Documentation Archive.. 34

Edge Insights for Vision

Edge Insights for Vision from Intel features a set of pre-integrated ingredients designed for computer vision and deep-learning inference for applications at the edge, optimized for Intel® architecture. Implement as a containerized architecture or a stand-alone runtime.

Edge Insights for Vision helps to address various monitoring usages for healthcare, worker safety, and retail, using deep learning models, facial recognition, and analytics, see [How it Works](#).

For operating system-specific installation instructions, refer to:

- [Get Started Guide using Linux*](#). Also includes an introduction to the Edge Software command line interface to learn how to manage Intel® Developer Catalog packages.

When set up is complete, choose one of the tutorials for step-by-step, hands-on walkthroughs of how to use and configure modules in Edge Insights for Vision:

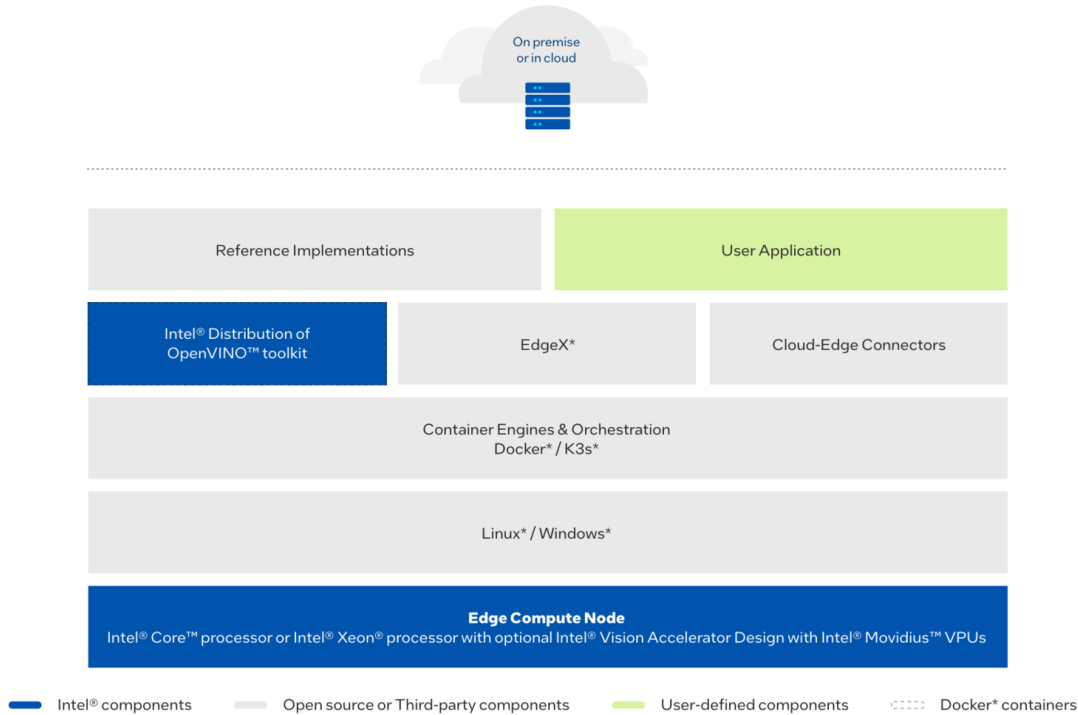
- Linux: [Single and Multi-Object Detection with Hardware Acceleration on Linux*](#)
- Linux: [Multi-Camera Detection of Social Distancing on Linux*](#)

As a next step, you can explore other [use cases and reference implementations](#) on the Edge Insights for Vision page.

How it Works

Edge Insights for Vision is a set of pre-validated modules, implemented as a containerized architecture or a stand-alone runtime, for deploying computer vision and deep learning workloads at the edge. The package features third-party modules for orchestration and cloud support, and the [Intel® Distribution of OpenVINO™ toolkit](#) for computer vision and deep learning apps optimized for Intel® architecture.

This section provides an overview of the modules and services featured with Edge Insights for Vision.



Modules and Services

The Intel® Distribution of OpenVINO™ toolkit (OpenVINO™)

Edge Insights for Vision includes the [Intel® Distribution of OpenVINO™ toolkit](#) so you can develop and optimize AI and computer vision applications. OpenVINO™ maximizes performance and extends workloads across Intel hardware, including accelerators.

Edge Insights for Vision supports the following OpenVINO™ packages to optimize inferencing on your edge IoT device:

- Intel® Distribution of OpenVINO™ toolkit in a Container builds Docker* images for the OpenVINO™ toolkit.
- Intel® Distribution of OpenVINO™ toolkit Runtime installs the toolkit runtime packages distributed through the APT repository. It does not run in a container.

Refer to the [Release Notes](#) for the supported version of Intel® Distribution of OpenVINO™ toolkit.

Container Engines and Orchestration

Edge Insights for Vision features [Docker*](#) and [Rancher k3s*](#) for automated container management.

- [Docker*](#) is a container framework widely used in enterprise environments. It allows applications and their dependencies to be packaged together and run as a self-contained unit.

- Rancher k3s* is a lightweight [Kubernetes*](#) distribution that provides automated container management for IoT and edge computing.

NOTE It is observed that Rancher k3s features may not work for PRC users.

Cloud Edge Connectors

Edge Insights for Vision features [Amazon Web Services \(AWS\)* Greengrass Prerequisites](#) and to connect edge devices with the cloud and each other.

Amazon Web Services (AWS)* Greengrass Prerequisites

AWS* Greengrass Prerequisites installs the following packages to enable local data collection, analysis, and communication for your AWS-connected edge devices:

- AWS* IoT Greengrass Core.
- AWS* IoT Device SDK.
- AWS* IoT Greengrass SDK.

Intel Tools

- [Intel® Edge Software Configurator](#) is a software tool that enables you to configure and optimize* Edge Software Hub Packages and custom applications, creating and managing Containers and Virtual Machines for edge computing with the same flexibility you would have in cloud computing operations.

Get Started Guide using Linux*

This step-by-step guide takes you through installing the Edge Insights for Vision on Linux and introduces you to the Edge Software command line interface from which you manage the Intel® Developer Catalog packages. After you complete this guide you will be ready to use a tutorial.

Refer to the [Linux* Requirements](#) section before you get started with installation.

To use these instructions, you must download the [Edge Insights for Vision package](#). The download file name is `edge_insights_vision.zip`

Important

Save the email message you get when you download the package. **Keep this message safe!** This message includes a product key that is required to complete the installation. If you do not get the email message, use the [Support Forum](#).

Refer to [Modules and Services](#) for the complete list of supported modules. At download time, you can choose the recommended configuration or you can customize your download by selecting modules.

Edge Insights for Vision is delivered as compressed `.zip` file that is compatible with the operating system you selected during the download. The `.zip` contains a binary executable file, a manifest file that lists the modules that will be installed, and a readme file.

NOTE

For PRC Users: We have observed that some modules, such as k3s, cannot be successfully downloaded under PRC network. The installation of those modules will be skipped automatically if PRC network is detected.

See [Linux* Troubleshooting](#) if you run into problems installing the software.

After installation, follow the [Introduction to the Edge Software Command Line Interface \(CLI\)](#). This introduces you to the Edge Software command line interface from which you manage the Intel Edge Software Hub packages.

Next, you will be ready to run a tutorial. Get started in your learning journey with [Single and Multi-Object Detection with Hardware Acceleration on Linux*](#) and [Multi-Camera Detection of Social Distancing on Linux*](#).

Linux* Requirements

In addition to the [Edge Insights for Vision package](#), you must have the following:

Target System

- One of the following processors:
 - Intel® Pentium® processor N4200/5, N3350/5, N3450/5 with Intel® HD Graphics.
 - Intel Atom® processor with Intel® SSE4.1 support.
 - 6th - 12th generation Intel® Core™ processors.
 - Intel® Xeon® processor E3, E5, and E7 family.
 - 2nd Generation Intel® Xeon® Scalable Processors.
 - 3rd Generation Intel® Xeon® Scalable Processors.
- At least 8 GB RAM.

- At least 64 GB hard drive.
- An Internet connection.
- Ubuntu* 20.04.4 LTS†

† If Kernel version is less than 5.17.15, then the kernel will be upgraded to 5.17.15 for 11th generation Intel® Core™ processors and 12th generation Intel® Core™ processors during the installation of the package.

Knowledge/Experience

- You are familiar with executing Linux* commands.

Install Edge Insights for Vision

During the installation, you will be prompted to enter your product key. This key is in the email you received from the Intel® Registration Center. Contact [Support Forum](#) if you do not have this email message.

The steps below explain how to:

- Download the Edge Insights for Vision package.
- Prepare your target system.
- Copy the package.
- Complete the installation steps.

Step 1: Download the Package

Download the [Edge Insights for Vision package](#) by selecting the Linux* operating system in the Target System OS selector. The download file name is `edge_insights_vision.zip`

Important

Save the email message you get when you download the package. **Keep this message safe!** This message includes a product key that is required to complete the installation. If you do not get the email message, use the [Support Forum](#).

Step 2: Prepare the Target System

Make sure your target system has a fresh Operating System installation that corresponds to the version of Edge Insights for Vision that you downloaded. If you need help installing Ubuntu*, follow these steps:

1. Download the following file to your developer workstation:
 - Select the file named `ubuntu-20.04.4-desktop-amd64.iso` at this link: [Ubuntu* v20.04.4 Desktop ISO file](#)
2. Create a bootable flash drive using an imaging application, such as Startup Disk Creator, available on target system.
3. After flashing the USB drive, power off your target system, insert the USB drive, and power on the target system.

If the target system doesn't boot from the USB drive, change the boot priority in the system BIOS.
4. Follow the prompts to install the OS with the default configurations. For detailed instructions, see the [Ubuntu guide](#).
5. Power down your target system and remove the USB drive.
6. Power up the target system. You will see Ubuntu is successfully installed.

If the kernel version is less than 5.17.15, then the kernel will be upgraded to 5.17.15 for 11th generation Intel® Core™ processors and 12th generation Intel® Core™ processors during the installation of the package.

For older generations of Intel® processors, you will be prompted asking whether to upgrade the kernel to 5.17.15. If you select **Yes|Y**, then the kernel will be upgraded to 5.17.15.

