

Edge Insights for Fleet

Contents

Chapter 1: Edge Insights for Fleet

How it Works	3
Get Started Guide	6
Requirements	6
Install Edge Insights for Fleet	7
Introduction to the Edge Software Command Line Interface (CLI)	14
Set Up Windows* Subsystem for Linux* (Optional)	29
Reference Implementations	31
Set Up ThingsBoard* Cloud Data	31
Set Up Amazon Web Services* Cloud Storage	53
Update an Application-Over-The-Air	57
Set Up KnowGo* Simulator and Vehicle Simulator Data	89
Use Case Manager.....	101
Release Notes	106

1

Edge Insights for Fleet

Edge Insights for Fleet from Intel is a set of pre-validated ingredients for integrating video analytics on edge compute nodes.

Edge Insights for Fleet helps to address various commercial vehicle fleet usages, which include data collection, storage, and analytics using an in-vehicle computer edge node on the information about the driver, the vehicle, and the cargo load. See [How it Works](#).

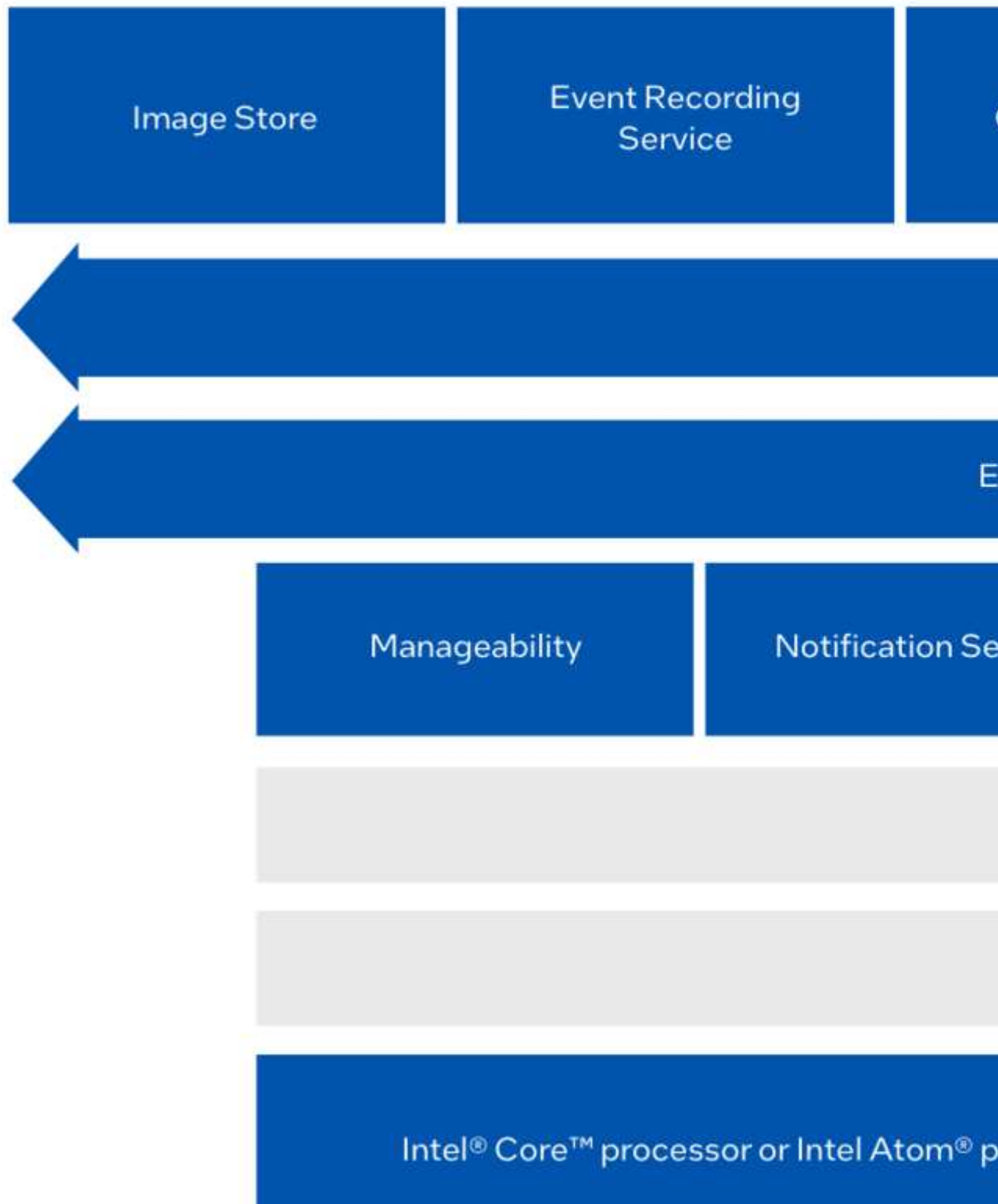
Use the [Get Started Guide](#) for installation instructions and an introduction to the Edge Software command line interface to learn how to manage Intel® Developer Catalog packages.

When set up is complete, choose the [Reference Implementations](#) section for step-by-step, hands-on walkthroughs of how to use and configure modules in Edge Insights for Fleet.

How it Works

Edge Insights for Fleet is a set of pre-integrated ingredients designed to accelerate the development and deployment of solutions for the industrial sector. The package is meant to be deployed on the device closest to the data generation, such as the tool or machine assembling a product. This enables the ingestion of video and time-series data, storage of data, performing analytics, closing the loop by transmitting a control message, and publishing the results.

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



Modules and Services

Video Ingestion and Analytics

Video Analytics

Classifier algorithms are executed in the Video Analytics container. The container uses the Edge Insights Software Data Bus to subscribe to the input stream published by the Video Ingestion container. The container also uses MQTT through Kuiper* service to send data to the Alert Notification service.

The algorithms included in the container are optimized for Intel hardware using the Intel® Distribution of OpenVINO™ toolkit (OpenVINO™ and the Intel® Math Kernel Library. You can create multiple instances of the Video Analytics container to classify multiple video input sources.

- The [Intel® Distribution of OpenVINO™ toolkit](#) is a software development kit (SDK) for deploying deep learning computer vision inference applications. It optimizes inferencing on your edge IoT device for Intel® architecture.
- The [Intel® Math Kernel Library](#) is a library of highly-optimized, threaded, and vectorized math routines designed to maximize performance on Intel® processors.

Video Ingestion

Edge Insights for Fleet supports ingestion from:

- Video files.
- GigE cameras.
- USB cameras.
- RTSP cameras.

Edge Insights for Fleet uses [GStreamer*](#) and OpenCV to provide a preconfigured ingestion pipeline that you can modify for different camera types and preprocessing algorithms. The Video Ingestion container publishes video data, consisting of metadata and frames, to the Edge Insights Software Data Bus. You can create multiple instances of the Video Ingestion container to ingest video from multiple input sources. In addition, the Video Ingestion container itself can run UDFs performing analytics. This is recommended for faster processing of frames to avoid transmission delays.

Training and Learning Suite (TLS)

The Training and Learning Suite has a web-based user interface for training deep learning models. The key TLS features are:

- Remote deployment of a TLS-trained model into an Edge Insights for Fleet system.
- Visualization of Edge Insights for Fleet video data classification results.

Image Store

The image store provides storage and retrieval of images as binary blobs. You can store the images persistently by using [SQLite*](#), an open source, SQL database engine.

Security and Configuration

The Data Security Framework enables security through a two-stage process involving provisioning and execution on an Edge Compute Node. Its primary objective is to prevent unauthorized access of generated data from within the system or via external network interfaces.

Edge Insights for Fleet uses [etcd*](#) for configuration management. etcd* is a consistent, distributed key-value store that provides a reliable way to store data that needs to be accessed by a distributed system or cluster of machines.

Docker

[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.

Edge Insights Software Data Bus

The Edge Insights Software Data Bus is an abstraction over ZeroMQ*, which is used for all inter-container communication.

ZeroMQ* is a brokerless message bus that transfers data from the source directly to the destination. ZeroMQ* is used in TCP and IPC mode with pub-sub and request-response patterns.

Kuiper* MQTT Communication

Kuiper* is an edge lightweight IoT data analytics/streaming software implemented by Golang*. It can be run on many types of resource-constrained edge devices.

Get Started Guide

This step-by-step guide takes you through installing the Edge Insights for Fleet 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 [Requirements](#) section before you get started with installation.

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

NOTE 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](#).

Edge Insights for Fleet 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.

See [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® Developer Catalog packages.

Requirements

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

Target System

- One of the following processors:
 - 6th, 7th, or 8th generation Intel® Core™ processor.
 - 6th, 7th, or 8th generation Intel® Xeon® processor.
 - Intel® Pentium® processor N4200/5, N3350/5, N3450/5 with Intel® HD Graphics.
- At least 16 GB RAM.
- At least 64 GB hard drive.
- An Internet connection.
- Ubuntu* 20.04 LTS.

Knowledge/Experience

You are familiar with executing Linux* commands.

Install Edge Insights for Fleet

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:

- Prepare your target system.
- Copy the package.
- Complete the installation steps.

NOTE Be aware that screenshots may show a package version number that is different from the current release. See the [Release Notes](#) for information on the current release.

Step 1: Prepare the Target System

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

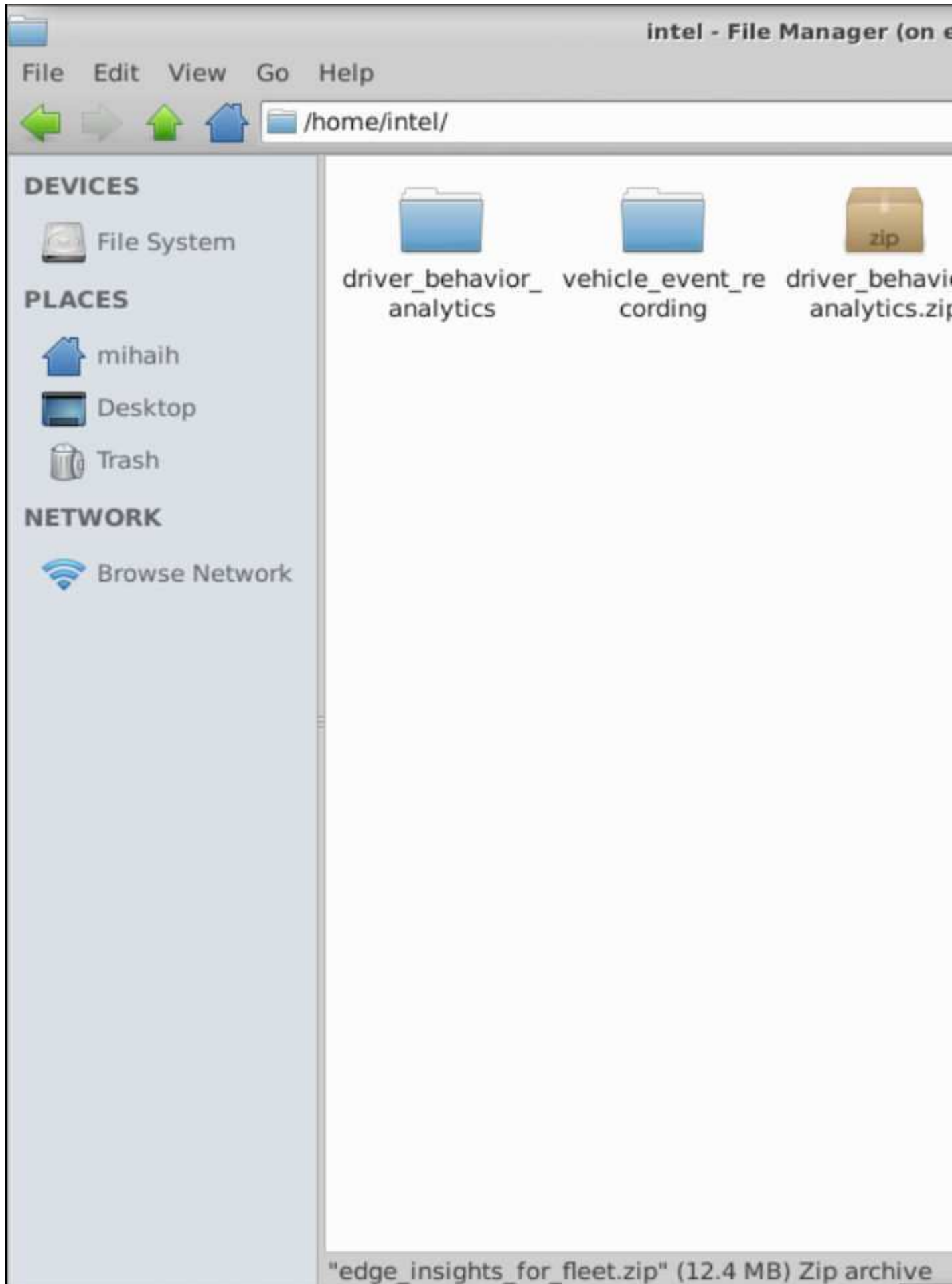
1. Download [Ubuntu package](#) to your developer workstation.
2. Create a bootable flash drive using an imaging application, such as Startup Disk Creator, available on Ubuntu*.
3. After flashing the USB drive, power off your target system, insert the USB drive, and power on the target system.

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 Ubuntu* 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* Desktop is successfully installed.
7. You must have `sudo` access for the target system.
8. You must have GUI access for the target system.

Step 2: Copy the Edge Insights for Fleet .zip File to the Target System

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

1. Copy `edge_insights_for_fleet.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:



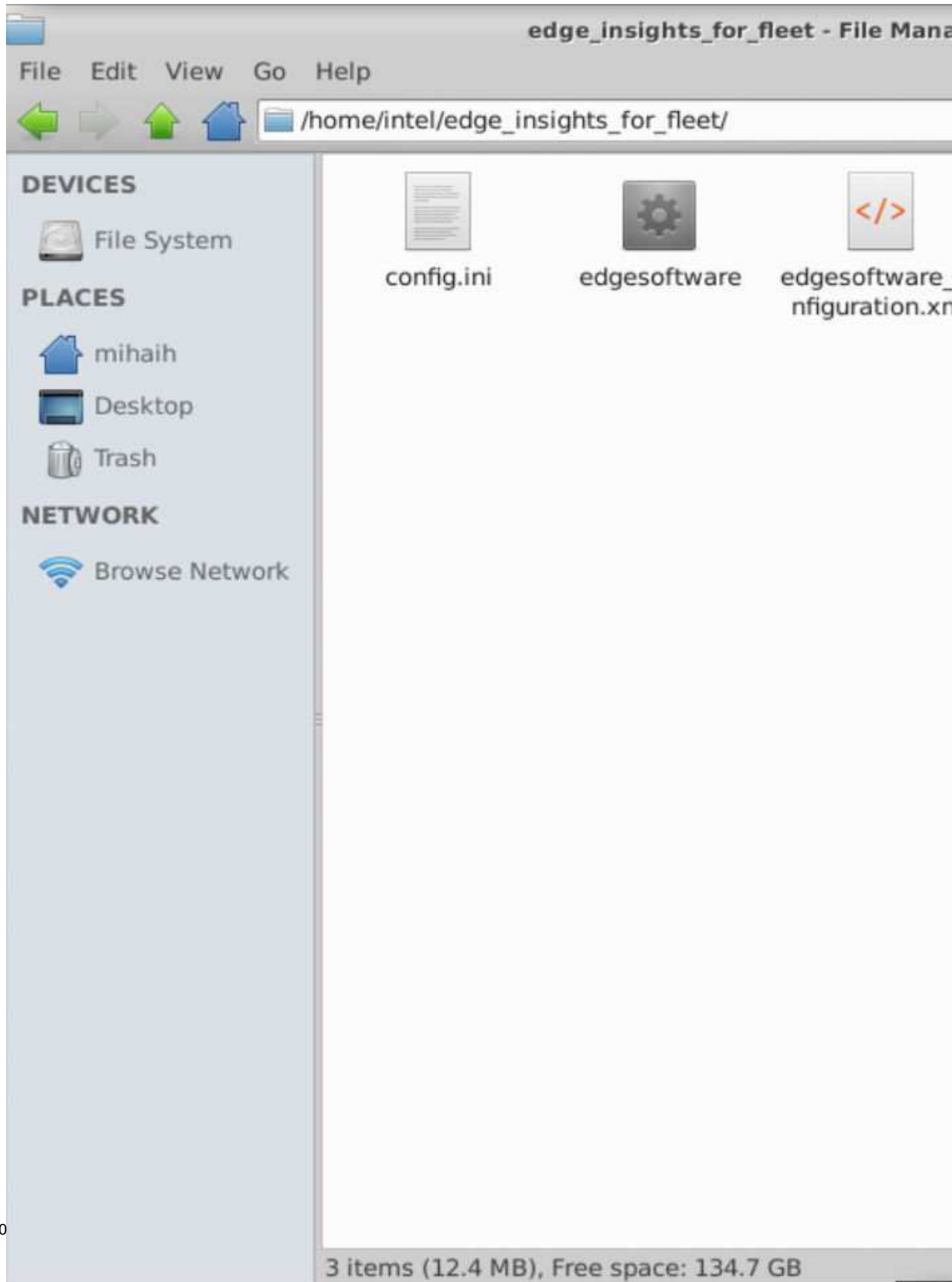
Step 3: Extract the Edge Insights for Fleet Software

In this step you extract `edge_insights_for_fleet.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_for_fleet.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 4: Install the Edge Insights for Fleet Software

NOTE If you are running behind a proxy server, please be sure that the proxy settings are configured correctly. The `edgesoftware` tool uses these proxy settings to download and install the modules.

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

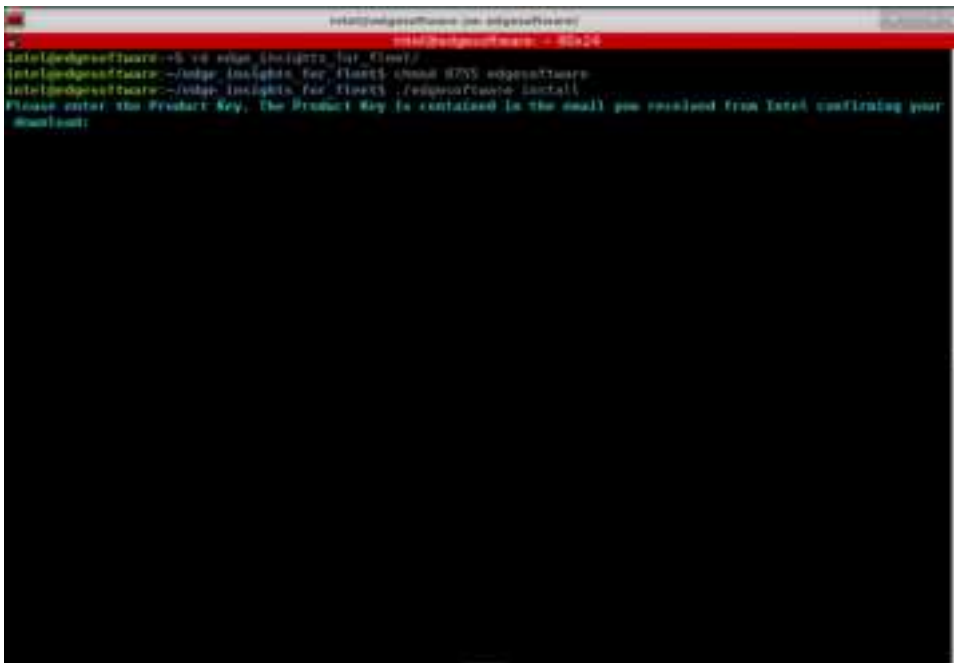
The software installation will take 1 to 2 hours. The completion time depends on your target system and Internet connection.

1. Run these commands:

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

NOTE If you encounter any Docker* pull-related issues during the installation process, refer to the [Troubleshooting](#) section.

2. Type the product key at the prompt:



3. 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.

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

```

Activities  X-terminal-emulator  ▾
int
int
Step 21/21 : ENTRYPOINT ["python3.6", "web_visualizer.py"]
---> Running in d2af17f13b66
Removing intermediate container d2af17f13b66
---> 011d4c6c2b31

[Warning] One or more build-args [EIS_UID] were not consumed
Successfully built 011d4c6c2b31
Creating edgeinsightssoftware_ia_openvino_base_1          ... done
Creating ia_video_ingestion                               ... done
Creating ia_video_analytics                               ... done
Verifying Docker images
[sudo] password for hutatum:
SUCCESS : Verified ia_eisbase
SUCCESS : Verified ia_common
SUCCESS : Verified ia_video_common
SUCCESS : Verified ia_openvino_base
SUCCESS : Verified ia_eis_azure_bridge
SUCCESS : Verified ia_azure_simple_subscriber
SUCCESS : Verified ia_eis_zmq_broker
SUCCESS : Verified ia_etcd_ui
SUCCESS : Verified ia_factoryctrl_app
SUCCESS : Verified ia_imagestore
SUCCESS : Verified ia_influxdbconnector
SUCCESS : Verified ia_opcua_export
SUCCESS : Verified ia_rest_export
SUCCESS : Verified ia_tls_remoteagent
SUCCESS : Verified ia_video_analytics
SUCCESS : Verified ia_video_ingestion
SUCCESS : Verified ia_visualizer
SUCCESS : Verified ia_web_visualizer
Building Insights Images [.....]
Edge Insights Visualizer [.....]
Successfully installed eii_installer took 30 minutes 32.84 seconds
Installation of package complete
***Recommended to reboot system after installation***

+-----+-----+-----+
| Id | Module | Status |
+-----+-----+-----+
| 5f21392e9e63c9002a6fd88d | Docker Community Edition CE | SUCCESS |
| 5f213aae9e63c9002a6fd88e | Docker Compose | SUCCESS |
| 5face41098a1ef002af292fd | eii_installer | SUCCESS |

```

Activities X-terminal-emulator ▾

```
intel@edgesoftware:~/edge_insights_for_fleet$ ./edgesoftware list
```

ID	Module	Status
5f21392e9e63c9002a6fd88d	Docker Community Edition CE	SUCCESS
5f213aae9e63c9002a6fd88e	Docker Compose	SUCCESS
5face41098a1ef002af292fd	eii installer	SUCCESS

```
intel@edgesoftware:~/edge_insights_for_fleet$
```

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 \(CLI\)](#).

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 Ubuntu* installation. Earlier software, especially Docker* and Docker Compose*, can cause issues.
- You may experience an installation timeout issue when using the People's Republic of China (PRC) internet network. Make sure that you have a stable internet connection while installing the packages. If you experience timeouts due to Linux* apt or Python* pip installation, try to reinstall the package.

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

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_for_fleet/` 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 installed as a part 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.
```

Download Package Modules

- Command:

```
./edgesoftware download
```

- Response: Downloads and unzips the modules of the package.

```
Activities X-terminal-emulator v
intel@edgesoftware:~/edge_insights_for_fleet$ ./edgesoftware download
Please enter the Product Key. The Product Key is contained in the email
Starting the setup...
ESB CLI version: 2021.3
Target OS: Ubuntu 18.04
Python version: 3.6.9
Checking Internet connection
Connected to the Internet
Validating product key
Successfully validated Product Key
Checking for prerequisites
All dependencies met
-----SYSTEM INFO-----
Package Name: Edge Insights for Fleet 2021.2
Product Name: Intel(R) Client Systems NUC7i7BNH
CPU SKU: Intel(R) Core(TM) i7-7567U CPU @ 3.50GHz
Memory Size: 32 GB
Operating System: Ubuntu 18.04.5 LTS
Kernel Version: 4.15.0-154-generic
Accelerator: None
CPU Utilization: 20.9%
Available Disk Space: 30 GB
Downloading modules...
Downloading component esb_common
ZIP file for module 5e8c4742e02f17002a2a6976 already exists. Validating
Module validation passed for 5e8c4742e02f17002a2a6976
Skipping download...
Downloading component Docker_Community_Edition_CE
ZIP file for module 5f21392e9e63c9002a6fd88d already exists. Validating
Module validation passed for 5f21392e9e63c9002a6fd88d
Skipping download...
Downloading component Docker_Compose
ZIP file for module 5f213aae9e63c9002a6fd88e already exists. Validating
Module validation passed for 5f213aae9e63c9002a6fd88e
Skipping download...
Downloading component IEdgeInsights
ZIP file for module 60891e1187b610002a2c3494 already exists. Validating
Module validation passed for 60891e1187b610002a2c3494
Skipping download...
Downloading component IEdgeInsights/Samples
```

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.

Activities X-terminal-emulator ▾

```
intel@edgesoftware:~/edge_insights_for_fleet$ ./edgesoftware list
```

ID	Module	Status
5f21392e9e63c9002a6fd88d	Docker Community Edition CE	SUCCESS
5f213aae9e63c9002a6fd88e	Docker Compose	SUCCESS
5face41098a1ef002af292fd	eii installer	SUCCESS

```
intel@edgesoftware:~/edge_insights_for_fleet$
```

List Modules Available for Download

- Command:

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

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

```

Activities X-terminal-emulator ▾

intel@edgesoftware:~/edge_insights_for_fleet$ ./edgesoftware list --d
Modules in the recommended configuration for 616edaa8de4d14002158ff66
+-----+-----+-----+
| ID | Module | Version |
+-----+-----+-----+
| 615415e40513ff0020ee01d4 | EII Core | 2.6.1 |
| 615415950513ff0020ee01b3 | EII-MessageBus | 2.6.1 |
| 6154159d0513ff0020ee01c0 | EII-C-Utills | 2.6.1 |
| 6154158a0513ff0020ee01a6 | EII Samples | 2.6.1 |
| 6154155b0513ff0020ee018d | Video Analytics | 2.6.1 |
| 615415530513ff0020ee017f | EII Installer | 2.6.1 |
| 615414390513ff0020ee00c0 | Video-Ingestion | 2.6.1 |
| 615414320513ff0020ee00b6 | Image Store | 2.6.1 |
| 615413a90513ff0020ee0097 | Video-Native-Visualizer | 2.6.1 |
| 615413990513ff0020ee0079 | Video Custom UDFs | 2.6.1 |
| 615413a00513ff0020ee0086 | Video-Web-Visualizer | 2.6.1 |
| 615413740513ff0020ee005e | Video Common | 2.6.1 |
| 6154136c0513ff0020ee004f | Video-Analytics | 2.6.1 |
| 615413640513ff0020ee0042 | EII-Tools | 2.6.1 |
| 615412d60513ff0020ee0024 | EII ETCD UI | 2.6.1 |
| 60e327614c1e9d002a6d6a7a | Docker Compose* | 1.29.0 |
| 5f21392e9e63c9002a6fd88d | Docker Community Edition (CE)* | 20.10.5 |
| 616ed88e1ed17e0021bb4ff6 | Edge Insights for Fleet | 3.0.0 |
| 615415ee0513ff0020ee01e0 | Device Manageability | 2.6.1 |
+-----+-----+-----+

intel@edgesoftware:~/edge_insights_for_fleet$

```

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

```
Activities X-terminal-emulator ▾

intel@edgesoftware:~/edge_insights_for_fleet$ ./edgesoftware log
=====Start of installer log=====
Tue Nov 23 12:06:47 IST 2021 - INFO - Artifacts file config_install.yml
Tue Nov 23 12:06:48 IST 2021 - INFO - Product key requirement status i
Tue Nov 23 12:08:23 IST 2021 - INFO - ESB CLI version: 2021.4
Target OS: Ubuntu 18.04
Tue Nov 23 12:08:23 IST 2021 - INFO - Python version: 3.6.9
Tue Nov 23 12:08:23 IST 2021 - INFO - Checking Internet connection
Tue Nov 23 12:08:24 IST 2021 - INFO - Connected to the Internet
Tue Nov 23 12:08:24 IST 2021 - INFO - Validating product key
Tue Nov 23 12:08:24 IST 2021 - INFO - Successfully validated Product K
Tue Nov 23 12:08:24 IST 2021 - INFO - Connected to a network in United
Tue Nov 23 12:08:24 IST 2021 - INFO - Checking for prerequisites
Tue Nov 23 12:08:38 IST 2021 - INFO - All dependencies met
Tue Nov 23 12:08:42 IST 2021 - INFO - -----SYSTEM INFO-----
Tue Nov 23 12:08:42 IST 2021 - INFO - Package Name: Edge Insights for
Tue Nov 23 12:08:42 IST 2021 - INFO - Product Name: Intel(R) Client Sy
Tue Nov 23 12:08:42 IST 2021 - INFO - CPU SKU: Intel(R) Core(TM) i7-75
Tue Nov 23 12:08:42 IST 2021 - INFO - Memory Size: 32 GB
Tue Nov 23 12:08:42 IST 2021 - INFO - Operating System: Ubuntu 18.04.6
Tue Nov 23 12:08:42 IST 2021 - INFO - Kernel Version: 4.15.0-162-gener
Tue Nov 23 12:08:42 IST 2021 - INFO - Accelerator: None
Tue Nov 23 12:08:42 IST 2021 - INFO - CPU Utilization: 4.9%
Tue Nov 23 12:08:42 IST 2021 - INFO - Available Disk Space: 27 GB
Tue Nov 23 12:08:44 IST 2021 - INFO - Starting installation
Tue Nov 23 12:08:44 IST 2021 - INFO - Modules to be downloaded by pack
/EtcdUI', 'IEdgeInsights/common/util/c', 'IEdgeInsights/common/libs/EI
EdgeInsights/VideoIngestion', 'IEdgeInsights/Visualizer', 'IEdgeInsigh
s', 'IEdgeInsights/OpcuaExport', 'IEdgeInsights/RestDataExport', 'IEdg
_detection', 'textile_defect_classifier', 'weld_porosity_detection', '
Tue Nov 23 12:08:44 IST 2021 - INFO - Downloading modules...
Tue Nov 23 12:08:44 IST 2021 - INFO - Downloading component esb_common
Tue Nov 23 12:08:44 IST 2021 - INFO - Sending request to download modu
Tue Nov 23 12:08:49 IST 2021 - INFO - Sending request to validate modu
Tue Nov 23 12:08:49 IST 2021 - INFO - Module validation passed for 5e8
Tue Nov 23 12:08:49 IST 2021 - INFO - Successfully downloaded module e
Tue Nov 23 12:08:49 IST 2021 - INFO - Downloading component Docker_Com
Tue Nov 23 12:08:49 IST 2021 - INFO - Sending request to download modu
Tue Nov 23 12:08:52 IST 2021 - INFO - Sending request to validate modu
Tue Nov 23 12:08:52 IST 2021 - INFO - Module validation passed for 5f2
```

See the Installation Event Log for a Module

- Command:

```
./edgesoftware log <MODULE_ID>
```

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

NOTE To find the module ID, use:

```
./edgesoftware list
```

- Response: The installation log for the module.

Activities X-terminal-emulator ▾

```
intel@edgesoftware:~/edge_insights_for_fleet$ ./edgesoftware log 5f213
=====Start of log for module Docker Compose=====
Mon Aug 30 11:26:01 IST 2021 - INFO - Starting installation docker-com
Mon Aug 30 11:26:01 IST 2021 - INFO - Removed image from cache memory
Mon Aug 30 11:26:06 IST 2021 - INFO - Docker Verified
Mon Aug 30 11:26:06 IST 2021 - INFO - Removed image from cache memory
Mon Aug 30 11:26:08 IST 2021 - INFO - Completed installation of docker
Mon Aug 30 11:26:08 IST 2021 - INFO - Verified docker-compose
Mon Aug 30 11:26:08 IST 2021 - INFO - Successfully installed docker-co
Mon Aug 30 11:26:08 IST 2021 - INFO - Completed cleanup of temporary f
=====End of log for module Docker Compose=====
intel@edgesoftware:~/edge_insights_for_fleet$
```

Install Package Modules

This `edgesoftware` command installs package modules on the target system. To do so, the command looks at `edgesoftware_configuration.xml` that was downloaded from the Intel® Developer Catalog when you installed the Edge Insights for Fleet software. This file contains information about the modules to install.

During the installation, you will be prompted to enter your product key. The product key is in the email message you received from Intel confirming your Edge Insights for Fleet download.

NOTE Do not manually edit the `edgesoftware_configuration.xml` file.

1. Open a terminal window.
2. Go to the `edge_insights_for_fleet/` directory.
3. Run the install command:

```
./edgesoftware install
```

Update the Package Modules

NOTE On a fresh Linux installation, you might need to use the `install` command at least once before performing an update. `install` makes sure all dependencies and packages are installed on the target system.

```
./edgesoftware install
```

When you are ready to perform the update, use:

```
./edgesoftware update <MODULE_ID>
```

During the installation, you will be prompted to enter your product key. The product key is in the email message you received from Intel confirming your Edge Insights for Fleet download.

NOTE To find the module ID, use:

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

```
Activities X-terminal-emulator v
intel@edgesoftware:~/edge_insights_for_fleet$ ./edgesoftware update 5f21
Please enter the Product Key. The Product Key is contained in the email
Updating ['5f213aae9e63c9002a6fd88e'] modules of package 60ed9a6ef177fe0
Starting the setup...
ESB CLI version: 2021.3
Target OS: Ubuntu 18.04
Python version: 3.6.9
Checking Internet connection
Connected to the Internet
Validating product key
Successfully validated Product Key
Checking for prerequisites
All dependencies met
-----SYSTEM INFO-----
Package Name: Edge Insights for Fleet 2021.2
Product Name: Intel(R) Client Systems NUC7i7BNH
CPU SKU: Intel(R) Core(TM) i7-7567U CPU @ 3.50GHz
Memory Size: 32 GB
Operating System: Ubuntu 18.04.5 LTS
Kernel Version: 4.15.0-154-generic
Accelerator: None
CPU Utilization: 20.0%
Available Disk Space: 30 GB
Starting installation
Downloading modules...
Downloading component Docker_Community_Edition_CE
ZIP file for module 5f21392e9e63c9002a6fd88d already exists. Validating
Module validation passed for 5f21392e9e63c9002a6fd88d
Skipping download...
Downloading component Docker_Compose
ZIP file for module 5f213aae9e63c9002a6fd88e already exists. Validating
Module validation passed for 5f213aae9e63c9002a6fd88e
Skipping download...
Downloading modules completed...
Modules to be installed by package are ['Docker_Community_Edition_CE',
```

Export the Package for Installation

The edgesoftware CLI lets you package the installed modules, customer applications, and dependencies as part of a package. The export is provided in a `.zip` file that includes installation scripts, XML files, and an edgesoftware Python* executable.

Command:

```
./edgesoftware export
```

Uninstall the Packages

The edgesoftware CLI lets you uninstall the complete package or individual components from the package.

To uninstall an individual package, run the following command:

```
./edgesoftware uninstall <MODULE_ID>
```

To uninstall all the packages, run the following command:

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

NOTE This command will not uninstall Docker* Compose and Docker Community Edition CE.