NOTE

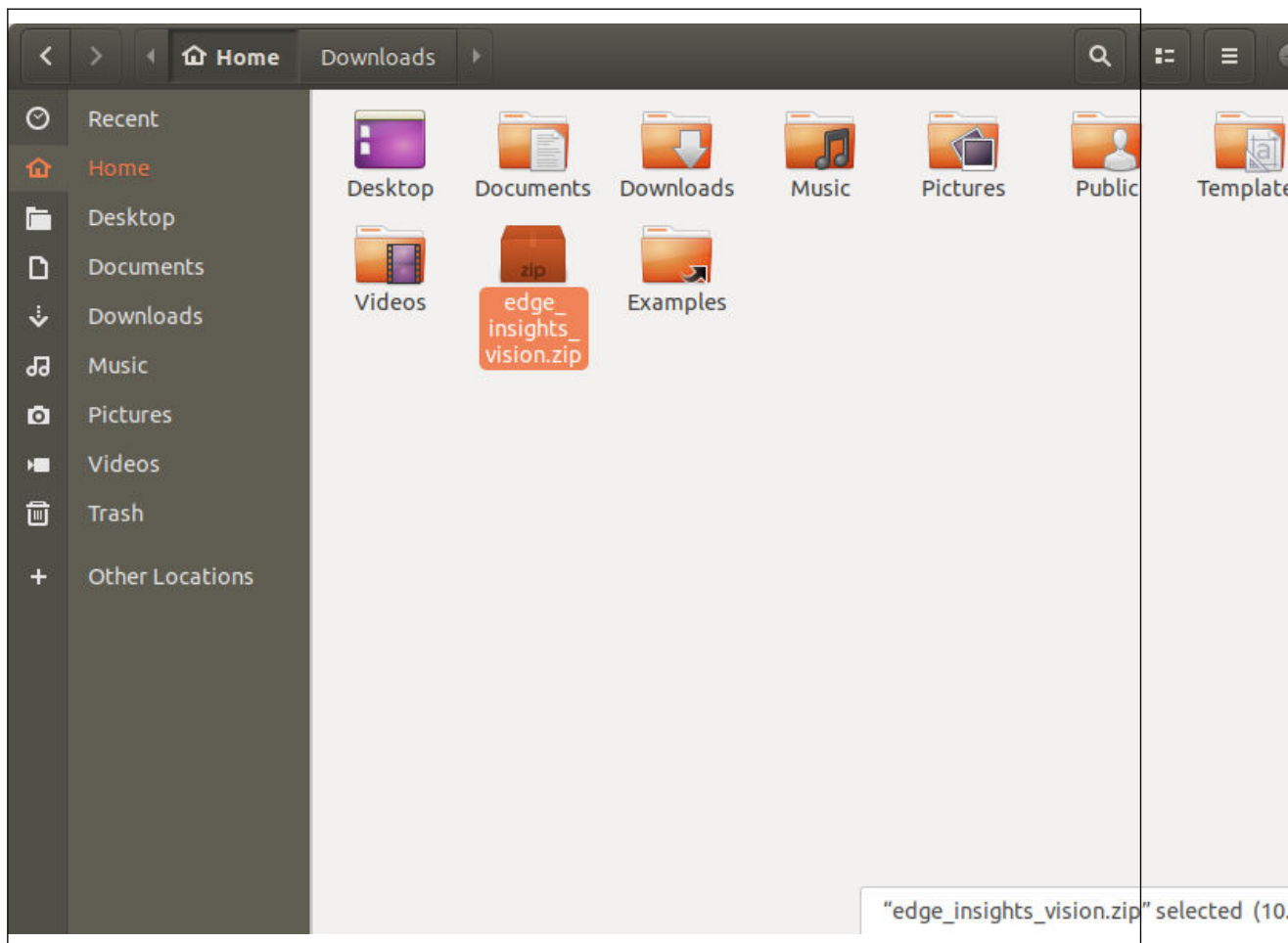
Rebooting of the system is mandatory to work with Intel® GPU.

Intel® Vision Accelerator (HDDL Accelerator) does not work on kernel 5.17.15. If you are using the Intel® Vision Accelerator (HDDL Accelerator), then you must use the kernel version ≤ 5.15 .

Step 3: Copy the Edge Insights for Vision .zip File to the Target System

In this step, you copy Edge Insights for Vision to your target system.

1. Copy `edge_insights_vision.zip` from the developer workstation to the Home directory on your target system. You can use a USB flash drive to copy the file. The icon looks like this:



Step 4: Extract the Edge Insights for Vision Software

In this step you extract `edge_insights_vision.zip`. You need to be on the target system to complete these steps.

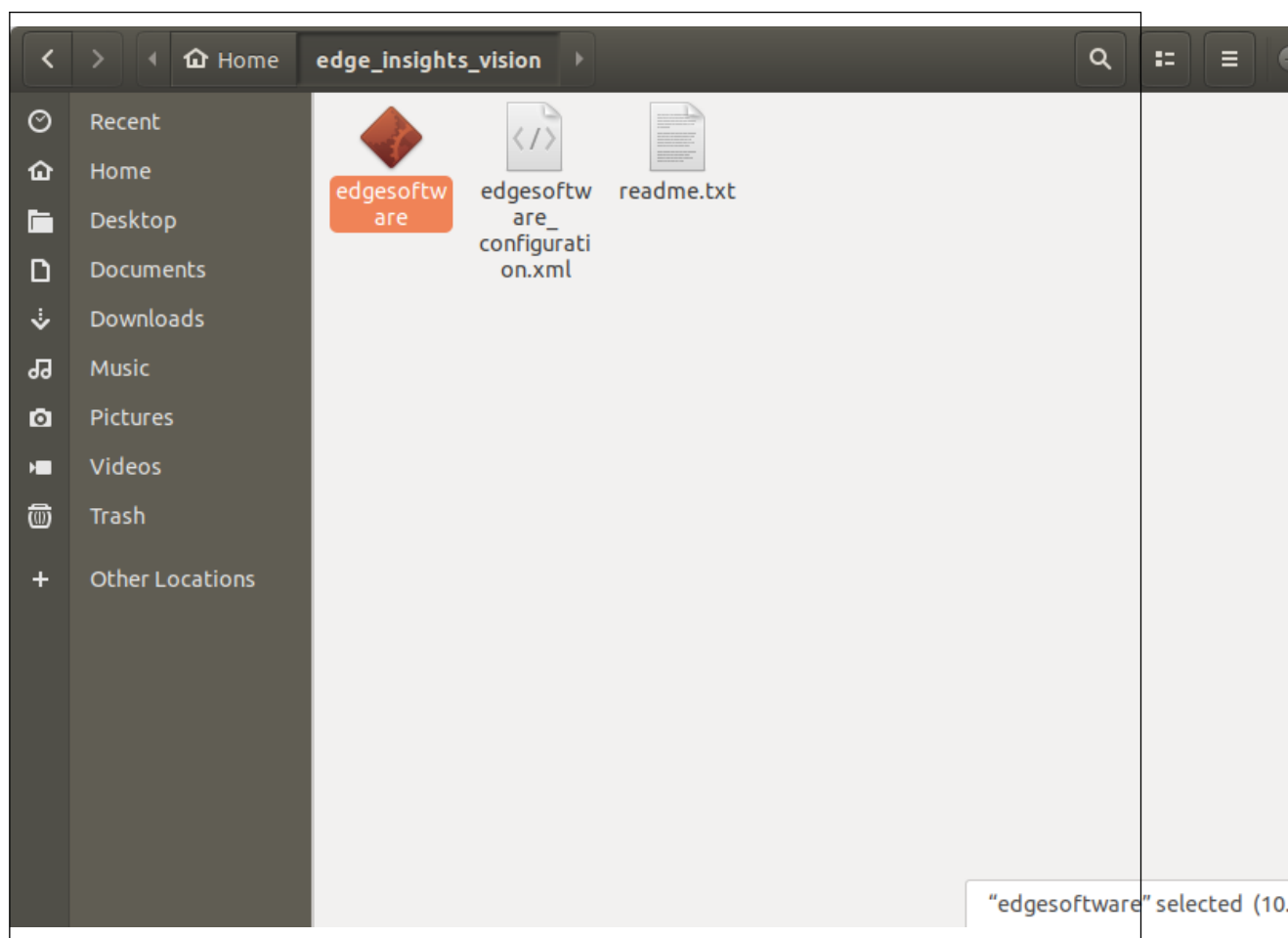
1. Make sure you have a working Internet connection.

2. Open a new terminal.
3. Extract the package:

```
unzip edge_insights_vision.zip
```

NOTE

If you download the file more than once, each download beyond the first is appended by a number.



4. (Optional) Use the link in the `readme` file to open this Get Started Guide on the target system for an easy way to copy and paste commands.

Step 5: Install the Edge Insights for Vision Software

You will now run a script that will download components and install Edge Insights for Vision on your target system.

The software installation will take 1 to 2 hours. The completion time depends on your target system and Internet connection.

NOTE During the installation, pip and apt packages, as well as code from Git repositories, is downloaded from the Internet. If you are in a region that has access restrictions, you will be prompted to provide alternate download URLs.

1. Run these commands:

```
cd edge_insights_vision/  
chmod 775 edgesoftware  
./edgesoftware install
```

2. Type the product key at the prompt:

```
intel@edgesoftware: ~/edge_insights_vision  
File Edit View Search Terminal Help  
intel@edgesoftware:~/edge_insights_vision$ ./edgesoftware install  
Please enter the Product Key. The Product Key is contained in the email you received from Intel confirming your download:  
  
Starting the setup...  
ESB CLI version: 2020.1  
-----SOFTWARE INFO-----  
OS version: Ubuntu 18.04 LTS  
Kernel: 4.15.0-20-generic  
  
-----HARDWARE INFO-----  
Hardware Architecture: x86_64  
Processor: Intel(R) Xeon(R) CPU E3-1268L v5 @ 2.40GHz  
Memory size: 31G  
  
-----HARDWARE ACCELERATOR-----  
Found 9 Intel® Movidius™ Visual Processing Unit (VPU)  
Checking Internet connection  
Connected to the Internet  
[sudo] password for intel:  
Checking for prerequisites  
All dependencies met  
Validating product key  
Successfully validated Product Key  
Starting installation  
Downloading modules...  
Downloading component esb_common  
Module validation passed for 5e7d7eb975e5f9002bf5016b  
Unzipping the module esb_common...  
Downloading component Amazon Web Services Greengrass Prerequisites  
Module validation passed for 5e68f908f3b390001f4a9816  
Unzipping the module Amazon Web Services Greengrass Prerequisites...  
Downloading component Azure Edge Prerequisites  
Module validation passed for 5e68f8d3f3b390001f4a9815  
Unzipping the module Azure Edge Prerequisites...  
Downloading component K3s lightweight Kubernetes  
Module validation passed for 5e68f9c5f3b390001f4a9817  
Unzipping the module K3s lightweight Kubernetes...
```

Based on components selected and system configuration, you might be prompted for additional actions. For example, if your system is behind a proxy, you are asked to enter proxy settings.

If a People's Republic of China (PRC) network is detected, then the message below is displayed and asks for an alternate mirror URL.

```
Connected to a network in China. Module availability is restricted in your region.  
For a successful installation, enter the URL for a local mirror site for pip and apt package  
managers  
Do you want to override settings? Enter Yes or No:
```

If you are connected to a PRC network, then enter **Yes** for overriding the settings and enter a local mirror site as shown in the example below:

```
Please enter the URL for a local mirror site for pip and apt package managers: mirrors.aliyun.com  
Modifying settings
```

When the installation is complete, you see the message **Installation of package complete** and the installation status for each module.

```

Stopping OpenVINO Containers [.....] 100%
Building Docker Image. This may take up to half an hour...
Building docker image [.....] 100%
[sudo] password for intel:
Successfully installed Intel Distribution of OpenVINO toolkit 2019_R3 in a Container took 30 minutes 14.82 seconds
Installing Intel Vision Accelerator Design Slicer
Installing HDL Allocator Configuration [.....] 100%

Successfully installed Intel_Vision_Accelerator_Design_Slicer took 0.63 seconds
Installing ONNX Runtime
Installing Dependencies [.....] 100%
Installing Cmake. This step may take approximately 5 mins.
Installing Cmake [.....] 100%
Installing Protobuf [.....] 100%
Installing Onnx Prerequisites [.....] 100%
Downloading Onnxruntime. This step may take approximately 5 mins.
Downloading Onnxruntime [.....] 100%
Building Onnxruntime. This step may take upto half an hour.
Building Onnxruntime [.....] 100%
Installing Wheel [.....] 100%
Downloading Mobilenet [.....] 100%
Running Onnxruntime unit test [.....] 100%
Running Onnxruntime perf test [.....] 100%
Running Onnxruntime accuracy test [.....] 100%
Running Onnxruntime python test [.....] 100%
Successfully installed ONNX Runtime took 25 minutes 13.76 seconds
Installation of package complete
+-----+-----+
| Module | Status |
+-----+-----+
| Amazon Web Services Greengrass Prerequisites | SUCCESS |
| Azure Edge Prerequisites | SUCCESS |
| K3s lightweight Kubernetes | SUCCESS |
| Intel Distribution of OpenVINO toolkit 2020_R3_Runtime | SUCCESS |
| Docker Community Edition CE | SUCCESS |
| Intel Distribution of OpenVINO toolkit 2020_R3 in a Container | SUCCESS |
| Intel_Vision_Accelerator_Design_Slicer | SUCCESS |
| ONNX_Runtime | SUCCESS |
+-----+-----+

```

NOTE

You must reboot your system to work with Intel® GPU.

To confirm your installation was successful and for a quick look at how it works, use the [Introduction to the Edge Software Command Line Interface](#).

Introduction to the Edge Software Command Line Interface (CLI)

edgesoftware is a command line interface (CLI) that helps you manage packages on the Intel® Developer Catalog.

This guide describes the CLI commands and their usage. In this guide you will:

- Try out commands and get familiar with the CLI and the package you installed.
- Learn to update modules.
- Learn to install custom components.
- Learn to export the package you installed, including custom modules, so you can install it on other Edge Nodes.

Get Started with the edgesoftware CLI

Use the information in this section to try out the edgesoftware CLI commands.

To begin:

1. Open a terminal window.

2. Go to the `edge_insights_vision/` directory.
3. Try out the following commands.

Get Help or List the Available Commands

- Command:

```
./edgesoftware --help
```

- Response:

```
Usage: edgesoftware [OPTIONS] COMMAND [ARGS]...
A CLI wrapper for management of Intel® Edge Software Hub packages

Options:
  -v, --version  Show the version number and exit.
  --help        Show this message and exit.

Commands:
  download  Download modules of a package.
  export    Exports the modules of a package.
  install   Install modules of a package.
  list      List the modules of a package.
  log       Show log of CLI events.
  pull      Pull Docker image.
  uninstall Uninstall the modules of a package.
  update    Update the modules of a package.
  upgrade   Upgrade a package.
```

View the Software Version

- Command:

```
./edgesoftware --version
```

- Response: The edgesoftware version, build date, and target OS.

List the Package Modules

- Command:

```
./edgesoftware list
```

- Response: The modules installed and status.

```
intel@edge-software:~/Downloads/edge_insights_vision$ ./edgesoftware list
```

Module	Status
Amazon_Web_Services_Greengrass_Prerequisites	SUCCESS
K3s_Lightweight_Kubernetes	SUCCESS
Azure_Edge_Prerequisites	SUCCESS
Intel_Distribution_of_OpenVINO_toolkit_2020_R3_Runtime	SUCCESS
Docker_Community_Edition_CE	SUCCESS
Intel_Distribution_of_OpenVINO_toolkit_2020_R3_in_a_Container	SUCCESS
Intel_Vision_Accelerator_Design_Slicer	SUCCESS
ONNX_Runtime	SUCCESS

```
intel@edge-software:~/Downloads/edge_insights_vision$
```

List Modules Available for Download

- Command:

```
./edgesoftware list --default
```

- Response: All modules available for download for that package version, modules ID and version.

```
intel@edgesoftware:~/Downloads/edge_insights_vision$ ./edgesoftware list --default
```

Modules in the recommended configuration for 5ee8ff042e0a27002a53dbb1

ID	Module	Version
5e68f908f3b390001f4a9816	Amazon Web Services* Greengrass Prerequisites	1.10
5e68f8d3f3b390001f4a9815	Azure Edge* Prerequisites	1.0.9
5e68f9c5f3b390001f4a9817	K3s* (Lightweight Kubernetes*)	0.9.0
5ee8fbc9fela62002ac4d512	Intel® Distribution of OpenVINO™ toolkit 2020.3 LTS Runtime	2020.
5e68f9f6f3b390001f4a9818	Docker Community Edition (CE)*	19.03.
5ee8fb75fela62002ac4d511	Intel® Distribution of OpenVINO™ toolkit 2020.3 LTS in a Container	2020.
5ee8fcddfela62002ac4d517	Intel® Vision Accelerator Design Slicer	2020.
5ee8fc81fela62002ac4d515	ONNX* Runtime	1.3.0

```
intel@edgesoftware:~/Downloads/edge_insights_vision$
```

Display the CLI Event Log

- Command:

```
./edgesoftware log
```

- Response: CLI event log information, such as:
 - target system information (hardware and software)
 - system health
 - installation status

- modules you can install

```

intel@edgesoftware:~/edge_insights_vision$ ./edgesoftware log
=====Start of installer log=====
Fri May 08 03:37:36 IST 2020 - INFO - ESB CLI version: 2020.1
Fri May 08 03:37:36 IST 2020 - INFO - -----SOFTWARE INFO-----
Fri May 08 03:37:36 IST 2020 - INFO - OS version: Ubuntu 18.04 LTS
Fri May 08 03:37:36 IST 2020 - INFO - Kernel: 4.15.0-20-generic

Fri May 08 03:37:36 IST 2020 - INFO - -----HARDWARE INFO-----
Fri May 08 03:37:36 IST 2020 - INFO - Hardware Architecture: x86_64
Fri May 08 03:37:36 IST 2020 - INFO - Processor: Intel(R) Xeon(R) CPU E3-1268L v5 @ 2.40GHz
Fri May 08 03:37:36 IST 2020 - INFO - Memory size: 31G

Fri May 08 03:37:36 IST 2020 - INFO - -----HARDWARE ACCELERATOR-----
Fri May 08 03:37:36 IST 2020 - INFO - Found 9 Intel® Movidius™ Visual Processing Unit (VPU)
Fri May 08 03:37:36 IST 2020 - INFO - Checking Internet connection
Fri May 08 03:37:38 IST 2020 - INFO - Connected to the Internet
Fri May 08 03:37:41 IST 2020 - INFO - Checking for prerequisites
Fri May 08 03:37:41 IST 2020 - INFO - Installing prerequisites. This may take some time...
Fri May 08 03:37:43 IST 2020 - ERROR - Failed to install prerequisites: None
Fri May 08 03:42:57 IST 2020 - INFO - ESB CLI version: 2020.1
Fri May 08 03:42:57 IST 2020 - INFO - -----SOFTWARE INFO-----
Fri May 08 03:42:57 IST 2020 - INFO - OS version: Ubuntu 18.04 LTS
Fri May 08 03:42:57 IST 2020 - INFO - Kernel: 4.15.0-20-generic

Fri May 08 03:42:57 IST 2020 - INFO - -----HARDWARE INFO-----
Fri May 08 03:42:57 IST 2020 - INFO - Hardware Architecture: x86_64
Fri May 08 03:42:57 IST 2020 - INFO - Processor: Intel(R) Xeon(R) CPU E3-1268L v5 @ 2.40GHz
Fri May 08 03:42:57 IST 2020 - INFO - Memory size: 31G

Fri May 08 03:42:57 IST 2020 - INFO - -----HARDWARE ACCELERATOR-----
Fri May 08 03:42:57 IST 2020 - INFO - Found 9 Intel® Movidius™ Visual Processing Unit (VPU)
Fri May 08 03:42:57 IST 2020 - INFO - Checking Internet connection
Fri May 08 03:42:58 IST 2020 - INFO - Connected to the Internet
Fri May 08 03:43:08 IST 2020 - INFO - Checking for prerequisites
Fri May 08 03:43:08 IST 2020 - INFO - Installing prerequisites. This may take some time...
Fri May 08 03:46:38 IST 2020 - INFO - -----Successfully installed prerequisites-----
Fri May 08 03:46:38 IST 2020 - INFO - All dependencies met
Fri May 08 03:46:38 IST 2020 - INFO - Validating product key
Fri May 08 03:46:40 IST 2020 - INFO - Successfully validated Product Key
Fri May 08 03:46:40 IST 2020 - INFO - Starting installation

```

See the Installation Event Log for a Module

- Command:

```
./edgesoftware log <MODULE_NAME>
```

You can specify multiple <MODULE_NAME> arguments by listing them with a space between each.