```

Activities X-terminal-emulator
Removing image : ['701e69b4200e', '2.4.2', 'ia_imagestore']
Successfully removed : 701e69b4200e
Removing image : ['f36fc23ff1a0', '2.4.2', 'ia_factoryctrl_app']
Successfully removed : f36fc23ff1a0
Removing image : ['316fb490b6e6', '2.4.2', 'ia_etcd_ui']
Successfully removed : 316fb490b6e6
Removing image : ['372c21a35e12', '2.4.2', 'ia_eis_zmq_broker']
Successfully removed : 372c21a35e12
Removing image : ['d406ae3f36d7', '2.4.2', 'ia_azure_simple_subscriber']
Successfully removed : d406ae3f36d7
Removing image : ['19fcc801490d', '2.4.2', 'ia_eis_azure_bridge']
Successfully removed : 19fcc801490d
Removing image : ['14268137ca91', '2.4.2', 'ia_openvino_base']
Successfully removed : 14268137ca91
Removing image : ['0ea8b0e9f9c3', '2.4.2', 'ia_video_common']
Successfully removed : 0ea8b0e9f9c3
Removing image : ['59f9bdcc25a5', '2.4.2', 'ia_common']
Successfully removed : 59f9bdcc25a5
Removing image : ['6e2fe991a95f', '2.4.2', 'ia_eisbase']
Successfully removed : 6e2fe991a95f
Removing image : ['ca19c6eb38da', '2.4.2', 'ia_etcd_provision']
Successfully removed : ca19c6eb38da
Removing image : ['38edf939df1f', '2.4.2', 'ia_etcd']
Successfully removed : 38edf939df1f
Removing Images [.....]
Removing Volumes [.....]
Removing Networks [.....]
Removing Edge Insights Industrial User [.....]
Removing Edge Insights Industrial directory [.....]
Successfully uninstalled eii_installer took 1 minutes 38.09 seconds
Uninstalling Docker_Compose
Module does not support uninstall
Uninstalling Docker_Community_Edition_CE
Module does not support uninstall
Uninstall Finished
+-----+-----+-----+
| Id | Module | Status |
+-----+-----+-----+
| 5face41098a1ef002af292fd | eii installer | SUCCESS |
| 5f213aae9e63c9002a6fd88e | Docker Compose | NOT SUPPORTED |
| 5f21392e9e63c9002a6fd88d | Docker Community Edition CE | NOT SUPPORTED |

```

Set Up Windows* Subsystem for Linux* (Optional)

You can use the Windows* Subsystem for Linux* (WSL) if you don't want to install a standalone Ubuntu* machine.

Prerequisites

- Minimum Windows 10 is required.
To check your windows version, press the **Windows logo key + R**. Type `winver` and select **OK**.
- You must have at least 40 GB storage available.

Procedure

- If your Windows 10 version is older than 2004, you must do the first 5 steps in this Microsoft* guide before continuing:

<https://docs.microsoft.com/en-us/windows/wsl/install-manual>

Then continue with the steps below.

- If your version is 2004 (Build 19041) or higher, perform the steps below.

1. Open Windows PowerShell*.
2. Run the following command to install the Ubuntu 20.04 distribution:

```
wsl --install -d Ubuntu-20.04
```

As a result, Ubuntu 20.04 will start and in a few minutes a pop-up window will ask you for username and password. After the install ends, the pop-up window is a Ubuntu 20.04 terminal.

You must follow the instructions in the order shown to complete the installation successfully.

3. If your machine is behind a proxy network, make sure you set proxies for `apt` and for the system. If your machine is not behind a proxy, you can skip this step.

NOTE Replace the http/https links and ports in the commands presented below with your proxies links and ports.

- a. Set up apt proxies:

```
sudo touch /etc/apt/apt.conf.d/proxy.conf
sudo echo "Acquire::http::proxy \"http://http_proxy_link:http_proxy_port/\";" | sudo tee -
a /etc/apt/apt.conf.d/proxy.conf
sudo echo "Acquire::https::proxy \"http://https_proxy_link:https_proxy_port/\";" | sudo tee -
a /etc/apt/apt.conf.d/proxy.conf
```

- b. Add your proxy settings to `~/.bashrc` and to `/etc/environment` as well for http, https and no_proxy.

```
http_proxy=http://<http_proxy_link>:<http_proxy_port>
https_proxy=http://<https_proxy_link>:<https_proxy_port>
no_proxy=localhost,127.0.0.1/8, hostIp
```

- c. Reboot from the Windows PowerShell by using the following steps:

- a. Open another Windows PowerShell as administrator and run the following command:

```
Get-Service LxssManager | Restart-Service
```

- b. In the user Windows PowerShell, run the following command to relaunch Ubuntu:

```
wsl -d Ubuntu-20.04
```

- c. At this point, Windows PowerShell will give access to Ubuntu and the prompt should look like this:

```
user@<hostname>:/mnt/c/Users/User
```

4. Update your Ubuntu distribution with the command:

```
sudo apt-get update && sudo apt-get upgrade -y
```

5. Install git and apply systemctl required changes:

```
sudo apt-get install git
git clone https://github.com/DamionGans/ubuntu-wsl2-systemd-script
cd ubuntu-wsl2-systemd-script
sudo ./ubuntu-wsl2-systemd-script.sh
```

6. Install terminator to have it as default terminal:

```
sudo apt-get install terminator
```

7. Set up Windows RDP in order to cover the Reference Implementation prerequisites where you need GUI.

- a. Install xfce with the command:

```
sudo apt-get install -y xfce4 xfce4-goodies
```

- b. Install xrdp with the command:

```
sudo apt-get install -y xrdp
```

- c. Configure xrdp with the following commands:

- a. Comment the last two lines, add startxfce4 at the end of the following file, and save:

```
sudo nano /etc/xrdp/startwm.sh
```

- b. Copy the original ini file as bak format:

```
sudo cp /etc/xrdp/xrdp.ini /etc/xrdp/xrdp.ini.bak
```

- c. Change the xrdp port to not have a conflict with Windows OS port and complete configuration by running the following commands:

```
sudo sed -i 's/3389/3390/g' /etc/xrdp/xrdp.ini
sudo sed -i 's/max_bpp=32/#max_bpp=32\nmax_bpp=128/g' /etc/xrdp/xrdp.ini
sudo sed -i 's/xserverbpp=24/#xserverbpp=24\nxserverbpp=128/g' /etc/xrdp/xrdp.ini
echo xfce4-session > ~/.xsession
```

- d. Enable xrdp

```
sudo systemctl start xrdp
sudo /etc/init.d/dbus start
sudo /etc/init.d/xrdp start
```

8. Install missing component with the command:

```
sudo apt-get install -y zenity
```

9. Open Remote Desktop Connection Windows application and type localhost:3390 for "Computer" and connect. Login with the username and password from step 1.

10. After login, click on **Use Default Config**.

11. Set up for Reference Implementations.

- a. Run source /etc/environment if you are behind a proxy.

- b. Install your preferred browser. This example uses Firefox*.

```
sudo apt-get install -y firefox
```

- c. Create a Docker group and add your user to the group with the commands:

```
sudo groupadd docker
sudo usermod -aG docker $USER
```

- d. Log out and reconnect via the Remote Desktop Connection Windows application.

12. Open a terminal and start a web browser in the background. This example uses Firefox.

- a. Enter the command:

```
firefox &
```

- b. Open the <RI link>.
c. Download the recommended archive.
d. In the terminal, run the command:

```
cp <archive> ~/
```

- e. Unzip the archive and install.

NOTE To delete the distro, use the command:

```
wsl --unregister Ubuntu-20.04
```

Known Limitations

1. If you receive the Docker error `toomanyrequests`, you may need to create an account for Docker Hub. After that, login using your credentials:

```
sudo docker login
```

2. GPU and VPU are not available in WSL. The RI can be launched only on CPU.
3. Disable screen lock on WSL to avoid lockup:

```
gsettings set org.gnome.desktop.screensaver lock-enabled false
```

Reference Implementations

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

To run your first inference application using Edge Insights for Fleet, try one of these reference implementations:

- [Vehicle Event Recording](#)
- [Driver Behavior Analytics](#)
- [Cargo Management](#)
- [Public Transit Analytics](#)
- [Automated License Plate Recognition](#)
- [Address Recognition and Analytics](#)
- [Work Zone Analytics](#)
- [Road Sign Detection and Classification](#)

Set Up ThingsBoard* Cloud Data

To access the cloud data feature, you need to create a cloud ThingsBoard* server using your AWS Elastic Compute Cloud version 2 Instance machine that will provide the possibility to have application monitoring dashboard.

The reference implementations can connect with the ThingsBoard service to send data such as alerts, Driver's drowsiness, driver name or driving mode.

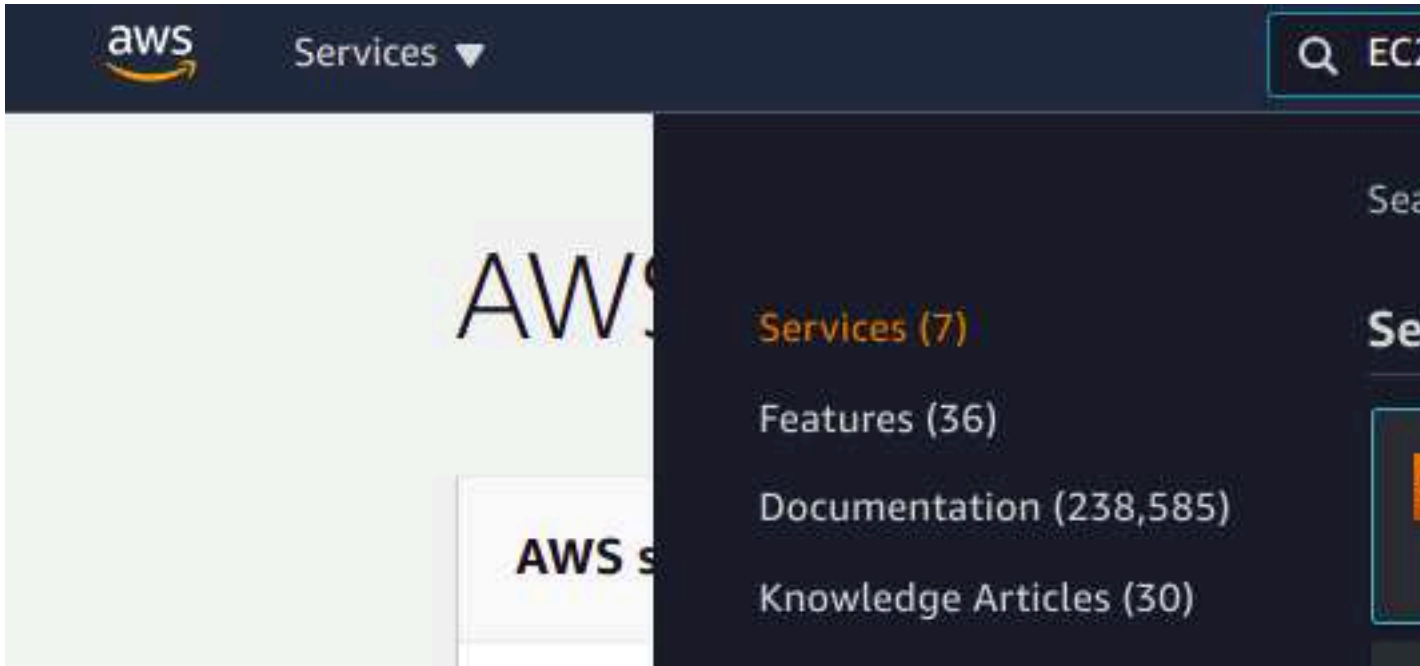
If you have done the AWS S3 Bucket setup you will be able to see and play the videos uploaded by the Reference Implementation.

At the end of the setup you will have Thingsboard_Access_Token, Thingsboard_link_or_ip and Thingsboard_port to be used on on your Edge Insights for Fleet Reference Implementation Cloud Data - Configuration.

To enable this feature you will need to install the package on your machine or on local area network machine.

Set up AWS EC2 Instance

1. Login on the [AWS console](#) and search the EC2 service:



2. Click on the **Launch instances** button to create a new EC2 instance:



3. Search and select an instance with Ubuntu 20.04:

Recents

Quick Start

Amazon Linux
aws

Ubuntu
ubuntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Linux
SUSE

Search

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type
ami-0c4f7023847b90238 (64-bit (x86)) / ami-0d70a59d7191a8079 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Canonical, Ubuntu, 20.04 LTS, amd64 focal image build on 2022-04-19

Architecture

AMI ID

64-bit (x86)

ami-0c4f7023847b90238

4. Select an instance with 2GB RAM:

[1. Choose AMI](#)[2. Choose Instance Type](#)[3. Configure Instance](#)[4. Add Storage](#)

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. To get the most out of Amazon EC2, choose an appropriate mix of resources for your applications. [Learn more](#) about instance type

Filter by:

All instance families

Current generation

[Show/Hide](#)

Currently selected: t2.small (- ECUs, 1 vCPUs, 2.5 GHz, -, 2 GiB memory, EBS

Note: The vendor recommends using a **m5.large** instance (or larger) for the best

	Family	Type	
<input type="checkbox"/>	t2	t2.nano	
<input type="checkbox"/>	t2	t2.micro Free tier eligible	
<input checked="" type="checkbox"/>	t2	t2.small	

5. Enable auto-assign public IP:

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances.

Number of instances	<input type="text" value="1"/>
Purchasing option	<input type="checkbox"/> Request Spot instances
Network	(default)
Subnet	No preference (default subnet)
Auto-assign Public IP	Enable
Placement group	<input type="checkbox"/> Add instance to placement group
Capacity Reservation	Open
Domain join directory	No directory
IAM role	None
Shutdown behavior	Stop
Stop - Hibernate behavior	<input type="checkbox"/> Enable hibernation as an alternative
Enable termination protection	<input type="checkbox"/> Protect against accidental termination
Monitoring	<input checked="" type="checkbox"/> Enable CloudWatch detailed monitoring. Additional charges apply.
Tenancy	Shared - Run a shared hardware tenancy. Additional charges will apply.

6. Assign a storage size:

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can modify the settings of the root volume. You can also attach additional EBS volumes after instance launch using the [Amazon EC2 console](#) options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ
Root	/dev/sda1	

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) storage with usage restrictions.

7. Configure a security group and add port 8080 TCP in Inbound Rules in the Security Group of cloud instance. It can also be configured when the instance is created.

[1. Choose AMI](#)[2. Choose Instance Type](#)[3. Configure Instance](#)[4. Add Storage](#)

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. You can create a new security group or select an existing one. You can also restrict access to the HTTP and HTTPS ports. You can create a new security group or select an existing one.

Assign a security group: ☐ Create a new security group
☒ Select an **existing** security group

8. Review the instance and launch it:



9. Download your access file. Save it on your working environment with the name `access.pem`. It will not be possible to download it later.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you use to connect to your instance securely. For Windows AMIs, the private key file is used to obtain the password used to log into your instance. For Linux AMIs, the private key file is used to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pairs.

Note: The selected key pair will be added to the set of keys authorized for this instance profile. For more information, see [about removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

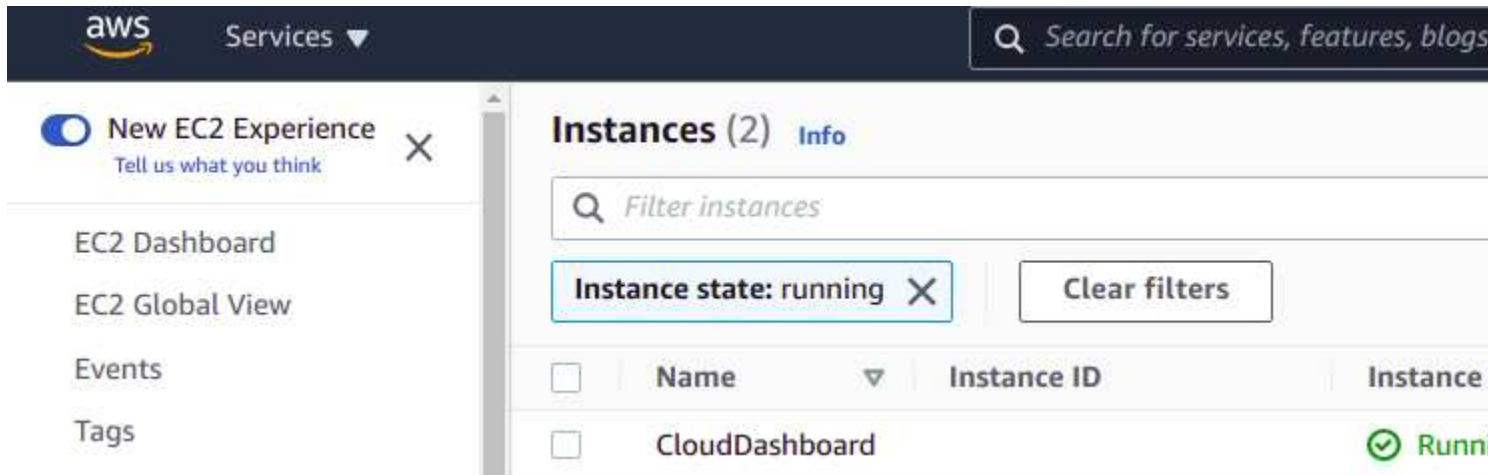
RSA

☒ I acknowledge that I have access to the corresponding private key file, and that if I lose the private key file, I won't be able to log into my instance.

Cancel

10. After a few minutes, the instance will be generated.

Go back to the EC2 panel. The instance state will be Running.



Installation

1. Connect to the AWS EC2 instance using ssh and the key you have saved:

```
ssh -i "your_key.pem" ubuntu@your_ec2_instance_dns_link
```

2. Configure the Ubuntu machine by running the following commands:

```
sudo apt-get update && sudo apt-get upgrade
```

3. Install Docker and Docker Compose.

- Get gpg key for docker binaries and add the apt repository to your apt source list.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor-o=/usr/share/keyrings/docker-archive-keyring.gpg
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

4. From the ssh terminal window, create a folder called thingsboard:

```
mkdir thingsboard
```

5. Change directory to thingsboard:

```
cd thingsboard
```

6. Using your preferred file editor, create a file called docker-compose.yml and add the following contents:

```
version: '2.2'
services:
  mytb:
    restart: always
    image: "thingsboard/tb-postgres"
    ports:
      - "8080:9090"
      - "1883:1883"
      - "31000:7070"
      - "5683-5688:5683-5688/udp"
      - "31001:5432"
    environment:
      TB_QUEUE_TYPE: in-memory
```

```
SSL_ENABLED: "true"
SSL_CREDENTIALS_TYPE: "PEM"
SSL_PEM_CERT: /config/server.pem
SSL_PEM_KEY: /config/server_key.pem
SSL_PEM_KEY_PASSWORD: secret
volumes:
- ~/.mytb-data:/data
- ~/.mytb-logs:/var/log/thingsboard
- ~/.mytb-config:/config
```

7. Create folders for the ThingsBoard database and logs:

```
mkdir ~/.mytb-data
mkdir ~/.mytb-logs
mkdir ~/.mytb-config
```

8. Change user and group permissions to let ThingsBoard access those two folders:

```
sudo chown 799:799 ~/.mytb-data
sudo chown 799:799 ~/.mytb-logs
sudo chown 799:799 ~/.mytb-config
```

9. Generate self-signed certificates to enable the ThingsBoard https feature:

```
# Generate Certificate
openssl ecparam -out server_key.pem -name prime256v1 -genkey
# Generate the key for the certificate
openssl req -new -key server_key.pem -x509 -nodes -days 365 -out server.pem
```



```
~/test$ openssl ecparam -out server_key.pem -name prime256v1 -genkey
~/test$ openssl req -new -key server_key.pem -x509 -nodes -days 365 -out
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []:CA
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your_Org
Organizational Unit Name (eg, section) []:YO
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:test_tb@your_org.com
~/test$ |
```

10. Copy the files to the `.mytb-config` folder:

```
sudo cp server*.pem ~/.mytb-config/
```

11. Pull the Docker Hub image with the command: (If your user is not in a Docker group, use `sudo` in the command.)

```
docker-compose -f docker-compose.yml pull
```

12. Start the ThingsBoard server with the command:

```
docker-compose -f docker-compose.yml up -d
```

13. In about 2-3 minutes, you should be able to access the ThingsBoard page.

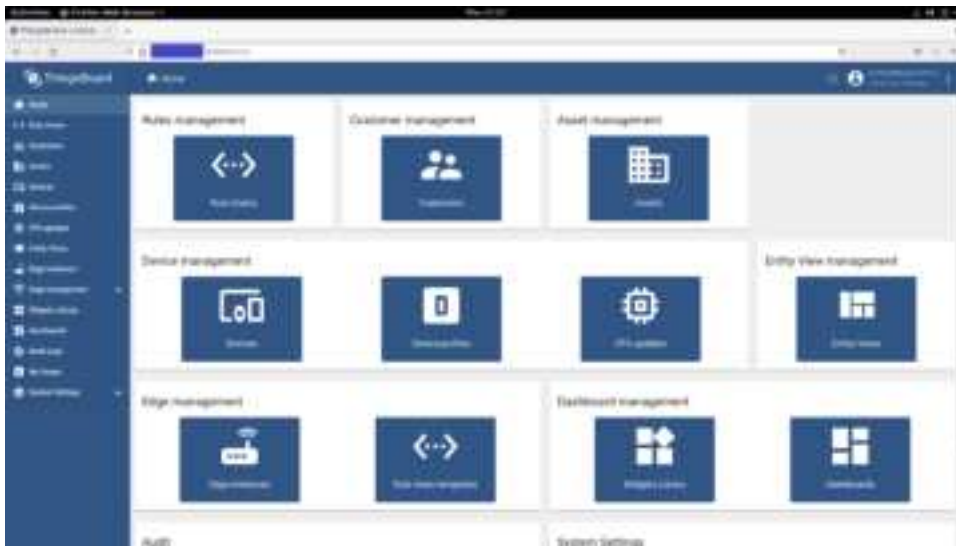
- a. Go to your preferred browser and access: `https://<aws_ec2_dns>:8080/login`
- b. Due to the self-signed certificate, a warning will occur from your preferred browser. Click on **Advanced** and click to view certificate. Download it locally and accept the connection.
- c. Copy the downloaded certificate to Edge Insights for Fleet local storage using the following command:

```
sudo cp <your_pem_file>.pem /opt/intel/eii/local_storage/server_pub_tb.pem
```

- d. Log in with the default user and password set up by the public Docker image:

```
User: tenant@thingsboard.org
```

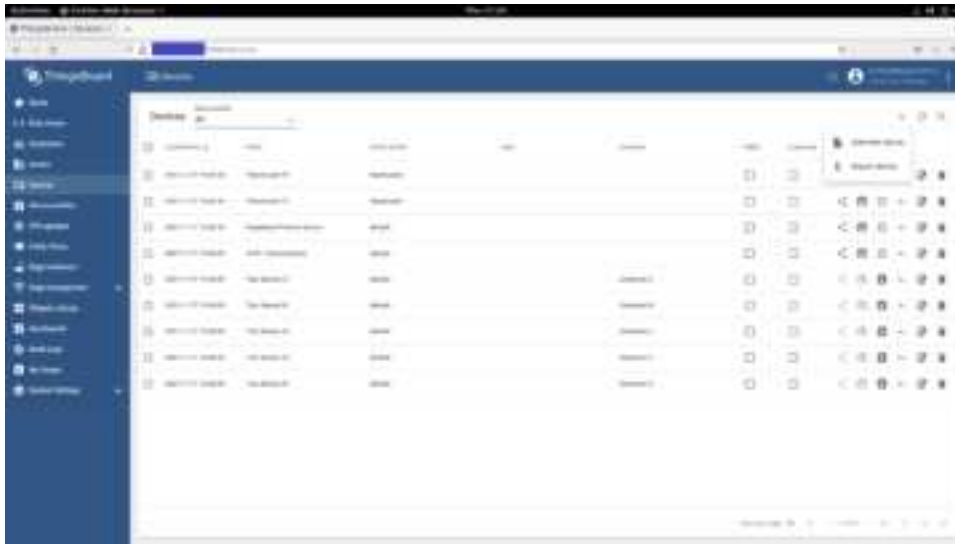
```
Password: tenant
```



ThingsBoard* Cloud Setup

Devices and device profile setup:

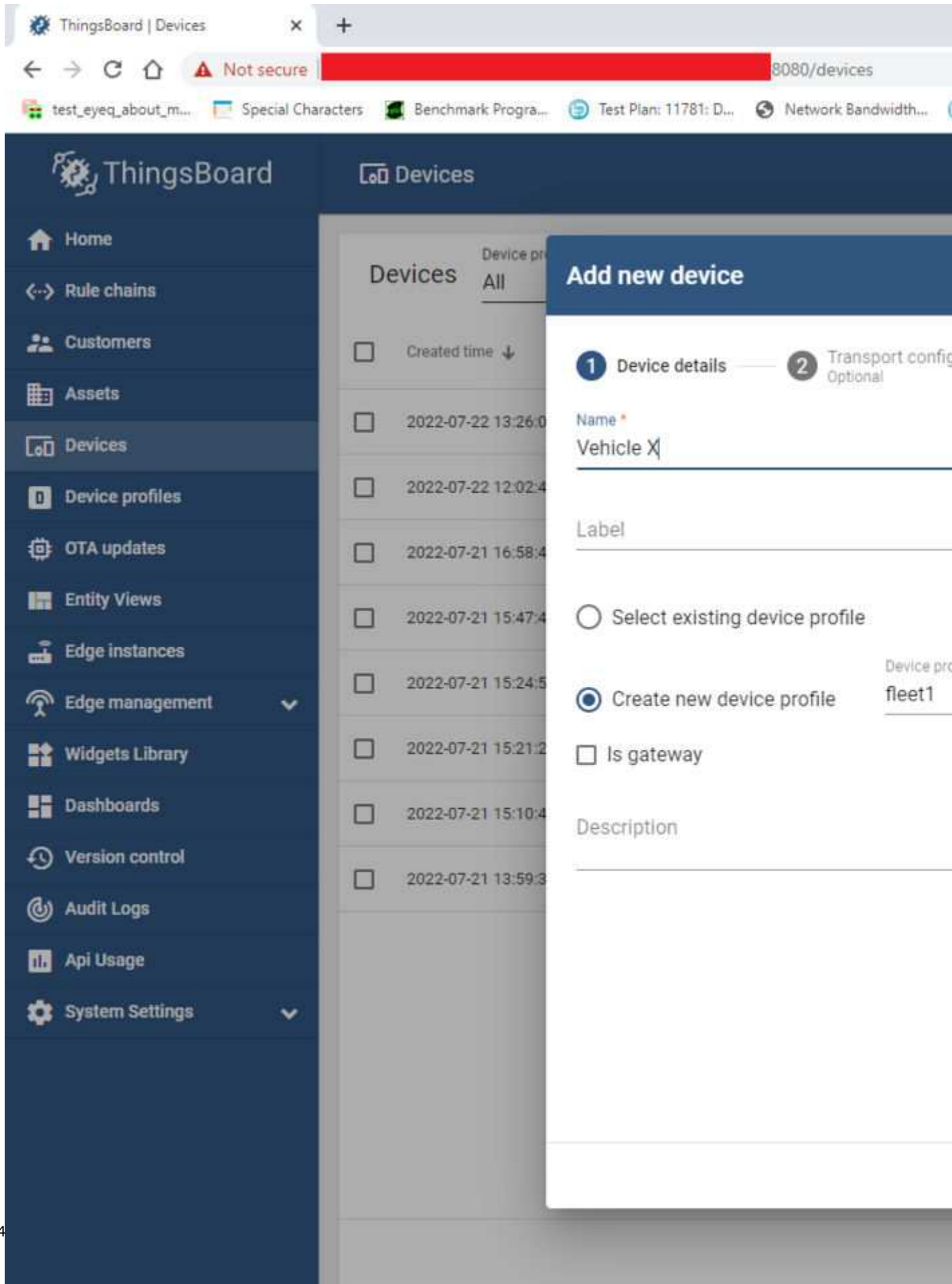
1. First we need to create the Device and Device Profile.
2. Open the **Devices** tab from the main page or from the list on the left.



- Click on **+** and then click on **Add new device**.
- A pop-up window with **Add new device** title is displayed.
- If this is the first device, select **Create new device profile**.

Enter the Device name using the format `Vehicle <Name>`, where <Name> is user-defined.

NOTE Make sure that the Device profile name includes `fleet`, for example: `fleet1` or `fleet`.



6. Click on **Credentials** to open the Credentials tab.
7. Click on **Add credentials options**.
8. Set up a token for this device. Save this token to use on the Reference Implementations you have installed on the **Cloud Data -> Configuration** tab.
9. Click on **Add** to finish.

NOTE If you want to have multiple **Vehicle** devices for Edge Insights for Fleet Reference Implementations, you just need to create a new one, assign the **fleet** device profile to it and set an access token.

Create your ThingsBoard* Environment

This setup will provide the steps to import the templates required for your ThingsBoard Cloud Setup.

If you have the AWS* Cloud Storage setup, you should complete the **Cloud Data -> Configuration** in order to have your **Thingsboard** Server connected with AWS. This allows access to multimedia files (videos or snapshots) uploaded by the Reference Implementation on your AWS S3 Bucket server.

1. Open the **Rule Chains** page from the main page or from the list on the left.
2. Click on **+** and select **Import rule chain**.
3. A pop-up window will appear to drop the json file or click to select the file.
4. Click on the area and in your installation folder, go to the following path:

```
<install_path>/<package_name>/<package_name><version>/<use_case_folder>/src/Dashboard/imports/
```

For example:

```
/home/intel/driver_behavior_analytics/Driver_Behavior_Analytics_2021.4/Driver_Behavior_Analytics/  
EII-DriverBehavior-UseCase/src/Dashboard/imports/
```

5. Select the **1-RuleChain.json** file and click on the **Import** button.
6. Click on the **Verified mark** button to acknowledge the rule chain.

The screenshot shows the ThingsBoard Rule Chain editor. The left sidebar contains a navigation menu with the following items: Home, Rule chains, Customers, Assets, Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management, Widgets Library, Dashboards, Version control, Audit Logs, Api Usage, and System Settings. The main panel is titled 'Rule chains' and 'Intel Root Rule Chain'. It features a search bar and a filter dropdown. The filter dropdown is currently open, showing two categories: 'Filter' and 'Enrichment'. The 'Filter' category contains the following nodes: check alarm status, check existence fields, check relation, gps geofencing filter, message type, message type switch, originator type, originator type switch, script, and switch. The 'Enrichment' category contains the following nodes: calculate delta, customer attributes, customer details, and originator attributes. On the right side of the screen, a partial view of a rule chain diagram is visible, showing an 'Input' node connected to a 'script check repeated alerts' node.

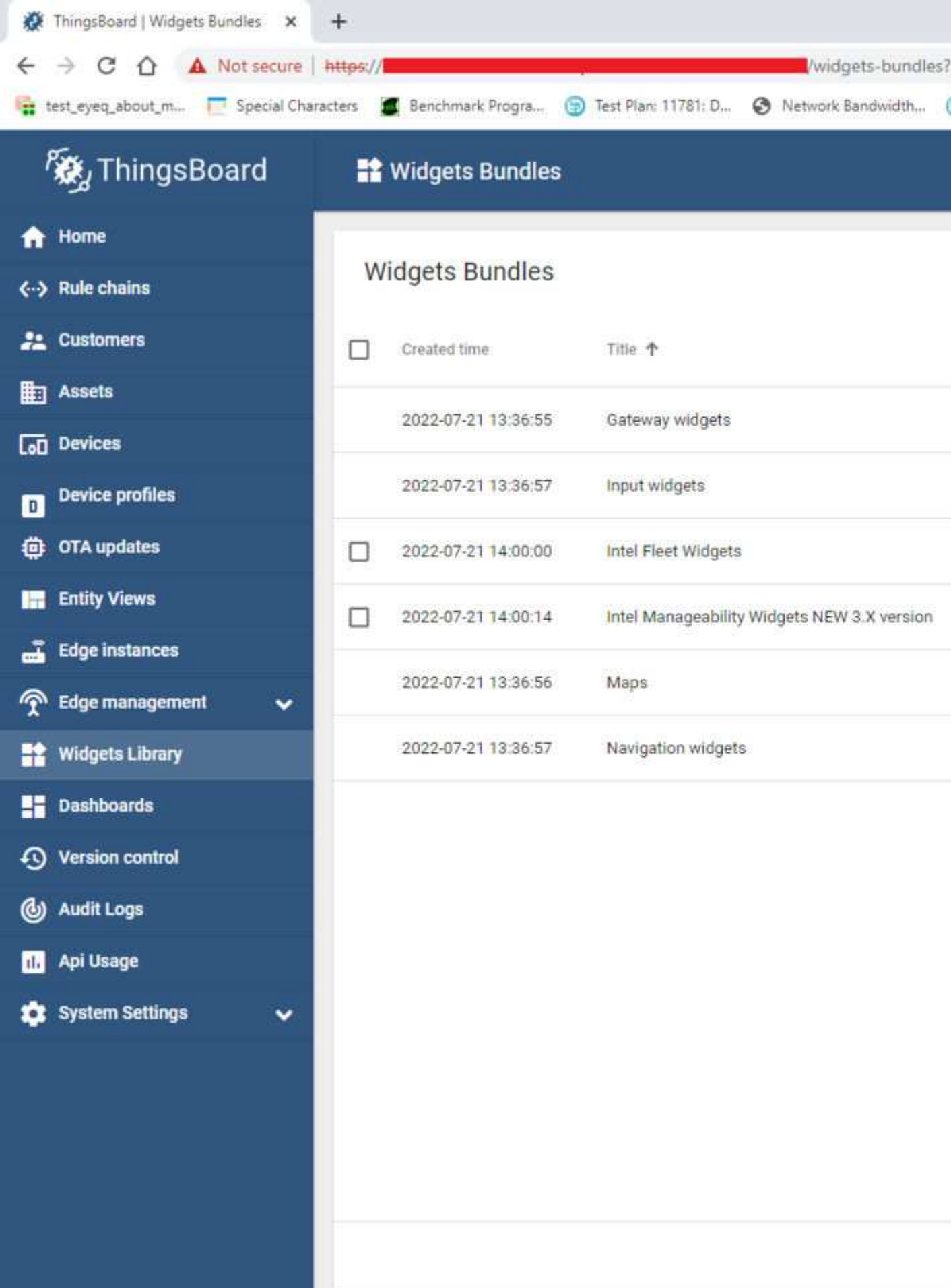
7. Click on the **Widgets Library** tab button.
8. Click on **+** and select **Import widgets bundle**.
9. A pop-up window will appear to drop the json file or click to select the file.
10. Click on the area and in your installation folder, go to the following path:

```
<install_path>/<package_name>/<package_name><version>/<use_case_folder>/src/Dashboard/imports/
```

For example:

```
/home/intel/driver_behavior_analytics/Driver_Behavior_Analytics_2021.4/Driver_Behavior_Analytics/  
EII-DriverBehavior-UseCase/src/Dashboard/imports/
```

11. Select the 2-WidgetsA-intel_fleet_widgets.json file and click on the **Import** button.
12. Select the 3-WidgetsB-intel_manageability_widgets_new_3_x_version.json file and click on the **Import** button.



The screenshot shows the ThingsBoard web interface. The left sidebar contains the following menu items: Home, Rule chains, Customers, Assets, Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management, Widgets Library (selected), Dashboards, Version control, Audit Logs, Api Usage, and System Settings. The main content area is titled 'Widgets Bundles' and displays a table with the following data:

<input type="checkbox"/>	Created time	Title ↑
	2022-07-21 13:36:55	Gateway widgets
	2022-07-21 13:36:57	Input widgets
<input type="checkbox"/>	2022-07-21 14:00:00	Intel Fleet Widgets
<input type="checkbox"/>	2022-07-21 14:00:14	Intel Manageability Widgets NEW 3.X version
	2022-07-21 13:36:56	Maps
	2022-07-21 13:36:57	Navigation widgets

13. Open the **Dashboards** page from the main page or from the list on the left.
14. Click on **+** and select **Import Dashboard**.



15. A pop-up window will appear to drop the json file or click to select the file.
16. Click on the area and in your installation folder, go to the following path:

```
<install_path>/<package_name>/<package_name><version>/<use_case_folder>/src/webui/templates
```

For example:

```
/home/intel/driver_behavior_analytics/Driver_Behavior_Analytics_2021.3/Driver_Behavior_Analytics/  
EII-DriverBehavior-UseCase/src/webui/templates/
```

17. Select the 5-Dashboard-v2.json file and click on the **Import** button.
18. The Intel Fleet Manager Dashboard should be visible on your dashboards list.
19. To open a dashboard, click on the following icon:



20. Your dashboard should automatically map the vehicle device you have previously created. To activate the Reference Implementation monitoring, you will need to complete the **Cloud Data -> Configuration** and **Run the Use Case**.

ThingsBoard | Dashboard

Not secure | https://[redacted]/dashboards/14a6[redacted]

test_eyeq_about_m... Special Characters Benchmark Progra... Test Plan: 11781: D... Network Bandwidth...

ThingsBoard

Dashboards > Intel Fleet Manager v2.0

Fleet

Fleet List

LABEL ↑	driver_name	en
Vehicle A	Mike	ON
Vehicle B	Unknown	ON
Vehicle C	Unknown	ON

Items per page: 10

Alarms

Realtime - last day

Created time ↓	Originator	Type
No alarms found		

Items per page: 10

21. On the dashboard, click on the device you configured for your Reference Implementation.

For example, if you added the **Vehicle A** access token to your RI and you started the use case, then click on **Vehicle A** inside the dashboard in order to open the vehicle page that will provide the detection data and the simulated data live.

ThingsBoard | Dashboard

Not secure | https://[redacted]:8080/dashboards/14a6...

test_eyeq_about_m... Special Characters Benchmark Progra... Test Plan: 11781: D... Network Bandwidth...

ThingsBoard

Dashboards > Intel Fleet Manager v2.0

Fleet > Vehicle Details Vehicle A

Home Rule chains Customers Assets Devices Device profiles OTA updates Entity Views Edge instances Edge management Widgets Library Dashboards Version control Audit Logs Api Usage System Settings

Running

Use Case

Vehicle

Label	Fuel Level	Speed	Driving Mode	Heading
		 46.60		 E 90.25

FUEL_LEVEL

100% 100

0

SPEED

0 47 180

MPH

Temperature

Engine temperature	Cabin temperature	Cargo ar
normal	63.18 °F	66.89

Details

Troubleshooting

If you don't have access to the ThingsBoard web page, you may need to change the port forwarding inside the `docker-compose.yml` file.

Follow these steps:

1. Run the following command to stop ThingsBoard server:

```
docker-compose -f docker-compose.yml down
```

2. Modify the `docker-compose.yml` file with your port forwarding setup. The left side ports are used on your host.

```
- "8080:9090"  
- "1883:1883"  
- "7070:7070"  
- "5683-5688:5683-5688/udp"  
- "30000:5432"
```

For example, if port 8080 is blocked, your new port forwarding configuration for this port could be:

```
- "30300:9090"
```

The server will access 8080 port inside the Docker container and it will forward the data to port 30300 on your host machine.

3. Save the file and re-run the command to start the server:

```
docker-compose -f docker-compose.yml up -d
```

Set Up Amazon Web Services* Cloud Storage

To enable Cloud Storage on the installed Reference Implementation, you will need Amazon Web Services* (AWS*) paid/free subscription to enable your root user account that has to support the following services:

- Identity and Access Management (IAM)
- Amazon S3 Bucket

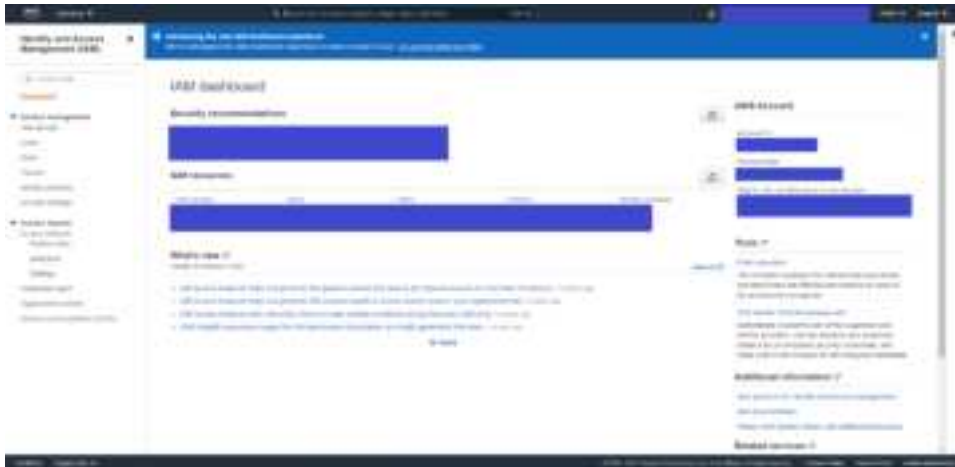
After finishing the setup for IAM and S3, you will have your `AWS_ACCESS_KEY`, `AWS_SECRET_ACCESS_KEY` and `AWS_S3_BUCKET_NAME` to be used on your Edge Insights for Fleet Reference Implementation Cloud Data - Configuration.

References

- [AWS IAM Official Documentation](#)
- [AWS IAM Create and Setup Official documentation](#)

Setup Steps

1. From your AWS management console, search for IAM and open the IAM Dashboard.



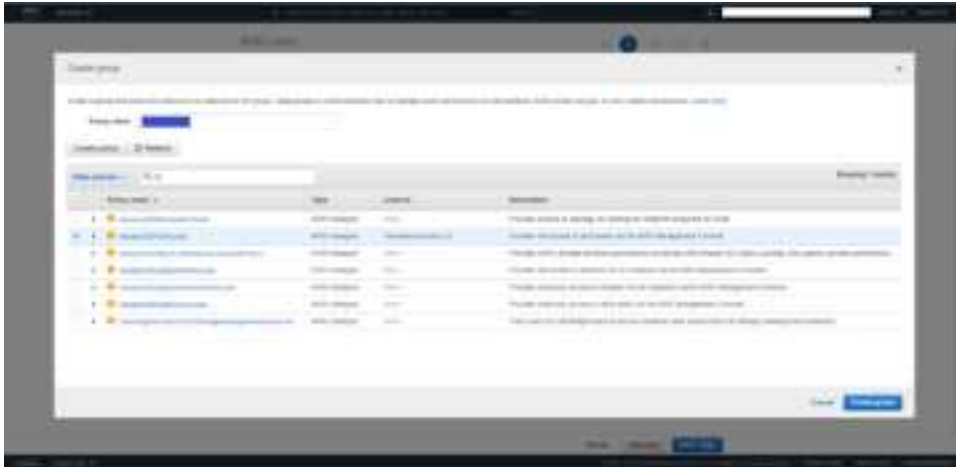
2. On the left menu of the dashboard, go to **Access management** and click on **Users** to open the IAM Users tab.



3. From the IAM users tab, click on **Add User** to access the AWS add user setup.
4. On the first tab, provide the username and select the AWS credentials type to be **Access key**.



5. On the second tab, create a group to attach policies for the new IAM user.
 - a. Search for S3 and select **AmazonS3FullAccess** policy.
 - b. Click on **Create group**.



6. Select the group you have created and click on **Next: Tags**.
7. Tags are optional. If you don't want to add tags, you can continue to the Review tab by clicking on **Next: Review**.
8. After review, you can click on the **Create User** button.
9. On this page, you have access to AWS Key and AWS Secret Access key. (Click on **Show** to view them.)
 - a. Save both of them to be used later on your Cloud Data - Configuration on the Edge Insights for Fleet Reference Implementation you have installed.

NOTE The AWS Secret Key is visible only on this page, you cannot get the key in another way.

- b. If you forget to save the AWS Secret Key, you can delete the old one and create another key.



10. After you have saved the keys, close the tab. You are returned to the IAM Dashboard page.
11. Click on the user created and save the **User ARN** to be used on S3 bucket setup.

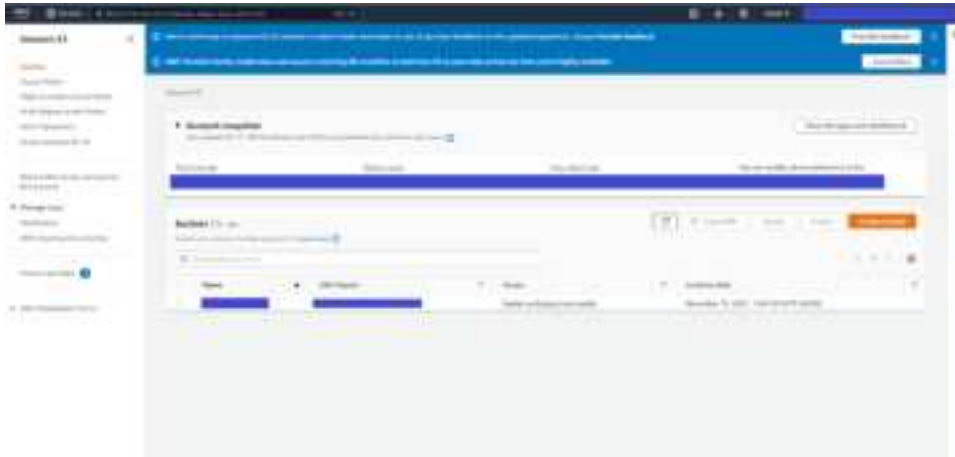
NOTE In case you forgot to save the AWS Secret key from the User tab, you can select **Security Credentials**, delete the Access Key and create another one.

S3 Bucket

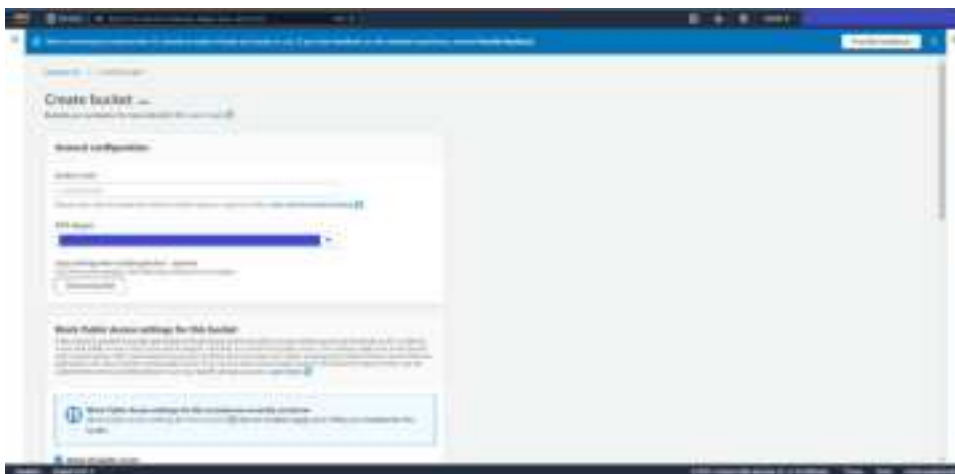
S3 bucket service offers cloud storage to be used on cloud based applications.

S3 Bucket Service setup

1. Open the Amazon Management Console and search for Amazon S3.
2. Click on **S3** to open the AWS S3 Bucket dashboard.



3. On the left side menu, click on **Buckets**.
4. Click on the **Create Bucket** button to open the Create Bucket dashboard.
5. Enter a name for your bucket and select your preferred region.



6. Scroll down and click on **Create Bucket**.
7. From the S3 Bucket Dashboard, click on the newly created bucket and go to the **Permissions** tab.
8. Scroll to **Bucket Policy** and click on **Edit** to add a new statement in statements tab that is already created to deny all the uploads.



9. You must add a comma before adding the following information.

```
{
  "Sid": "<Statement name>",
  "Effect": "Allow",
  "Principal": {
    "AWS": "<User_ARN_Saved>"
  },
  "Action": "s3:*",
  "Resource": [
    "arn:aws:s3:::<bucket_name>",
    "arn:aws:s3:::<bucket_name>/*"
  ]
}
```

- Update with the following statement with statement name, your user ARN saved at IAM setup - step 11 and your bucket name.
- Click on **Save changes**. If the change is successful, you will see a **success saved** message, otherwise you need to re-analyze the json file to fix the error.

Update an Application-Over-The-Air

Application-Over-The-Air (AOTA) updates enable cloud to edge manageability of application services running on Edge Insights for Fleet (EIF) enabled systems through the Device Manageability component. Device Manageability is a Device Manageability Software which includes SOTA, FOTA, AOTA, and few system operations.

For EIF use case, only AOTA features from Device Manageability are validated and will be supported through ThingsBoard* cloud-based management front-end service.

The following section will walk you through setting up ThingsBoard*, creating and establishing connectivity with the target systems, as well as updating applications on those systems.

NOTE Device manageability was previously named Turtle Creek. Remnants of the previous name still exists in some components. The name replacement is ongoing and will be completed in future releases.

Install Device Manageability Functionality

You need 2 hosts to run AOTA. One host is the server on which an EIF Reference Implementation will be installed, and one host is the worker on which the Reference Implementation will be deployed through AOTA for execution.

In this guide, the server host will be referred to as *Server* and the worker host will be referred to as *Worker*.

Similarly, `$BUILD_PATH` refers to the build ingredients directory of EIF where most of the commands will be executed. The value is: `/opt/intel/eif/IEdgeInsights/build/`

Server Prerequisites

1. Refer to [Requirements](#) for the hardware requirements.
2. Ubuntu* 20.04 LTS
3. One of the EIF [Reference Implementations](#) is installed.

Worker Prerequisites

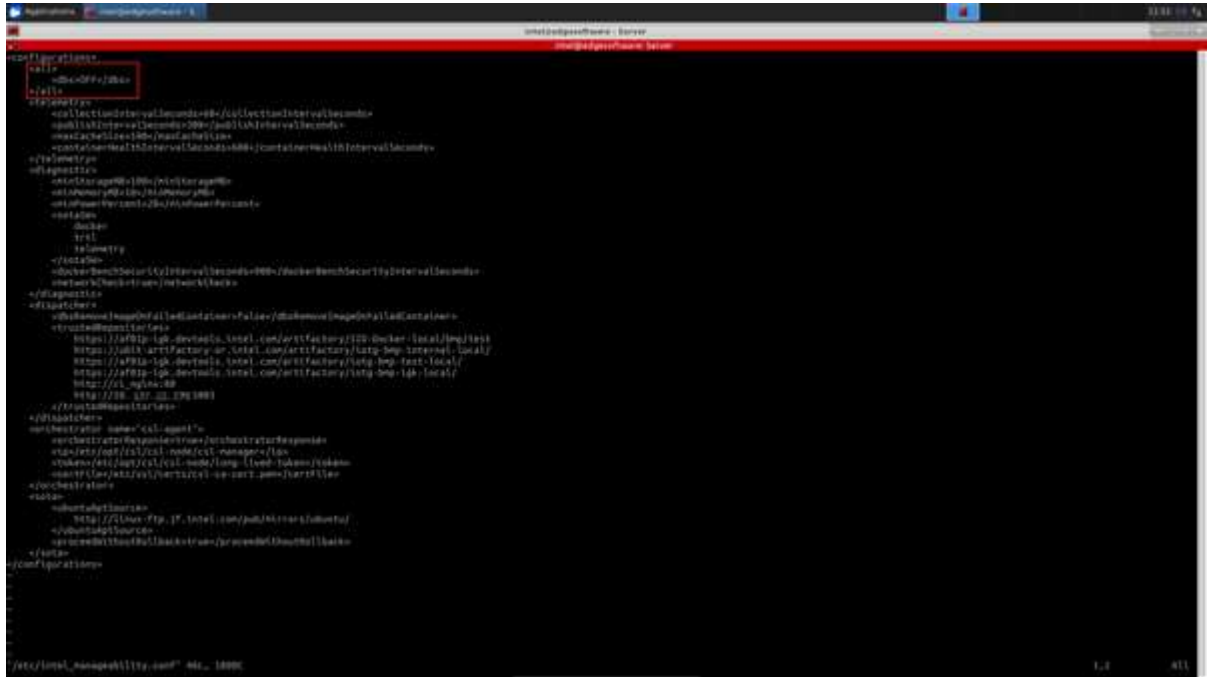
1. Refer to [Reference Implementations](#) for the specific hardware requirements for the worker host.
2. Ubuntu* 20.04 LTS

4. Once Device Manageability has been installed successfully, edit the `/etc/intel_manageability.conf` file and update the required values.