NOTE To find the module names, use

```
./edgesoftware list
```

- Response: The installation log for the module.

```

intel@edgesoftware: ~/edge_insights_vision
File Edit View Search Terminal Help

intel@edgesoftware:~/edge_insights_vision$ ./edgesoftware log Intel Distribution of OpenVINO toolkit 2019_R3_Runtime
=====Start of log for module Intel Distribution of OpenVINO toolkit 2019_R3_Runtime=====
Fri May 08 12:11:30 IST 2020 - INFO - Starting main installation openvino_runtime
Fri May 08 12:11:30 IST 2020 - INFO - -----
Fri May 08 12:11:30 IST 2020 - INFO - Starting installation openvino_runtime
Fri May 08 12:11:30 IST 2020 - INFO - -----
Fri May 08 12:11:30 IST 2020 - INFO - Installing OpenVINO
Fri May 08 12:11:41 IST 2020 - INFO - Installing dependencies
Fri May 08 12:11:41 IST 2020 - INFO - intel-openvino-runtime-ubuntu18-2019.3.344 installing package
Fri May 08 12:13:52 IST 2020 - INFO - Installed the following package(s): intel-openvino-runtime-ubuntu18-2019.3.344
Fri May 08 12:13:52 IST 2020 - INFO - Dependencies installed
Fri May 08 12:13:52 IST 2020 - INFO - Installing dependencies
Fri May 08 12:13:52 IST 2020 - INFO - intel-openvino-model-optimizer-2019.3.344 installing package
Fri May 08 12:13:58 IST 2020 - INFO - Installed the following package(s): intel-openvino-model-optimizer-2019.3.344
Fri May 08 12:13:58 IST 2020 - INFO - Dependencies installed
Fri May 08 12:13:58 IST 2020 - INFO - Installing dependencies
Fri May 08 12:13:58 IST 2020 - INFO - intel-openvino-omz-dev-2019.3.344 installing package
Fri May 08 12:14:09 IST 2020 - INFO - Installed the following package(s): intel-openvino-omz-dev-2019.3.344
Fri May 08 12:14:09 IST 2020 - INFO - Dependencies installed
Fri May 08 12:14:09 IST 2020 - INFO - Installing dependencies
Fri May 08 12:14:09 IST 2020 - INFO - intel-openvino-ie-samples-2019.3.344 installing package
Fri May 08 12:14:15 IST 2020 - INFO - Installed the following package(s): intel-openvino-ie-samples-2019.3.344
Fri May 08 12:14:15 IST 2020 - INFO - Dependencies installed
Fri May 08 12:14:15 IST 2020 - INFO - Installing dependencies
Fri May 08 12:14:15 IST 2020 - INFO - intel-openvino-omz-tools-2019.3.344 installing package
Fri May 08 12:14:21 IST 2020 - INFO - Installed the following package(s): intel-openvino-omz-tools-2019.3.344
Fri May 08 12:14:21 IST 2020 - INFO - Dependencies installed
Fri May 08 12:14:21 IST 2020 - INFO - Completed installation OpenVINO toolkit
Fri May 08 12:14:21 IST 2020 - INFO - Installing OpenVINO dependencies
Fri May 08 12:14:38 IST 2020 - INFO - Installing dependencies
Fri May 08 12:18:48 IST 2020 - INFO - Completed installation OpenVINO dependencies
Fri May 08 12:18:48 IST 2020 - INFO - Installing model optimizer configurations
Fri May 08 12:19:11 IST 2020 - INFO - Completed installation model optimizer configurations
Fri May 08 12:19:11 IST 2020 - INFO - Installing NCS related rules
Fri May 08 12:19:11 IST 2020 - INFO - Installing NCS udev rules dependencies
Fri May 08 12:19:12 IST 2020 - INFO - Completed installation NCS udev rules
Fri May 08 12:19:12 IST 2020 - WARNING - Reboot the target machine for changes to take effect
Fri May 08 12:19:12 IST 2020 - INFO - Installing GPU drivers This will take some time. Please wait.
Fri May 08 12:19:12 IST 2020 - INFO - Installing drivers prerequisites
Fri May 08 12:19:12 IST 2020 - INFO - Downloading OpenVINO toolkit

```

Uninstall the Module

The edgesoftware CLI lets you uninstall individual modules.

Command:

```
./edgesoftware uninstall <Module-ID>
```

```

intel@intel-SER0:~/Downloads/ESH_Prod/vision2020.4/edge_insights_visions$ ./edgesoftware list
+-----+-----+-----+
| ID | Module | Status |
+-----+-----+-----+
| 5f213bae9e63c9002a6fd88f | Amazon Web Services Greengrass Prerequisites | SUCCESS |
| 5f21457b9e63c9002a6fd89b | Azure IoT Edge Setup | SUCCESS |
| 5f213cf59e63c9002a6fd891 | K3s lightweight Kubernetes | SUCCESS |
| 5f2867ae4fc424002ac30328 | Intel Distribution of OpenVINO toolkit 2020 4 Runtime | SUCCESS |
| 5f286b834fc424002ac3032a | Deep Learning DL Streamer Pipelines | SUCCESS |
| 5f21392e9e63c9002a6fd88d | Docker Community Edition CE | SUCCESS |
| 5f213aae9e63c9002a6fd88e | Docker Compose | SUCCESS |
| 5f2868714fc424002ac30329 | Intel Distribution of OpenVINO toolkit 2020 4 in a Container | SUCCESS |
| 5f21411d9e63c9002a6fd898 | edgex foundry | SUCCESS |
| 5f286c7a4fc424002ac3032b | ONNX Runtime | SUCCESS |
| 5f213c3b9e63c9002a6fd890 | Azure IoT Edge Prerequisites | SUCCESS |
+-----+-----+-----+
intel@intel-SER0:~/Downloads/ESH_Prod/vision2020.4/edge_insights_visions$ ./edgesoftware uninstall 5f213c3b9e63c9002a6fd890
Components to be uninstalled are :['Azure_IoT_Edge_Prerequisites']
Uninstalling Azure_IoT_Edge_Prerequisites
Uninstalling AzureIoTEdge [.....] 100%
Successfully uninstalled Azure_IoT_Edge_Prerequisites took 1 minutes 3.73 seconds
Uninstall Finished
+-----+-----+-----+
| Id | Module | Status |
+-----+-----+-----+
| 5f213c3b9e63c9002a6fd890 | Azure IoT Edge Prerequisites | SUCCESS |
+-----+-----+-----+
intel@intel-SER0:~/Downloads/ESH_Prod/vision2020.4/edge_insights_visions$

```

Uninstall the Package

The edgesoftware CLI lets you uninstall the package.

Command:

```
./edgesoftware uninstall -a
```

Upgrade the Package

The edgesoftware CLI lets you upgrade the package. To check the available package for installation, use the command:

```
upgrade /edgesoftware <PACKAGE_ID>
```

During the upgrade process, the current package is uninstalled completely, and the new package gets installed.

Command:

```
./edgesoftware list -v
./edgesoftware upgrade <PACKAGE_ID>
```



```

intel@intel-SER0:~/Downloads/ESH_Prod/vision2020.4/edge_insights_visions$ ./edgesoftware list -v
Packages recommended for '2b4f68be-47e9-45e9-9e49-e53fe59ede01'
+-----+-----+-----+
| ID | Package | Version |
+-----+-----+-----+
| 5f286fa763b939002b6d8467 | Edge Insights for Vision | 2020.4 |
| 5f2bab180c6523002a9dba1c | Edge Insights for Vision | 2020.3 |
| 5e6909df8cce53001ff92e5c | Edge Insights for Vision | 2019.3 |
+-----+-----+-----+
intel@intel-SER0:~/Downloads/ESH_Prod/vision2020.4/edge_insights_visions$ ./edgesoftware upgrade 5f286fa763b939002b6d8467
Please enter the Product Key. The Product Key is contained in the email you received from Intel confirming your download: a5
The selected package version is already installed
intel@intel-SER0:~/Downloads/ESH_Prod/vision2020.4/edge_insights_visions$

```

Summary and Next Steps

By following this guide, you tried a few commands to familiarize yourself with the features of the edgesoftware CLI.

Go to [Single and Multi-Object Detection with Hardware Acceleration on Linux*](#) to run your first inference application using Edge Insights for Vision.

Go to [Intel® Edge Software Configurator Documentation](#) to learn how to use the tool to manage Edge Software Packages, create and manage Containers and Virtual Machines.

Linux* Troubleshooting

- Make sure you have an active internet connection during the full installation. If you lose Internet connectivity at any time, the installation might fail.
- Make sure you are using a fresh OS installation. Earlier software, especially Docker* and Docker Compose*, can cause issues.
- In proxy environment, if single user proxy is set (in the `.bashrc` file, for example) then some of the component installation may fail or installation hangs. So make sure you have set the proxy in `/etc/` environment.
- For PRC users: downloading and installation of CSP and K3s components may fail. Only the components below are supported in PRC:
 - Intel® Distribution of OpenVINO™ toolkit in a Container
 - Intel® Distribution of OpenVINO™ toolkit Runtime
 - Deep Learning (DL) Streamer Pipelines
 - Docker* Community Edition (CE)

If you're unable to resolve your issues, go to the [Support Forum](#).

Linux* Tutorials

Follow the tutorials in this section to learn how to use and configure Edge Insights for Vision for different use cases.

With step-by-step instructions covering real world usage scenarios, tutorials provide a learning path for developers to follow for mastering the usage of Edge Insights for Vision.

Get started in your learning journey with [Single and Multi-Object Detection with Hardware Acceleration on Linux*](#) and [Multi-Camera Detection of Social Distancing on Linux*](#).

Single and Multi-Object Detection with Hardware Acceleration on Linux*

This tutorial uses the sample application called "Object Detection YOLO* V3 Python* Demo." Object Detection YOLO V3 Python Demo uses OpenCV, a component of OpenVINO™, to display a frame with detections that are rendered as bounding boxes and labels, if provided. By default, this sample application displays latency and FPS.

Instructions are provided for three hardware configurations so you can choose the ones that fit your system's configuration:

- [Single and Multi-Object Detection on CPU on Linux*](#)
- [Single and Multi-Object Detection on GPU on Linux*](#)
- [Single and Multi-Object Detection on Intel® Vision Accelerator on Linux*](#)

Single and Multi-Object Detection on CPU on Linux*

This tutorial uses the sample application called "Object Detection YOLO* V3 Python* Demo." Object Detection YOLO V3 Python Demo uses OpenCV, a component of OpenVINO™, to display a frame with detections that are rendered as bounding boxes and labels, if provided. By default, this sample application displays latency and FPS.

Instructions in this tutorial are provided for a CPU.

For each configuration, the sample demonstrates two detection types:

- **Single detection** uses a basic data set to perform one-by-one person detection.
- **Multi-detection** uses an advanced data set to perform multi-object detection, such as a person and a car.

While running the sample applications, you will gain familiarity with the Intel® Distribution of OpenVINO™ toolkit.

Single and Multi-Object Detection with Hardware Acceleration on a CPU

Run these steps on the target system.

The screenshots below were taken on a system with the following configuration:

- Model name/Processor: 11th Gen Intel® Core™ i7-1185GRE @ 2.80 GHz
- Graphics: Intel® Iris® Xe Graphics (TGL GT2)
- Memory: 8 GB
- Disk space: 500 GB
- OS: Ubuntu* 20.04

Step 1: Initialize the Intel® Distribution of OpenVINO™ toolkit Environment

1. Open a terminal window.
2. Go to the sample application directory in which the Object Detection YOLO V3 Python demo is located:

```
cd $HOME/Downloads/YOLOv3
```

3. Initialize the OpenVINO™ environment:

```
source /opt/intel/openvino_2022/setupvars.sh
```

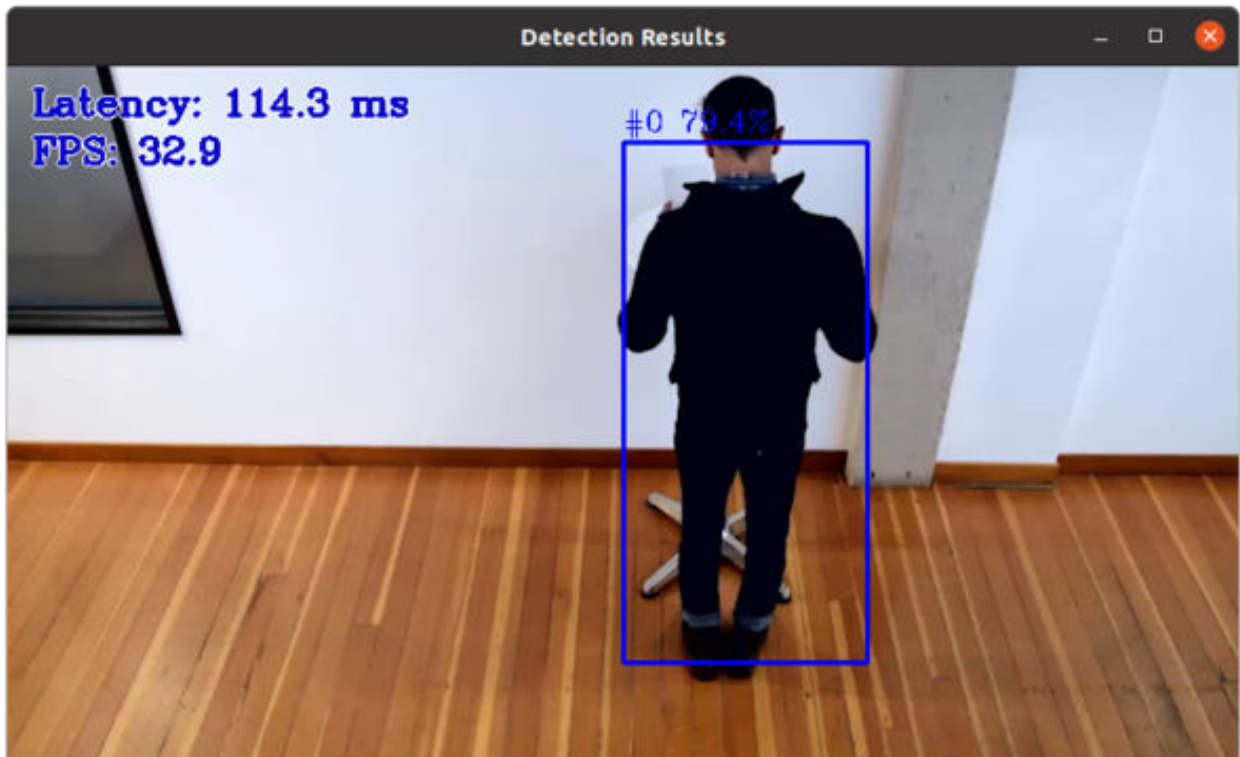
Leave the terminal window open for the next step.

Step 2: Run the Single Detection Application on the CPU

1. Run the Object Detection YOLO V3 Python Demo sample application:

```
python3 object_detection_demo.py -i $HOME/Downloads/YOLOv3/Sample_videos/one-by-one-person-detection.mp4 -m $HOME/Downloads/YOLOv3/tensorflow-yolo-v3/FP32/frozen_darknet_yolov3_model.xml -t 0.1 -at yolo
```

Success is indicated by an image that shows a single individual in a bounding box. At the left side of the image you see the latency. You might not clearly see some bounding boxes and detections if scene components are the same color as the bounding box or text.



2. Press **CTRL+C** on the terminal window to exit the demo.

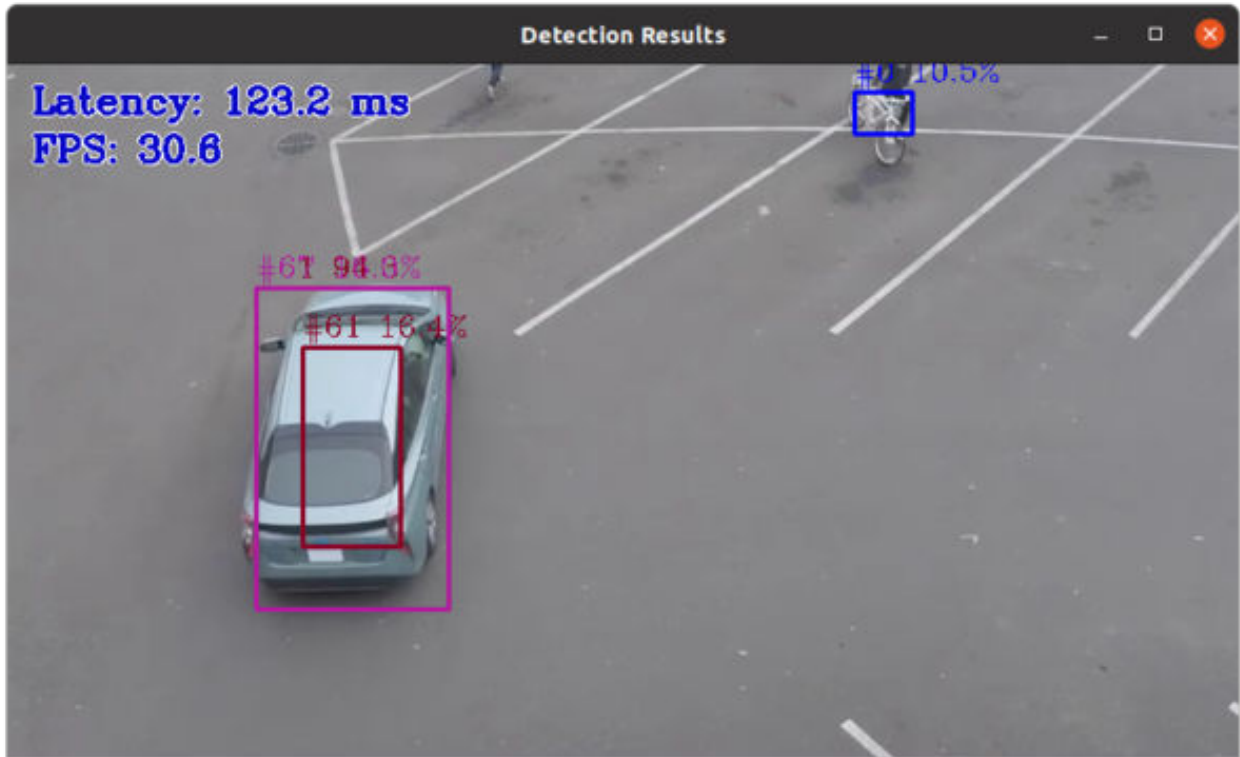
Leave the terminal window open for the next step.

Step 3: Run the Multi-Detection Application on the CPU

1. Run the Object Detection YOLO V3 Python Demo sample application:

```
python3 object_detection_demo.py -i $HOME/Downloads/YOLOv3/Sample_videos/person-bicycle-car-detection.mp4 -m $HOME/Downloads/YOLOv3/tensorflow-yolo-v3/FP32/frozen_darknet_yolov3_model.xml -t 0.1 -at yolo
```

Success is indicated by an image that shows one or more objects and/or people. At the left side of the image you see the latency. You might not clearly see some bounding boxes and detections if scene components are the same color as the bounding box or text.



2. Press **CTRL+C** on the terminal window to exit the demo.

If you want to run the sample application on a GPU or the Intel® Vision Accelerator, leave the terminal window open and begin with Step 2 of the GPU or Intel® Vision Accelerator instructions.

Summary and Next Steps

In this tutorial, you learned to run inference applications on different processing units using the sample application "Object Detection YOLO V3 Python Demo." In the process, you gained familiarity with the Intel® Distribution of OpenVINO™ toolkit, which was installed with the Edge Insights for Vision.