```
sudo vi /etc/intel_manageability.conf
```

- a.** Under the `<all>` `</all>` section, change `dfs` from `ON` to `OFF`.

DBS stands for Docker* Bench Security. This feature of Device Manageability is not used for EIF.



- b.** Add the IP endpoint for the developer files to the trustedRepositories:

```
<trustedRepositories>
  http://[Server_IP]:5003
</trustedRepositories>
```

- c.** Save and exit. You must restart the Worker before these changes can take effect.

Multi-Node Deployment

EIF deployment on multiple nodes requires the use of the Docker* registry. The following sections outline some of the commands to be run on the Server and on any newly added Workers.

Execute the following steps on the Server.

Configure Docker* Registry

1. Launch the local Docker* registry:

```
docker run -d -p 5002:5000 --name registry --restart unless-stopped registry:2
```

2. Configure the Docker* daemon to allow pushing images to the local Docker* registry:

- a.** Edit the `/etc/docker/daemon.json` file:

```
sudo vi /etc/docker/daemon.json
```

- b.** Add the following lines:

```
{
  "insecure-registries": ["<Server_IP>:5002"]
}
```


3. Edit the `eif.json` file and replace the `include_service` list with the list got above, but skip `ia_configmgr_agent` from the list:

```
sudo cp config.json eif.json
sudo vi eif.json
```

4. Generate the `mgr` archive that will be used to launch the Config Manager Agent:

```
sudo mkdir mgr
sudo cp ../eii_config.json mgr/
sudo python3 generate_eii_bundle.py -t mgr -c mgr.json
```

5. Generate the `eif` archive that will be used to launch AOTA:

```
sudo mkdir eif
sudo cp ../eii_config.json eif/
sudo python3 generate_eii_bundle.py -t eif -c eif.json
```

NOTE These commands will generate `mgr.tar.gz` and `eif.tar.gz` archives. Save these archives. They will be served through a Python* HTTP server and ThingsBoard* to the Worker to launch AOTA.

Cloud Service: ThingsBoard* Setup

Follow these instructions to set up ThingsBoard on the Server. It is possible to set up ThingsBoard on a different host, even one that is not part of the cluster, however, this tutorial describes how to deploy it on the AWS.

1. Follow the [Installation section of Set Up ThingsBoard Cloud Data](#).

NOTE ThingsBoard* can also be set up on a local host if AWS is not a viable option.

2. On the ThingsBoard* page, go to **Devices**. On the device list, click on the shield icon to find out the credentials of a device:

Activities Firefox Web Browser

ThingsBoard | Devices

https://[redacted]/devices

ThingsBoard

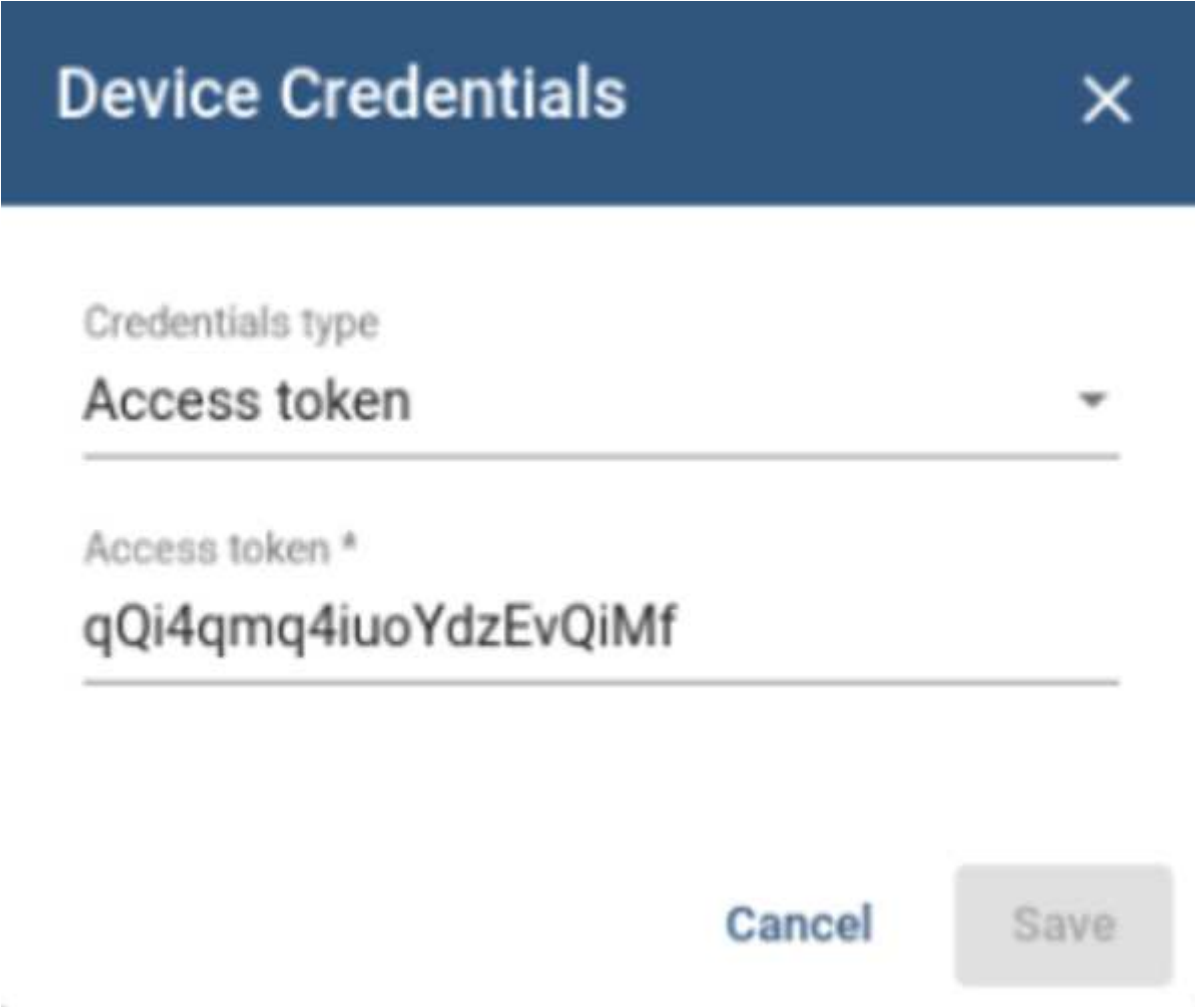
Devices

- Home
- Rule chains
- Customers
- Assets
- Devices**
- Device profiles
- OTA updates
- Entity Views
- Edge instances
- Edge management
- Widgets Library
- Dashboards
- Audit Logs
- Api Usage
- System Settings

Devices Device profile All ×

<input type="checkbox"/>	Created time ↓	Name
<input type="checkbox"/>	2022-09-06 07:46:51	Vehicle-1
<input type="checkbox"/>	2022-09-06 07:35:29	Thermostat T2
<input type="checkbox"/>	2022-09-06 07:35:29	Thermostat T1
<input type="checkbox"/>	2022-09-06 07:35:29	Raspberry Pi Demo Device
<input type="checkbox"/>	2022-09-06 07:35:29	DHT11 Demo Device
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device C1
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device B1
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A3
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A2
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A1

3. Save the credentials for Worker provisioning. In this dialog you have the option to add a custom Access Token instead of using the provided one. Do not forget to click the **Save** button if you modify the Access Token.



Device Credentials

Credentials type

Access token

Access token *

qQi4qmq4iuoYdzEvQiMf

Cancel Save

Worker Provisioning

With the Server and the Cloud Service set up, in this section you will set up the Worker. All the steps and commands in this section must be executed on the Worker.

Docker* Provisioning

1. Configure the Docker* daemon to allow pulling images from Server:

- a. Edit the `/etc/docker/daemon.json` file:

```
sudo vi /etc/docker/daemon.json
```

- b. Add the following lines:

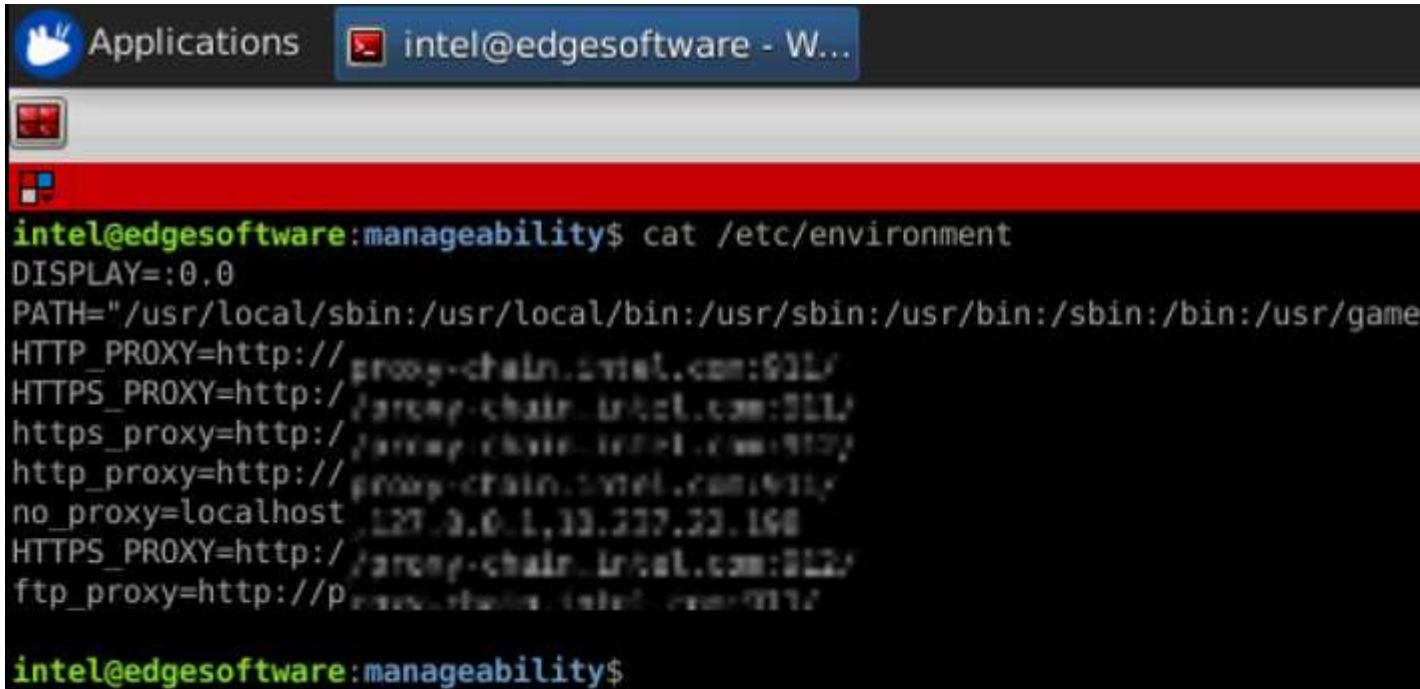
```
{  
  "insecure-registries": ["<Server_IP>:5002"]  
}
```

- c. Restart the Docker service to reload the change:

```
sudo service docker restart
```

ThingsBoard* Provisioning

1. Add the **DISPLAY** variable to the `/etc/environment` file.

A terminal window titled 'intel@edgesoftware - W...' is shown. The terminal output displays the contents of the /etc/environment file, listing various environment variables including DISPLAY, PATH, and several proxy settings (HTTP_PROXY, HTTPS_PROXY, https_proxy, http_proxy, no_proxy, ftp_proxy) pointing to proxy-chain.intel.com. The prompt is intel@edgesoftware:manageability\$.

```
intel@edgesoftware:manageability$ cat /etc/environment
DISPLAY=:0.0
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin"
HTTP_PROXY=http://proxy-chain.intel.com:812/
HTTPS_PROXY=http://proxy-chain.intel.com:812/
https_proxy=http://proxy-chain.intel.com:812/
http_proxy=http://proxy-chain.intel.com:812/
no_proxy=localhost,127.0.0.1,10.227.22.168
HTTPS_PROXY=http://proxy-chain.intel.com:812/
ftp_proxy=http://proxy-chain.intel.com:812/

intel@edgesoftware:manageability$
```

Update the value according to your environment.

2. Launch the provisioning binary:

```
sudo PROVISION_TPM=auto provision-tc
```

3. If the Worker was previously provisioned, the following message will appear. To override the previous cloud configuration, enter `Y`.

```
Applications intel@edgesoftware - W...
Setting Docker-Bench-Security
Enabling and starting mqtt (this may take some time to generate secrets)...
Removed /etc/systemd/system/multi-user.target.wants/diagnostic.service.
Removed /etc/systemd/system/multi-user.target.wants/dispatcher.service.
Removed /etc/systemd/system/multi-user.target.wants/telemetry.service.
Removed /etc/systemd/system/multi-user.target.wants/configuration.service.
Removed /etc/systemd/system/multi-user.target.wants/mqtt.service.
Removed /etc/systemd/system/multi-user.target.wants/cloudadapter.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mqtt.service → /
A cloud configuration already exists: "thingsboard"
Replace configuration? [Y/N] Y

Please choose a cloud service to use.
1: Telit Device Cloud
2: Azure IoT Central
3: ThingsBoard
4: Custom

3

Configuring to use ThingsBoard...