Go to [Intel® Edge Software Configurator Documentation](#) to learn how to use the tool to manage Edge Software Packages, create and manage Containers and Virtual Machines.

As a next step, see the [Single and Multi-Object Detection on GPU on Linux*](#) tutorial.

Single and Multi-Object Detection on GPU on Linux*

This tutorial uses the sample application called "Object Detection YOLO* V3 Python* Demo." Object Detection YOLO V3 Python Demo uses OpenCV, a component of OpenVINO™, to display a frame with detections that are rendered as bounding boxes and labels, if provided. By default, this sample application displays latency and FPS.

Instructions in this tutorial are provided for a GPU.

For each configuration, the sample demonstrates two detection types:

- **Single detection** uses a basic data set to perform one-by-one person detection.
- **Multi-detection** uses an advanced data set to perform multi-object detection, such as a person and a car.

While running the sample applications, you will gain familiarity with the Intel® Distribution of OpenVINO™ toolkit.

Single and Multi-Object Detection with Hardware Acceleration on a GPU

If you used the CPU instructions and left your terminal window open, skip ahead to **Step 2**.

Run these steps on the target system.

The screenshots below were taken on a system with the following configuration:

- Model name/Processor: 11th Gen Intel® Core™ i7-1185GRE @ 2.80 GHz
- Graphics: Intel® Iris® Xe Graphics (TGL GT2)
- Memory: 8 GB
- Disk space: 500 GB
- OS: Ubuntu* 20.04

Step 1: Initialize the Intel® Distribution of OpenVINO™ toolkit Environment

1. Open a terminal window.
2. Go to the sample application directory in which the Object Detection YOLO V3 Python demo is located:

```
cd $HOME/Downloads/YOLOv3
```

3. Initialize the OpenVINO™ environment:

```
source /opt/intel/openvino_2022/setupvars.sh
```

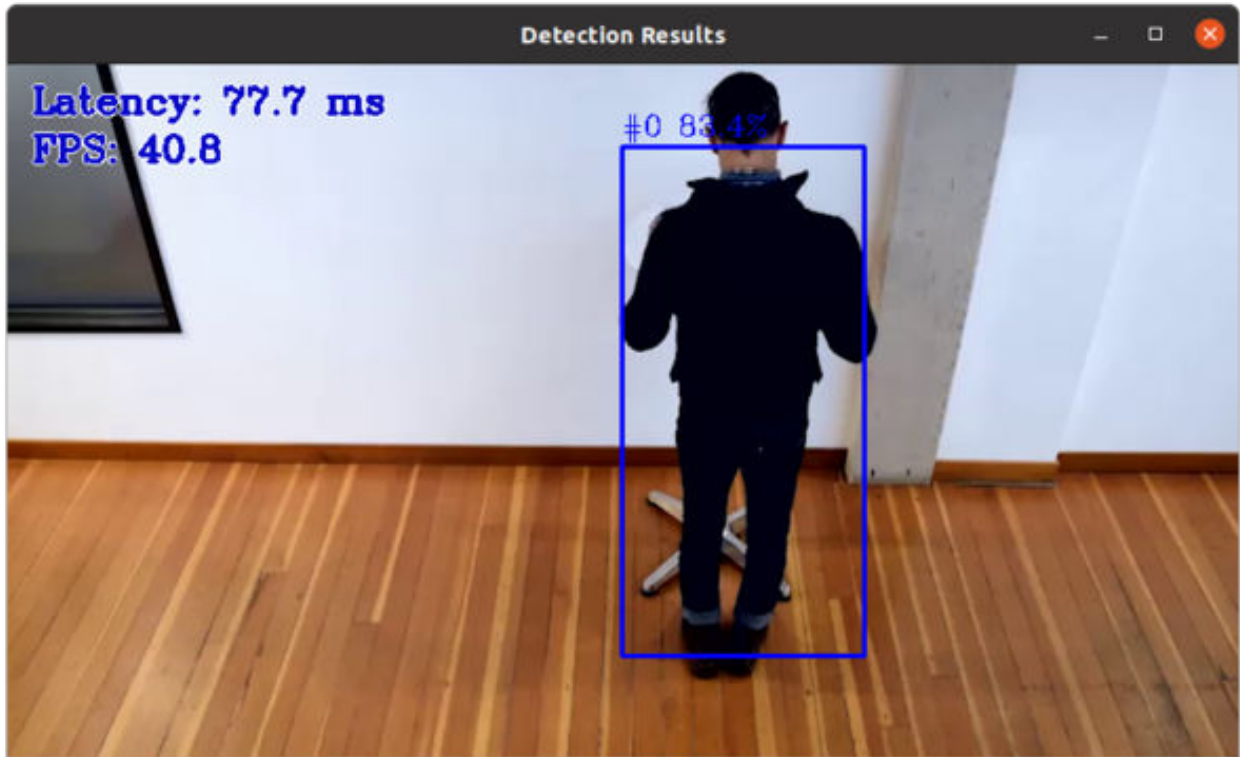
Leave the terminal window open for the next step.

Step 2: Run the Single Detection Application on the GPU

1. Run the Object Detection YOLO V3 Python Demo sample application:

```
python3 object_detection_demo.py -i $HOME/Downloads/YOLOv3/Sample_videos/one-by-one-person-detection.mp4 -m $HOME/Downloads/YOLOv3/tensorflow-yolo-v3/FP32/frozen_darknet_yolov3_model.xml -d GPU -t 0.1 -at yolo
```

Success is indicated by an image that shows a single individual in a bounding box. At the left side of the image you see the latency. You might not clearly see some bounding boxes and detections if scene components are the same color as the bounding box or text.



2. Press **CTRL+C** on the terminal window to exit the demo.

Leave the terminal window open for the next step.

Step 3: Run the Multi-Detection Application on the GPU

1. Run the Object Detection YOLO V3 Python Demo sample application:

```
python3 object_detection_demo.py -i $HOME/Downloads/YOLOv3/Sample_videos/person-bicycle-car-detection.mp4 -m $HOME/Downloads/YOLOv3/tensorflow-yolo-v3/FP32/frozen_darknet_yolov3_model.xml -d GPU -t 0.1 -at yolo
```

Success is indicated by an image that shows one or more objects and/or people. At the left side of the image you see the latency. You might not clearly see some bounding boxes and detections if scene components are the same color as the bounding box or text.



2. Press **CTRL+C** on the terminal window to exit the demo.

Summary and Next Steps

In this tutorial, you learned to run inference applications on different processing units using the sample application "Object Detection YOLO V3 Python Demo." In the process, you gained familiarity with the Intel® Distribution of OpenVINO™ toolkit, which was installed with the Edge Insights for Vision.

Go to [Intel® Edge Software Configurator Documentation](#) to learn how to use the tool to manage Edge Software Packages, create and manage Containers and Virtual Machines.

As a next step, see the [Single and Multi-Object Detection on Intel® Vision Accelerator on Linux*](#) tutorial.

Single and Multi-Object Detection on Intel® Vision Accelerator on Linux*

This tutorial uses the sample application called "Object Detection YOLO* V3 Python* Demo." Object Detection YOLO V3 Python Demo uses OpenCV, a component of OpenVINO™, to display a frame with detections that are rendered as bounding boxes and labels, if provided. By default, this sample application displays latency and FPS.

Instructions in this tutorial are provided for the Intel® Vision Accelerator.

NOTE

1. The Intel® Vision Accelerator is not supported with kernel version 5.17.15.
 2. Initialization of GPU and Intel® Vision Accelerator might take some time for the inference to start.
 3. You can ignore any warning messages that may be shown before starting inference.
-

For each configuration, the sample demonstrates two detection types:

- **Single detection** uses a basic data set to perform one-by-one person detection.

- **Multi-detection** uses an advanced data set to perform multi-object detection, such as a person and a car.

While running the sample applications, you will gain familiarity with the Intel® Distribution of OpenVINO™ toolkit.

Single and Multi-Object Detection with Hardware Acceleration on an Intel® Vision Accelerator

By running the application on the Intel® Vision Accelerator, you are offloading processing of inference to the Intel® Vision Accelerator and freeing up your CPU for other applications.

If you used the CPU instructions and left your terminal window open, skip ahead to **Step 2**.

Run these steps on the target system.

The screenshots below were taken on a system with the following configuration:

- Model name/Processor: 11th Gen Intel® Core™ i7-1185GRE @ 2.80 GHz
- Graphics: Intel® Iris® Xe Graphics (TGL GT2)
- Memory: 8 GB
- Disk space: 500 GB
- OS: Ubuntu* 20.04

Step 1: Initialize the Intel® Distribution of OpenVINO™ toolkit Environment

1. Open a terminal window.
2. Go to the sample application directory in which the Object Detection YOLO V3 Python demo is located:

```
cd $HOME/Downloads/YOLOv3
```

3. Initialize the OpenVINO™ environment:

```
source /opt/intel/openvino_2022/setupvars.sh
```

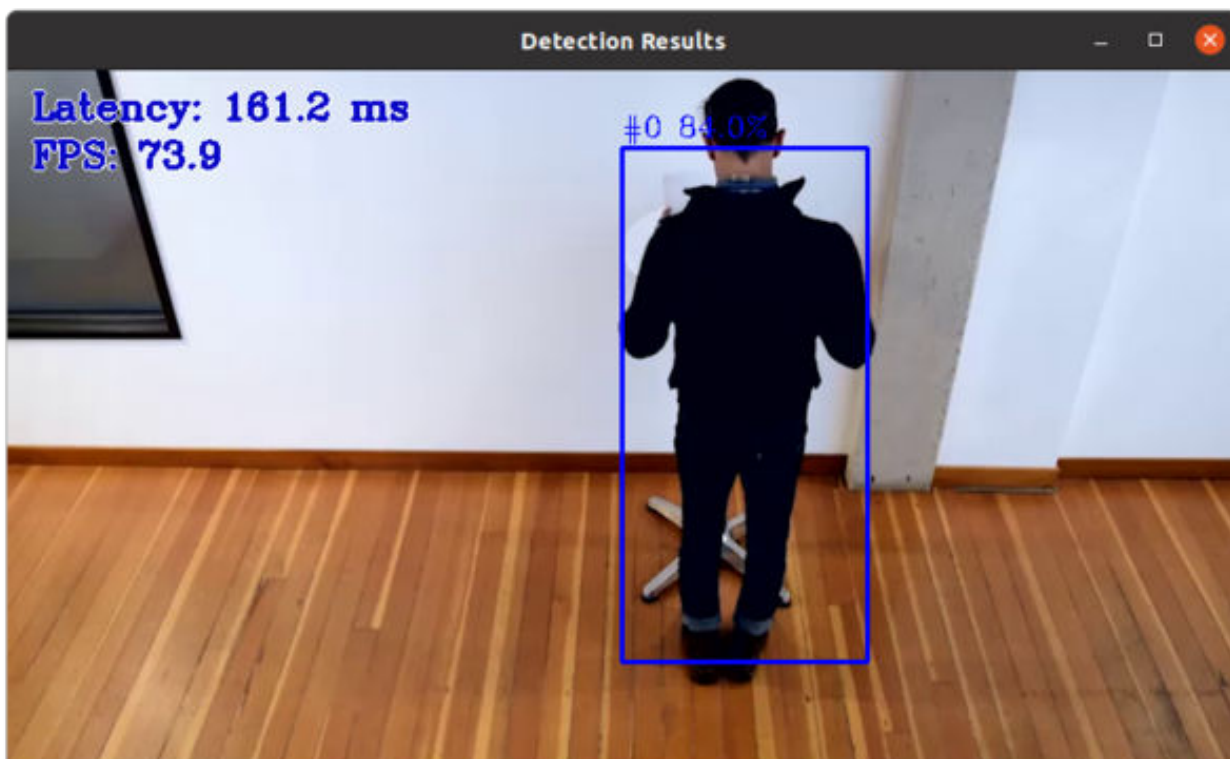
Leave the terminal window open for the next step.

Step 2: Run the Single Detection Application on an Intel® Vision Accelerator

1. Run the Object Detection YOLO V3 Python Demo sample application:

```
python3 object_detection_demo.py -i $HOME/Downloads/YOLOv3/Sample_videos/one-by-one-person-detection.mp4 -m $HOME/Downloads/YOLOv3/tensorflow-yolo-v3/FP32/frozen_darknet_yolov3_model.xml -d HDDL -t 0.1 -at yolo
```

Success is indicated by an image that shows a single individual in a bounding box. At the left side of the image you see the latency. You might not clearly see some bounding boxes and detections if scene components are the same color as the bounding box or text.



2. Press **CTRL+C** on the terminal window to exit the demo.

Leave the terminal window open for the next step.

Step 3: Run the Multi-Detection Application on an Intel® Vision Accelerator

1. Run the Object Detection YOLO V3 Python Demo sample application:

```
python3 object_detection_demo.py -i $HOME/Downloads/YOLOv3/Sample_videos/person-bicycle-car-  
detection.mp4 -m $HOME/Downloads/YOLOv3/tensorflow-yolo-v3/FP32/frozen_darknet_yolov3_model.xml -  
d HDDL -t 0.1 -at yolo
```

Success is indicated by an image that shows one or more objects and/or people. At the left side of the image you see the latency. You might not clearly see some bounding boxes and detections if scene components are the same color as the bounding box or text.



2. Press **CTRL+C** on the terminal window to exit the demo.

Summary and Next Steps

In this tutorial, you learned to run inference applications on different processing units using the sample application "Object Detection YOLO V3 Python Demo." In the process, you gained familiarity with the Intel® Distribution of OpenVINO™ toolkit, which was installed with the Edge Insights for Vision.

Go to [Intel® Edge Software Configurator Documentation](#) to learn how to use the tool to manage Edge Software Packages, create and manage Containers and Virtual Machines.

As a next step, see the [Multi-Camera Detection of Social Distancing on Linux*](#) tutorial.

Multi-Camera Detection of Social Distancing on Linux*

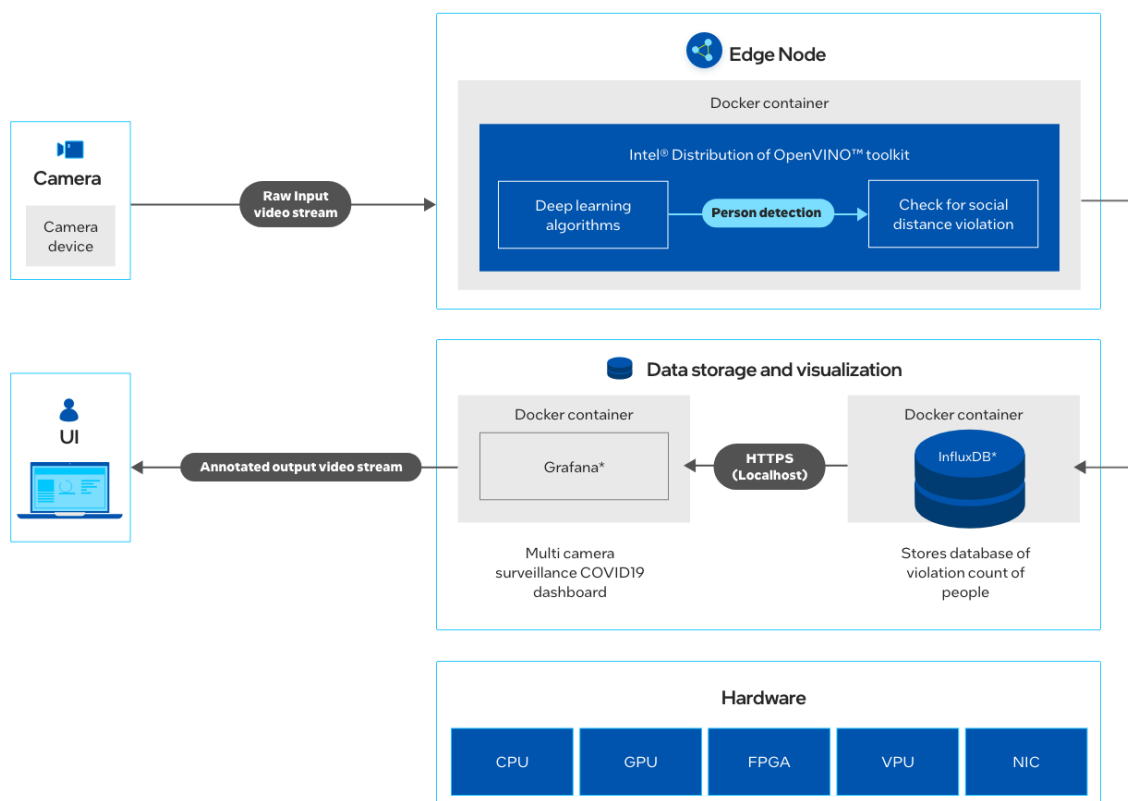
Overview

Social distancing is one of the most effective non-pharmaceutical ways to prevent the spread of disease. This tutorial gives a solution to prevent the spread of disease by using computer vision inference in the Intel® Distribution of OpenVINO™ toolkit to measure distance between people and store data to InfluxDB*. This data can be visualized on a Grafana* dashboard.

How It Works

A multi-camera solution demonstrates an end-to-end analytics pipeline to detect people and calculates social distance between people from multiple input feeds. Frames are transformed, scaled and normalized into BGR images which can be fed to the inference engine in the Intel® Distribution of OpenVINO™ toolkit. The steps below are performed for the inference.

- Apply Intel's person detection model, i.e., [person-detection-retail-0013](#) to detect people from all the video streams.
- Compute Euclidean distance between all the people from the above step.
- Based on above measurements, check whether any people are violating N pixels apart.
- Store total violations count of social distancing data in InfluxDB.
- Visualize the stored data of InfluxDB on Grafana dashboard.



Get Started

Step 1: Install

The Multi-Camera Detection of Social Distancing component will be installed with the [Edge Insights for Vision package](#) and will be available in the target system.

Go to the Multi-Camera Detection of Social Distancing component directory from the terminal by running the command:

```
cd $HOME/edge_insights_vision/Edge_Insights_for_Vision_<version>/
RI_MultiCamera_Social_Distancing/mcss-covid19/
```

Where <version> is the Edge Insights for Vision version selected while downloading.

Step 2: Download the Input Video

The application works better with input feed in which cameras are placed at eye level angle.

Please download [sample video](#) at **1280x720** resolution and place it in the `$HOME/edge_insights_vision/Edge_Insights_for_Vision_<version>/RI_MultiCamera_Social_Distancing/mcss-covid19/resources` directory.

Where <version> is the Edge Insights for Vision version selected while downloading.

(Data set subject to this [license](#). The terms and conditions of the dataset license apply. Intel® does not grant any rights to the data files.)

To use any other video, specify the path `INPUT1` in the `run.sh` file inside the application directory.

The application also supports multi-video as input. The appropriate code with comments is available in the `run.sh` file inside the application directory.

```
INPUT1="${PWD}/../resources/<name_of_video_file>.mp4"
MIN_SOCIAL_DIST1=<appropriate_minimum_social_distance_for_input1>
```

Where `<appropriate_minimum_social_distance_for_input1>` is measured in cm. 80 cm is recommended.

(Optional) Test with USB Camera

To test with a USB camera, specify the camera index in the `run.sh` file.

On Ubuntu, to list all available video devices, run the following command:

```
ls /dev/video*
```

For example, if the output of the command is `/dev/video0`, then make changes to the following variables such as `INPUT1` and `MIN_SOCIAL_DIST1` in the `run.sh` file inside the application folder.

```
INPUT1=/dev/video0
MIN_DIST1=<appropriate_minimum_social_distance_for_input1>
```

Step 3: Initialize Environment Variables

Run the following command to initialize OpenVINO™ environmental variables:

```
source /opt/intel/openvino_2022/setupvars.sh
```

NOTE

Set `no_proxy` in the terminal under proxy network if it's not set, using the command:

```
export no_proxy=localhost,127.0.0.1
```

Run the Application

Instructions in this tutorial are provided for three hardware configurations (CPU, GPU, and Intel® Vision Accelerator). Configure the application by modifying the `DEVICE1` parameter.