Please enter the server IP:
10.13.23.78

Please enter the server port (default 1883):

Please choose provision type.
1: Token authentication
2: X509 authentication

1

Please enter the device token:
qQi4qmq4iuoYdzEvQiMf

Configure TLS? [Y/N]
```

4. Select ThingsBoard* as the cloud service by entering **3** and **Enter**.
5. Provide the IP of the Server:
6. When asked for the server port, press **Enter** to use the default value **1883**.
7. When asked about provision type, choose **Token Authentication**.
8. Enter the device token extracted in the [Cloud Service: ThingsBoard* Setup](#) section.
9. When asked about **Configure TLS**, enter **N**.
10. When asked about signature checks for OTA packages, enter **N**.

The script will start the Intel® Manageability Services. When the script finishes, you will be able to interact with the device via the ThingsBoard* dashboard.

If at any time the cloud service configuration needs to be changed or updated, you must run the provisioning script again.

Configure Reference Implementation

1. Create the necessary folders and files for the Reference Implementation configuration:

```
sudo mkdir -p /opt/intel/eii/local_storage
sudo touch /opt/intel/eii/local_storage/credentials.env
sudo touch /opt/intel/eii/local_storage/cloud_dashboard.env
sudo mkdir /opt/intel/eii/local_storage/saved_images
sudo mkdir /opt/intel/eii/local_storage/saved_videos
sudo mkdir /opt/intel/eii/local_storage/alert_logs
```

2. Add the AWS* cloud credentials and ThingsBoard* credentials into `credentials.env` and `cloud_dashboard.env`, created above. These are the credentials that you would usually add into the Configuration page of the webpage of the Reference Implementation:

The screenshot shows a web interface titled "Configuration". It contains several input fields for credentials:

- AWS ACCESS_KEY_ID**: A text input field.
- AWS SECRET_ACCESS_KEY**: A text input field.
- AWS_S3_BUCKET_NAME**: A text input field.
- A blue button labeled "Save Credentials" with a lock icon, followed by the text "Saved."
- THINGSBOARD_ACCESS_TOKEN**: A text input field.
- THINGSBOARD_SERVER_LINK_OR_IP**: A text input field.
- THINGSBOARD_PORT**: A text input field.
- A blue button labeled "Save Token" with a lock icon, followed by the text "Saved."

- a. In the `/opt/intel/eii/local_storage/credentials.env` file, add the following variables along with the values you would've added in the webpage for AWS* credentials:

```
AWS_ACCESS_KEY=
AWS_SECRET_ACCESS_KEY=
AWS_BUCKET_NAME=
```

- b.** (Optional) In the `/opt/intel/eii/local_storage/cloud_dashboard.env` file, add the following variables along with the values you would've added in the webpage for ThingsBoard* credentials:

```
HOST=  
PORT=  
ACCESS_TOKEN=
```

This step is optional. It must be executed if you wish to continue receiving updates in the cloud dashboard and have the Reference Implementation save the events on AWS* storage.

- 3.** Add the ThingsBoard* public certificate, used when setting ThingsBoard* up, to `/opt/intel/eii/local_storage` with the command:

```
cp server.pem /opt/intel/eii/local_storage/server_pub_tb.pem
```

Perform AOTA

In the [Multi-Node Deployment](#) section you created the AOTA `mgr.tar.gz` and `eif.tar.gz` bundles.

On the Server, go to the path where those files are located, and, for development purposes only, launch a Python* HTTP server:

```
python3 -m http.server 5003
```

- 1.** Go to ThingsBoard* page, **Dashboard**, and select **Intel Fleet Manager v2.0**:

Activities Firefox Web Browser ▾

ThingsBoard | Dashboard × +

← → ↻ https://[redacted]/dashboards

ThingsBoard

Dashboards

- Home
- Rule chains
- Customers
- Assets
- Devices
- Device profiles
- OTA updates
- Entity Views
- Edge instances
- Edge management ▾
- Widgets Library
- Dashboards**
- Audit Logs
- Api Usage
- System Settings ▾

Dashboards

<input type="checkbox"/>	Created time ↓	Title
<input type="checkbox"/>	2022-09-06 07:48:45	Intel Fleet Manager v2.0
<input type="checkbox"/>	2022-09-06 07:35:29	Gateways
<input type="checkbox"/>	2022-09-06 07:35:29	Firmware
<input type="checkbox"/>	2022-09-06 07:35:29	Software
<input type="checkbox"/>	2022-09-06 07:35:29	Rule Engine Statistics
<input type="checkbox"/>	2022-09-06 07:35:29	Thermostats

2. Select the device that you chose in the [Cloud Service: ThingsBoard* Setup](#) section, then click on the **Trigger AOTA** button:

Activities Firefox Web Browser

ThingsBoard | Dashboard x +

← → ↻ https://[redacted]/dashboards/55aa4d10-2db8-11ed-bb88-5d

ThingsBoard

Dashboards > Intel Fleet Manager v2.0

Fleet > Vehicle Details Vehicle-1

0 0 MPH 180

0

Details

Vin	Engine speed	Gear position	Fuel efficiency

Temperature

Engine temperature	Cabin temperature	Cargo a

Tire pressure

Axis	Outer Left	Inner Left	Inner Right
<hr/>			

Device Manageability ...

- 3.** Once the Trigger AOTA dialog opens, complete each field per the information below:

Activities Firefox Web Browser ▾

ThingsBoard | Dashboard × +

← → ↻ https://[redacted]/dashboards/55aa4d10-2db8-11ed-bb88-5d

ThingsBoard

Dashboards > Intel Fleet Manager v2.0

Fleet > Vehicle Details Vehicle-1

0 MPH 180

0

Details

Vin	Engine speed	Gear position	Fuel efficiency

Temperature

Engine temperature	Cabin temperature	Cargo a

Tire pressure

Axis	Outer Left	Inner Left	Inner Right

Device Manageability ...

App: docker-compose

Command: up

Container Tag: mgr

Fetch: Enter the HTTP server that was set up at the start of this section.

Leave the other sections empty and click on the **Send** button.

In the step above, the Worker will access the Server through the local HTTP server to fetch the mgr bundle.

In the device telemetry log, you can see that mgr was fetched from the local server and was deployed successfully:

Activities Firefox Web Browser

ThingsBoard | Dashboard x ThingsBoard | Devices x +

← → ↻ https://192.168.1.100:8080/devices

ThingsBoard

Devices

Home

Rule chains

Customers

Assets

Devices

Device profiles

OTA updates

Entity Views

Edge instances

Edge management

Widgets Library

Dashboards

Audit Logs

Api Usage

System Settings

Devices

Device profile All

<input type="checkbox"/>	Created time ↓	Name	Device profile
<input type="checkbox"/>	2022-09-06 07:46:51	Vehicle-1	fleet
<input type="checkbox"/>	2022-09-06 07:35:29	Thermostat T2	therm
<input type="checkbox"/>	2022-09-06 07:35:29	Thermostat T1	therm
<input type="checkbox"/>	2022-09-06 07:35:29	Raspberry PI Demo Device	defa
<input type="checkbox"/>	2022-09-06 07:35:29	DHT11 Demo Device	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device C1	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device B1	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A3	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A2	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A1	defa

4. Click on the **Trigger AOTA** button again and complete each field per information below:

Activities Firefox Web Browser

ThingsBoard | Dashboard × +

← → ↻ https://192.168.1.100/dashboards/55aa4d10-2db8-11ed-bb88-5d...

ThingsBoard

Dashboards > Intel Fleet Manager v2.0

Fleet > Vehicle Details Vehicle-1

0 0 MPH 180

Details

Vin	Engine speed	Gear position	Fuel efficiency

Temperature

Engine temperature	Cabin temperature	Cargo a

Tire pressure

Axis	Outer Left	Inner Left	Inner Right

Device Manageability ...

App: docker-compose

Command: up

Container Tag: eif

Fetch: Enter the HTTP server that was set up at the start of this section.

Leave the other sections empty and click on the **Send** button.

In the step above, the Worker will access the Server through the local HTTP server to fetch the eif bundle.

In the device telemetry log, you can see that eif bundle was fetched from the local server and was deployed successfully:

Activities Firefox Web Browser

ThingsBoard | Dashboard x ThingsBoard | Devices x +

← → ↻ https://192.168.1.100:8080/devices

ThingsBoard

Devices

Device profile: All x

<input type="checkbox"/>	Created time ↓	Name	Device profile
<input type="checkbox"/>	2022-09-06 07:46:51	Vehicle-1	fleet
<input type="checkbox"/>	2022-09-06 07:35:29	Thermostat T2	therm
<input type="checkbox"/>	2022-09-06 07:35:29	Thermostat T1	therm
<input type="checkbox"/>	2022-09-06 07:35:29	Raspberry PI Demo Device	defa
<input type="checkbox"/>	2022-09-06 07:35:29	DHT11 Demo Device	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device C1	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device B1	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A3	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A2	defa
<input type="checkbox"/>	2022-09-06 07:35:29	Test Device A1	defa

- Home
- Rule chains
- Customers
- Assets
- Devices
- Device profiles
- OTA updates
- Entity Views
- Edge instances
- Edge management
- Widgets Library
- Dashboards
- Audit Logs
- Api Usage
- System Settings

NOTE To stop the application on Worker, trigger the AOTA events again and set **Command** to **down** instead of **up**.

To verify that the EIF Reference Implementation was successfully deployed on the new node, check the list of running containers with the command:

```
docker ps
```

The output will be similar to the following snapshot:

```
Applications [EII Visualizer App] intel@edgesoftware - W...
intel@edgesoftware:provision$ docker ps
CONTAINER ID        IMAGE                                     COMMAND
546b6961081c        10.237.32.181 5002/ia_visualizer_evmsc:2.6.1          "/vis
8a9e321e1bfd        10.237.32.181 5002/ia_image_store_evmsc:2.6.1         "pytho
8eb13b58862a        10.237.32.181 5002/ia_image_store_evmsc2:2.6.1        "pytho
eb4380ff8f66        10.237.32.181 5002/ia_rule_engine:2.6.1              "pytho
aec285a77c94        10.237.32.181 5002/ia_video_ingestion_evmsc2:2.6.1    "/Vid
151fd27c87b7        10.237.32.181 5002/ia_alert_notification_framework:2.6.1 "pytho
457cd6ea62f9        10.237.32.181 5002/ia_cloud_connector:2.6.1          "pytho
fd73588a791b        10.237.32.181 5002/ia_truck_simulator:2.6.1          "pytho
964263a41fec        10.237.32.181 5002/ia_video_ingestion_evmsc:2.6.1     "/Vid
21f48a58ac3b        10.237.32.181 5002/ia_video_ingestion_evmsc4:2.6.1    "/Vid
ba1234257ec1        10.237.32.181 5002/ia_image_store_evmsc3:2.6.1        "pytho
9f71435b213e        10.237.32.181 5002/ia_video_ingestion_evmsc3:2.6.1    "/Vid
87f351ca68e4        10.237.32.181 5002/ia_event_recording_framework:2.6.1 "pytho
60dbb2ce98cf        10.237.32.181 5002/ia_image_store_evmsc4:2.6.1        "pytho
intel@edgesoftware:provision$
```

As the Reference Implementation is launched with test videos, which are available only on the Server and are not deployed on Worker, it will crash on first launch. To fix this, you need to find out what test video needs to be copied from the Server to the Worker and where.

To find out which video to copy, first identify all the Video Ingestors/Analytics Docker* images that were launched for the RI with the command above, and for each of them run the following command:

```
docker logs <container_id>
```

You will see output similar to:

```

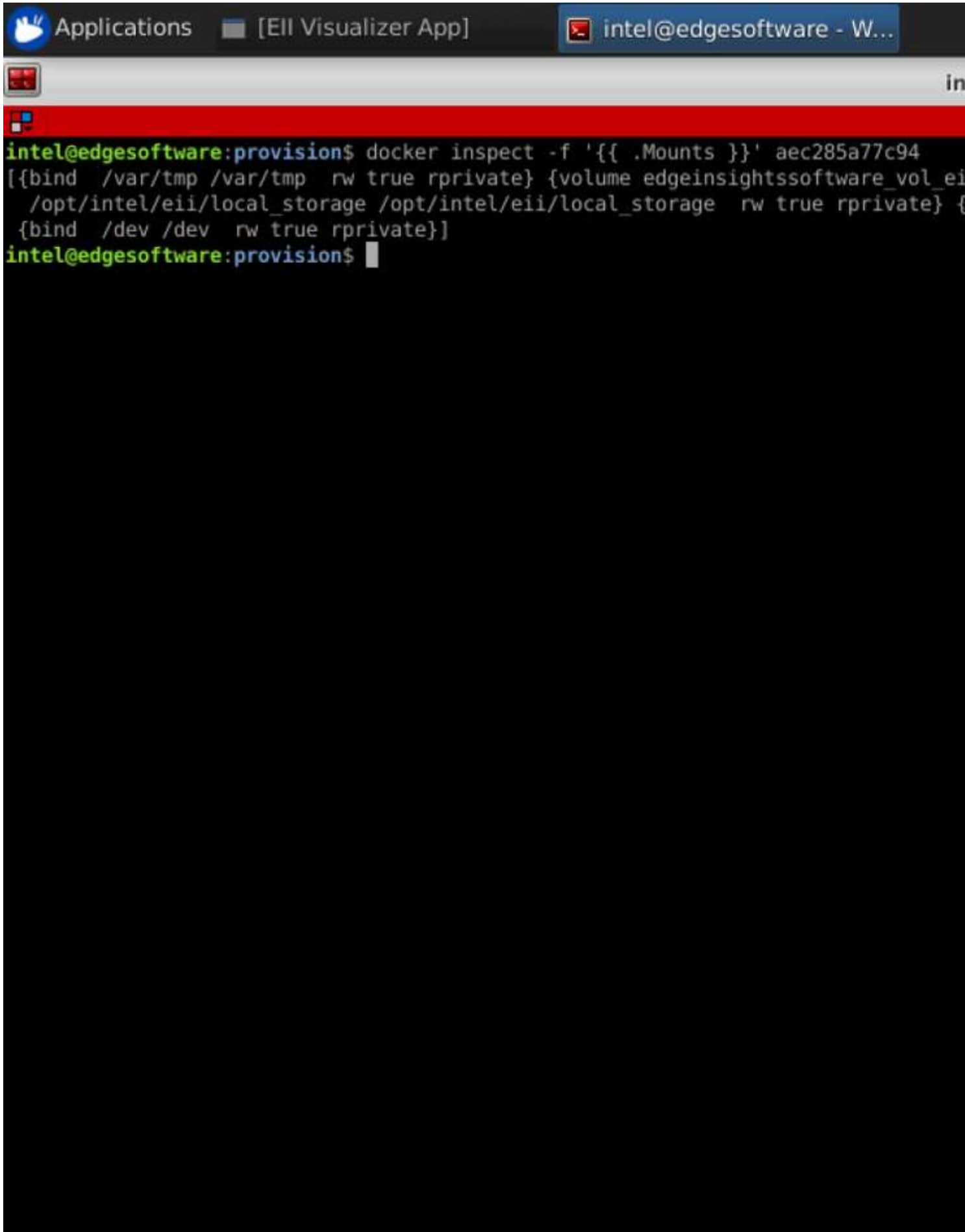
Applications [EII Visualizer App] intel@edgesoftware - W...
2022-02-18 15:31:04,957 : INFO : {} : SEMANTIC_SEGMENTATION : [semantic-segment
2022-02-18 15:31:04,957 : INFO : {} : SEMANTIC_SEGMENTATION : [semantic-segment
xml: ./semantic-segmentation-adas-0001/model/semantic-segmentation-adas-
bin: ./semantic-segmentation-adas-0001/model/semantic-segmentation-adas-
2022-02-18 15:31:05,047 : INFO : {} : SEMANTIC_SEGMENTATION : [semantic-segment
2022-02-18 15:31:05,511 : INFO : {} : COMMON_DATA : [common_data_udf.py] : __ini
[Fri Feb 18 15:31:05 2022] INFO:Ingestor:66: Poll interval: 0.040000
[Fri Feb 18 15:31:05 2022] WARN:get_config_value:204: JSON does not contain key
[Fri Feb 18 15:31:05 2022] INFO:OpenCvIngestor:68: cvt_pipeline initialized
[Fri Feb 18 15:31:05 2022] INFO:OpenCvIngestor:80: Pipeline: ./test_videos/Blir
(gst-plugin-scanner:48): GStreamer-WARNING **: 15:31:05.652: Failed to load plug
ct file: No such file or directory
(gst-plugin-scanner:48): GStreamer-WARNING **: 15:31:05.695: Failed to load plug
file: No such file or directory
(video-ingestion:23): GStreamer-CRITICAL **: 15:31:06.535: gst_element_make_from
[ WARN:0] global ../opencv/modules/videoio/src/cap_gstreamer.cpp (854) open Oper
[ WARN:0] global ../opencv/modules/videoio/src/cap_gstreamer.cpp (597) isPipelir
libva info: VA-API version 1.11.0
libva info: User environment variable requested driver 'iHD'
libva info: Trying to open /usr/lib/x86_64-linux-gnu/dri/iHD_drv_video.so
libva info: Found init function __vaDriverInit_1_11
libva info: va_openDriver() returns 0
MFX: Unsupported extension: ./test_videos/BlindspotFront.mp4
[Fri Feb 18 15:31:06 2022] ERROR:OpenCvIngestor:98: Failed to open gstreamer pip
[Fri Feb 18 15:31:06 2022] WARN:get_config_value:204: JSON does not contain key
[Fri Feb 18 15:31:06 2022] WARN:get_config_value:204: JSON does not contain key
[Fri Feb 18 15:31:06 2022] WARN:get_config_value:204: JSON does not contain key
[Fri Feb 18 15:31:06 2022] WARN:get_config_value:204: JSON does not contain key
[Fri Feb 18 15:31:06 2022] WARN:get_config_value:204: JSON does not contain key
[Fri Feb 18 15:31:06 2022] WARN:zap_initialize:205: Running ZeroMQ TCP sockets
[Fri Feb 18 15:31:06 2022] WARN:get_config_value:204: JSON does not contain key
[Fri Feb 18 15:31:06 2022] WARN:get_config_value:204: JSON does not contain key
[Fri Feb 18 15:31:06 2022] WARN:init_curve_server_socket:1985: ZeroMQ TCP socke
[Fri Feb 18 15:31:06 2022] INFO:start:345: Publisher thread started...
[Fri Feb 18 15:31:06 2022] INFO:start:349: Started udf manager
[Fri Feb 18 15:31:06 2022] INFO:start:358: Ingestor thread started...
[Fri Feb 18 15:31:06 2022] INFO:run:119: Ingestor thread running publishing on
[Fri Feb 18 15:31:06 2022] WARN:read:239: Video ended...
intel@edgesoftware:provision$

```

To identify the location where these videos need to be copied, run the command for each container as above:

```
docker inspect -f '{{ .Mounts }}' <container_id>
```

You will see output similar to:



The image shows a terminal window titled "intel@edgesoftware - W...". The terminal output displays the command `docker inspect -f '{{.Mounts}}' aec285a77c94` and its output, which lists three mounts: `/var/tmp` (bind mount), `/opt/intel/eii/local_storage` (volume mount), and `/dev` (bind mount).