1. Change to the application directory:

```
cd application
```

2. Inside the `run.sh` file, change the following parameters (if required):

```
PERSON_DETECTOR="${PWD}/../intel/person-detection-retail-0013/FP16/person-detection-retail-0013.xml"
DEVICE1="<device>"
```

where `<device>` can be CPU, GPU, or HDDL (Intel® Vision Accelerator).

3. Change the permissions for the `run.sh` file and run the script:

```
chmod +x run.sh
./run.sh
```

NOTE

1. Application parameters can be changed as per the requirements in the `run.sh` file.
2. Initialization of GPU and Intel® Vision Accelerator might take some time for the inference to start.

**Data Visualization on Grafana**

NOTE The application must be running in parallel to view the results in Grafana.

1. Navigate to **localhost:3000** on your browser.

NOTE If browser shows Unable to connect, then make sure Grafana service status is active using the command `sudo service grafana-server status`. If service is not active, then start the service by running the command `sudo service grafana-server start` in the terminal.

2. Login with user as **admin** and password as **admin**.
3. Go to **Configuration (Settings icon)** and select **Data Sources**.
4. Select **+ Add data source**, select **InfluxDB**, and provide the following details:

```
Name: Mcss Covid
URL: http://localhost:8086
Auth: Enable Skip TLS Verify
InfluxDB details:
  Database: McssCovid
  HTTPMethod: GET
```

5. Click **Save and Test**.

The screenshot shows the 'Data Sources / InfluxDB' configuration page in a dark-themed interface. The page is titled 'M Settings' and shows a configuration for a data source named 'Mcss Covid'. The 'Query Language' is set to 'InfluxQL'. Under the 'HTTP' section, the 'URL' is 'http://localhost:8086', 'Access' is 'Server (default)', 'Allowed cookies' is 'New tag (enter key to add)', and 'Timeout' is 'Timeout in seconds'. The 'Auth' section has 'Basic auth' and 'TLS Client Auth' both disabled, 'Skip TLS Verify' enabled, and 'Forward OAuth Identity' disabled. There is a 'Custom HTTP Headers' section with an 'Add header' button. The 'InfluxDB Details' section contains a 'Database Access' warning and a 'Database' field set to 'McssCovid'. At the bottom, there are fields for 'User', 'Password', 'HTTP Method' (set to 'GET'), 'Min time interval' (set to '10s'), and 'Max series' (set to '1000').

Data Sources / InfluxDB
Type: influxDB

M Settings

Name: Mcss Covid Default ☒

Query Language
InfluxQL

HTTP

URL: http://localhost:8086

Access: Server (default) Help

Allowed cookies: New tag (enter key to add)

Timeout: Timeout in seconds

Auth

Basic auth: ☐ With Credentials: ☐

TLS Client Auth: ☐ With CA Cert: ☐

Skip TLS Verify: ☒

Forward OAuth Identity: ☐

Custom HTTP Headers

+ Add header

InfluxDB Details

Database Access

Setting the database for this datasource does not deny access to other databases. The InfluxDB query syntax allows switching the database in the query. For example: `SHOW MEASUREMENTS ON _internal` or `SELECT * FROM "_internal"."database" LIMIT 10`

To support data isolation and security, make sure appropriate permissions are configured in InfluxDB.

Database: McssCovid

User:

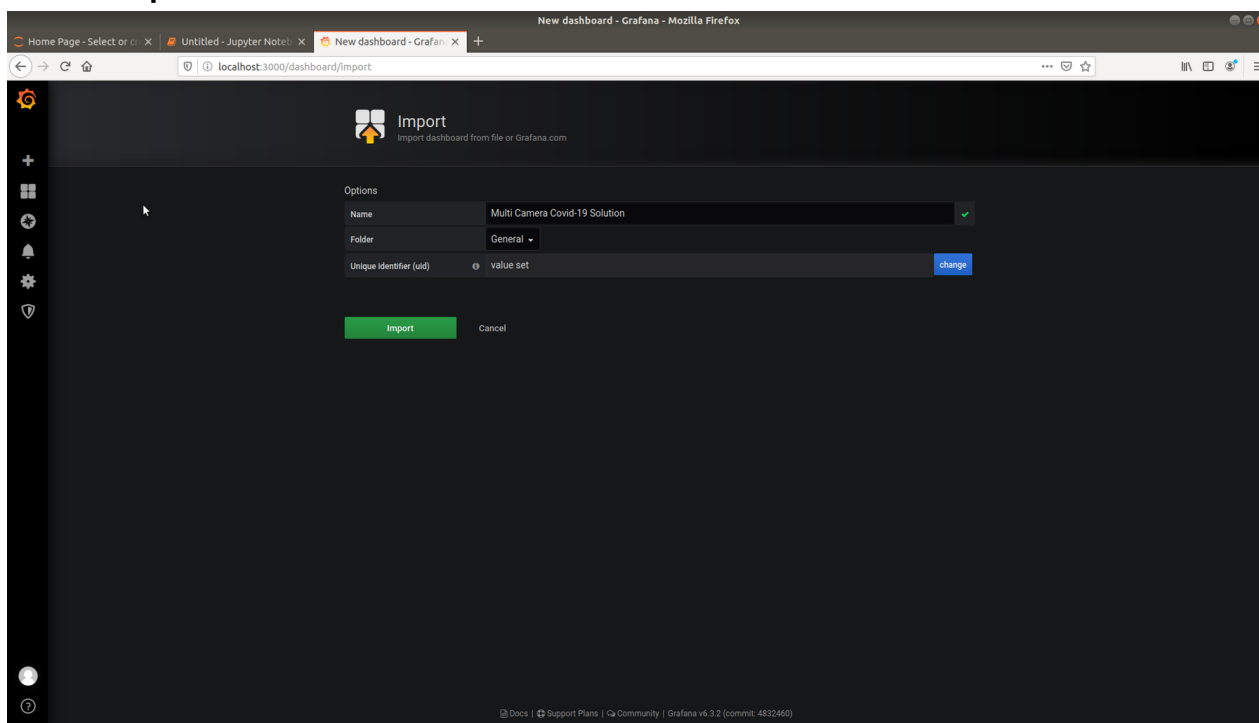
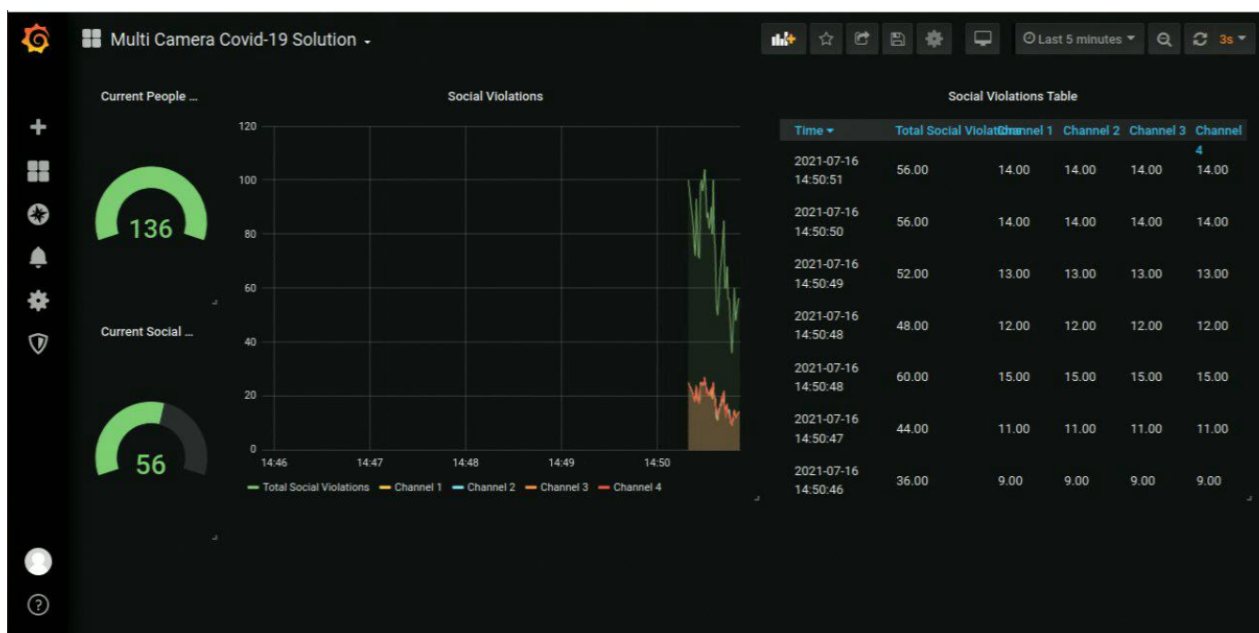
Password: Password

HTTP Method: GET

Min time interval: 10s

Max series: 1000

6. Go to Dashboard (icon on the left side of the window) and select **+ Imports**.
7. Choose **Upload.json File** and import the `mcss-covid19/resources/multi_cam.json` file.

8. Click on **Import**.9. Click on **Multi Camera Covid-19 Solution** dashboard to view real time violation data.

Summary and Next Steps

This application successfully leverages Intel® Distribution of OpenVINO™ toolkit plugins for detecting and measuring distance between the people and storing data to InfluxDB. It can be extended further to provide support for feed from network stream (RTSP camera) and the algorithm can be optimized for better performance.

As a next step, you can explore other [use cases and reference implementations](#) on the Edge Insights for Vision page.

Release Notes

New in this Release

- Added support for Intel® Distribution of OpenVINO™ toolkit version 2022.1.0 for the following components for Ubuntu* 20:
 - Intel® Distribution of OpenVINO™ toolkit 2022.1.0 Runtime
 - Intel® Distribution of OpenVINO™ toolkit 2022.1.0 in a Container
 - Intel® Deep Learning Streamer (Intel® DL Streamer) Pipelines
 - Reference Implementations - Single and Multi-Object Detection with Hardware Acceleration and Multi-Camera Detection of Social Distancing

Known Issues

- The installation is halted due to dpkg locks / error 1.
Workaround: Reboot the target machine or force stop the process holding the dpkg lock.
- Installation of OpenVINO™ runtime component might fail in VM environment.
(See: <https://github.com/openvinotoolkit/openvino/issues/7555>)

Documentation Archive

View earlier versions of Edge Insights for Vision documentation:

- [2021.4.2](#)

NOTE Only the most current version of the documentation is maintained and updated.