```
intel@edgesoftware:provision$ docker inspect -f '{{.Mounts}}' aec285a77c94
[{"bind": "/var/tmp", "source": "/var/tmp", "target": "/var/tmp", "type": "bind", "rw": true, "rprivate": true}, {"volume": "edgeinsightssoftware_vol_eii", "source": "/opt/intel/eii/local_storage", "target": "/opt/intel/eii/local_storage", "type": "volume", "rw": true, "rprivate": true}, {"bind": "/dev", "source": "/dev", "target": "/dev", "type": "bind", "rw": true, "rprivate": true}]
intel@edgesoftware:provision$
```

The name will be different based on the installed Reference Implementation.

The videos were deployed along with the Reference Implementation on the Server. Search for them in the installation folder and copy them to the folder shown above. Once the videos are in the right place, trigger AOTA with 'down' event and another one with 'up' event.

If the Visualizer does not appear, run the following command:

```
xhost +
```

Verify Triggered AOTA in Event

Once the AOTA event is triggered, you can verify the log of the triggered call. This can be one of the verification tasks done during the development phase.

1. Go to Worker and run the following command to see the logs:

```
journalctl -fu dispatcher & journalctl -fu cloudadapter
```

2. Note the event logs on the ThingsBoard* server show which commands have been run.

If the event log does not appear, follow these steps:

- Change settings from ERROR to DEBUG everywhere in these files:

```
/etc/intel-manageability/public/dispatcher-agent/logging.ini
/etc/intel-manageability/public/cloudadapter-agent/logging.ini
```

- Run the commands:

```
sudo systemctl restart dispatcher
sudo systemctl restart cloudadapter
```

Set Up KnowGo* Simulator and Vehicle Simulator Data

You can use a simulator to prepare the Edge Insights for Fleet and its Reference Implementations to cover a real world scenario. The simulated data offers an overview on how the reference implementations works with cloud features while sending and receiving car data such as: speed, location, tire pressure, fuel level and driving mode.

The package includes two options:

1. KnowGo* Simulator: Offers the possibility to create the injection data manually, while the use case is running.
2. Pre-recorded data using CSV files: Uses the pre-recorded data from the CSV file in a loop.

Configure KnowGo MQTT Connection

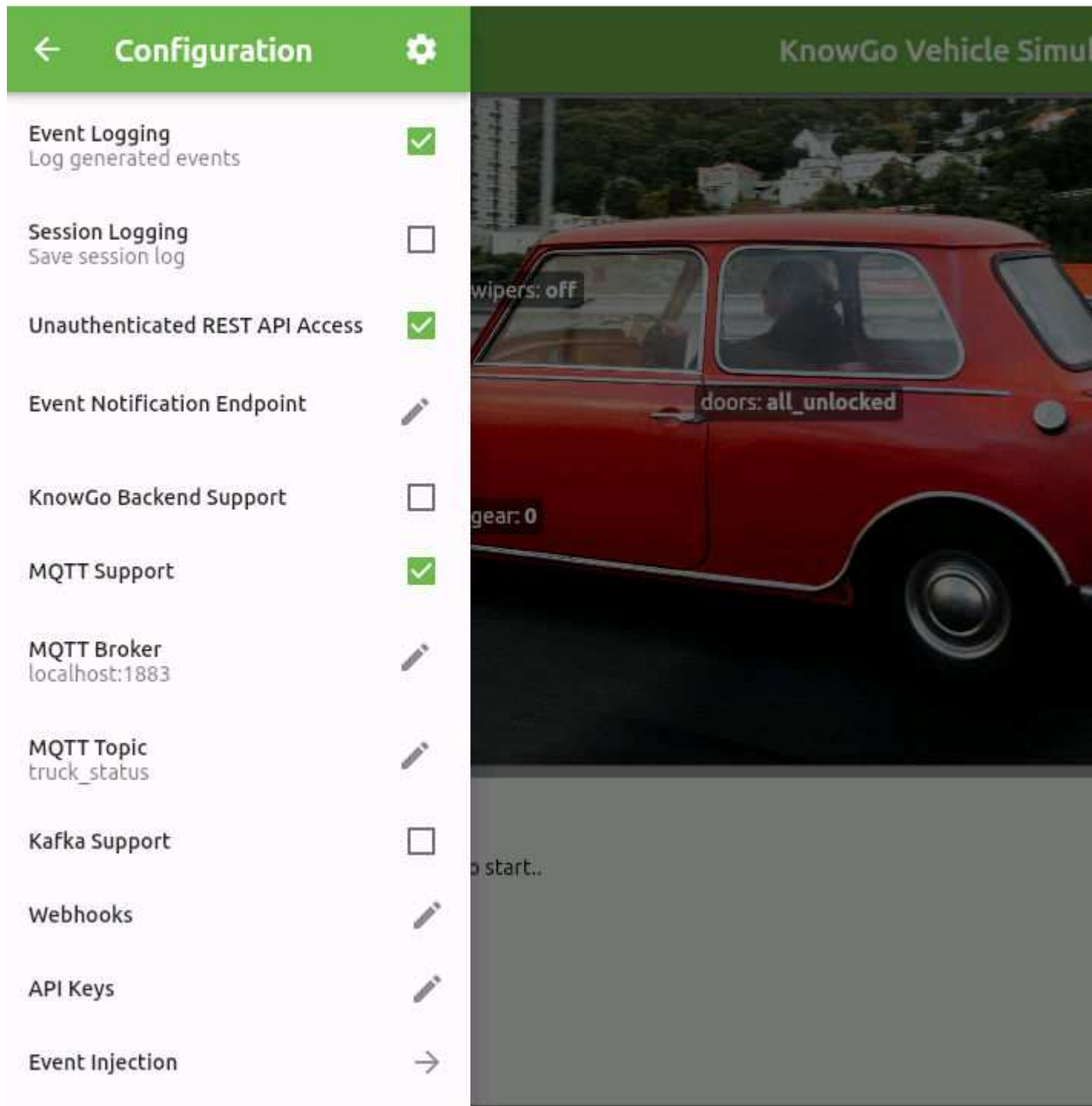
A **Data injection** option was added on the main web page of the Fleet Reference Implementations. If you select the KnowGo option after you configure and start the reference implementation, the KnowGo Simulator will start.

Run the following steps to configure the KnowGo Simulator:

1. Select the Navigation button:



2. Enable MQTT Support.
3. Edit the MQTT Broker by adding **localhost:1883**
4. Edit the MQTT Topic by adding **truck_status**

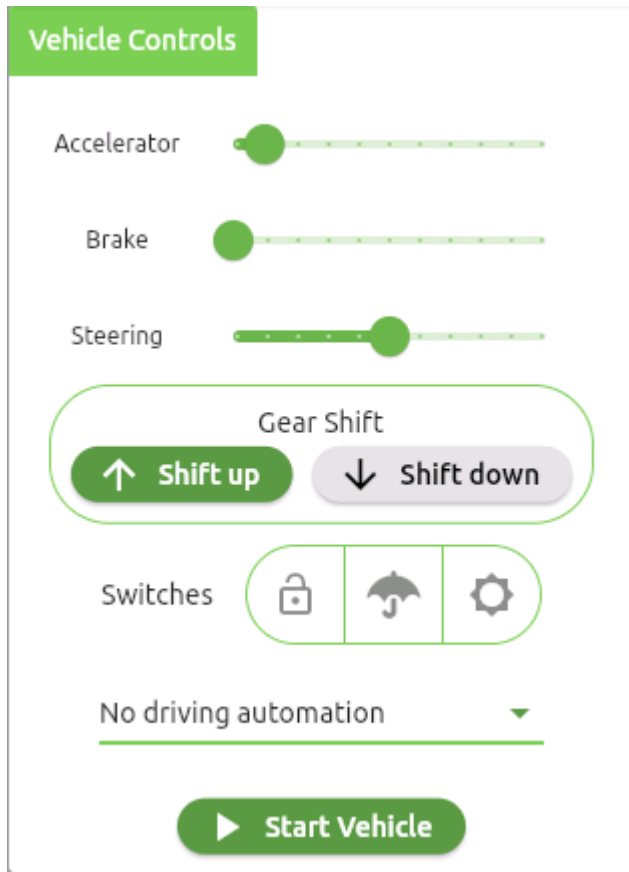


At this point, both the Reference Implementation Visualizer and ThingsBoard* vehicle page should show Surveillance Driving mode, and 0 km/h speed.

NOTE The configuration is not persistent. Each time you open the KnowGo* simulator, you have to reconfigure the MQTT connection.

KnowGo* Panels Overview

KnowGo Vehicle Controls Panel



This panel contains the following end-user options:

1. Three sliders to control the accelerator level, brake level and the steering simulation data.
2. Gear Shift panel that allows you to simulate the gear on which the vehicle is driven.
3. Switches panel that allows simulation of door lock, wipers and headlamps.
4. Drop-down list menu for Driving automation.
5. Start/Stop Vehicle button.

KnowGo Vehicle Information Panel

Based on the options selected on the Vehicle control panel, this panel will show the gear setting, the wipers setting, the door lock setting and the headlamp settings.



KnowGo Console

Logs will be printed containing the injected data.

KnowGo Vehicle Stats Panel

Vehicle Stats

VIN: MNB818PX91A929555
 Odometer: 51825.13 km
 Lat: 48.0202, Lng: 11.5848, Heading: 0°
 Distance Traveled: 0.00 km
 Vehicle Speed: 0.00 km/h
 Engine Speed: 0 RPMs

Fuel Consumed:

Fuel Level:

Based on the selections on the **Vehicle Controls** panel and the application that you run, this panel will print information such as speed, engine RPM, driving distance, GPS location, fuel level and consumption, a simulated Vehicle Identification Number (VIN) and the simulated odometer distance.

Configure KnowGo* Driving Modes

You can enable different driving modes on the KnowGo Simulator that can be used with all of the Edge Insights for Fleet reference implementations.

The following driving modes are recommended:

1. **Surveillance:** Proposed to be used on Address Recognition and Analytics and Work Zone Analytics Reference Implementations.
2. **Cargo:** Proposed to be used on Cargo Management, Address Recognition and Analytics, and Work Zone Analytics Reference Implementations.
3. **Urban Driving:** Proposed to be used on Vehicle Event Recording, Driver Behavior Analytics, Public Transport Analytics, Automated License Plate Recognition, and Road Sign Detection and Classification Reference Implementations.
4. **Highway:** Proposed to be used on Vehicle Event Recording, Driver Behavior Analytics, Public Transport Analytics, Automated License Plate Recognition, and Road Sign Detection and Classification Reference Implementations.
5. **Parking:** Proposed for any Reference Implementation, but not for Cargo Management.

Check the instructions below for each driving mode:

Surveillance Mode

- Reference Implementations:
 - Address Recognition and Analytics
 - Work Zone Analytics
- Settings:
 - Vehicle Stopped
 - Headlamps stopped
 - Accelerator - 0
 - Gear - 0
 - Speed - 0

Know

Know

wipers: off

headlamp: off

gear: 0

doors: all_unlock

Console

[2022-09-01 11:57:01] Turning headlamp off

[2022-09-01 11:56:51] Stopping vehicle

[2022-09-01 11:56:50] Event[eventID=249, autoID=311, driverID=908, automationLevel=0, engineSpeed=0.00, vehicleSpeed=0.00, acceleratorPedalPosition=0.0, brakePedalPosition=0.0, odometer=17554.69, ignitionStatus=run, fuelLevel=99.99, fuelConsumedSinceRestart=0.0, windshieldWiperStatus=false, latitude=48.020250, longitude=11.584850, bearing=0.0, heading=0.0]

[2022-09-01 11:56:49] Event[eventID=248, autoID=311, driverID=908, automationLevel=0, engineSpeed=0.00, vehicleSpeed=0.00, acceleratorPedalPosition=0.0, brakePedalPosition=0.0, odometer=17554.69, ignitionStatus=run, fuelLevel=99.99, fuelConsumedSinceRestart=0.0, windshieldWiperStatus=false, latitude=48.020250, longitude=11.584850, bearing=0.0, heading=0.0]

Cargo Mode

- Reference Implementations:
 - Cargo Management
 - Address Recognition and Analytics
 - Work Zone Analytics

- Settings:
 - Vehicle started
 - Accelerator - 0
 - Gear - 0
 - Headlamps - on
 - Door - unlocked

☰



Console

[2022-09-01 11:55:42] Event[eventID=181, autoID=311, driverID=908, automationLevel=1, engineSpeed=0.00, vehicleSpeed=0.00, acceleratorPedalPosition=0.0, brakePedalPosition=0.0, odometer=17554.69, ignitionStatus=run, fuelLevel=99.99, fuelConsumedSinceRestart=0.0, windshieldWiperStatus=false, latitude=48.020250, longitude=11.584850, bearing=0.0, heading=0.0]

[2022-09-01 11:55:41] Event[eventID=180, autoID=311, driverID=908, automationLevel=1, engineSpeed=0.00, vehicleSpeed=0.00, acceleratorPedalPosition=0.0, brakePedalPosition=0.0, odometer=17554.69, ignitionStatus=run, fuelLevel=99.99, fuelConsumedSinceRestart=0.0, windshieldWiperStatus=false, latitude=48.020250, longitude=11.584850, bearing=0.0, heading=0.0]

[2022-09-01 11:55:40] Event[eventID=179, autoID=311, driverID=908, automationLevel=1, engineSpeed=0.00, vehicleSpeed=0.00, acceleratorPedalPosition=0.0, brakePedalPosition=0.0, odometer=17554.69, ignitionStatus=run, fuelLevel=99.99, fuelConsumedSinceRestart=0.0, windshieldWiperStatus=false, latitude=48.020250, longitude=11.584850, bearing=0.0, heading=0.0]

Urban Driving Mode

- Reference Implementations:
 - Vehicle Event Recording
 - Driver Behavior Analytics
 - Public Transport Analytics
 - Automated License Plate Recognition
 - Road Sign Detection and Classification
- Settings:
 - Vehicle started
 - Accelerator $\leq 10\%$
 - Gear - 1-3
 - Headlamps - on
 - Door - locked/unlocked
 - Speed to be between 5 and 60 km/h on **Vehicle Stats** panel



Highway Mode

- Reference Implementations:
 - Vehicle Event Recording
 - Driver Behavior Analytics
 - Public Transport Analytics

- Automated License Plate Recognition
 - Road Sign Detection and Classification
- Settings:
 - Vehicle started
 - Accelerator > 10%
 - Gear - 4-6
 - Headlamps - on
 - Door - locked/unlocked
 - Speed - >= 60 km/h



Parking Mode

- Reference Implementations:
- Any Reference Implementation, but **not** Cargo Management

- Settings:
 - Vehicle started
 - From Highway mode or Urban Driving mode start braking until the speed is around 5km/h and the simulation will switch to parking.
 - Accelerator $\leq 10\%$
 - Gear - ≤ 1
 - Speed to be lower than 5 km/h on **Vehicle Stats** panel
 - Headlamps - on
 - Door - locked



The screenshot displays a simulation interface with a red car in the center. The car is labeled with several status indicators: "wipers: off", "headlamp: on", "gear: 1", and "doors: all_locked". The background shows a city street with buildings and a road. The interface includes a green header bar with a menu icon and a "Know" button. Below the car view is a "Console" panel showing a log of events.

Console

```
[2022-09-01 12:09:40] Event[eventID=958, autoID=311, driverID=908, automationLevel=2, engineSpeed=1208.65, vehicleSpeed=6.89, acceleratorPedalPosition=0.0, brakePedalPosition=0.0, odometer=17559.40, ignitionStatus=run, fuelLevel=99.79, fuelConsumedSinceRestart=0.0, windshieldWiperStatus=false, latitude=48.062658, longitude=11.584850, bearing=0.0]
[2022-09-01 12:09:39] Event[eventID=957, autoID=311, driverID=908, automationLevel=2, engineSpeed=1293.43, vehicleSpeed=7.38, acceleratorPedalPosition=0.0, brakePedalPosition=0.0, odometer=17559.40, ignitionStatus=run, fuelLevel=99.79, fuelConsumedSinceRestart=0.0, windshieldWiperStatus=false, latitude=48.062641, longitude=11.584850, bearing=0.0]
[2022-09-01 12:09:38] Event[eventID=956, autoID=311, driverID=908, automationLevel=2, engineSpeed=1384.17, vehicleSpeed=7.90, acceleratorPedalPosition=0.0, brakePedalPosition=0.0, odometer=17559.40, ignitionStatus=run, fuelLevel=99.79, fuelConsumedSinceRestart=0.0, windshieldWiperStatus=false, latitude=48.062641, longitude=11.584850, bearing=0.0]
```

Export KnowGo* Simulation Data

You can export the simulation data by clicking on the following button:



After you click on the button, the path will be printed on a pop-up.

Next, copy the file at the following location to use it as CSV option:

```
<INSTALL_PATH>/<RI_FOLDER>/<RI_MODULE_NAME>_<VERSION>/<RI_MODULE_NAME>/  
<USE_CASE_FOLDER>/src/scripts/truck_simulator_routes/
```

For example:

```
sudo cp <exported_file_path> /home/intel/driver_behavior_analytics/  
Driver_Behavior_Analytics_2022.2/Driver_Behavior_Analytics/EII-DriverBehavior-UseCase/src/  
scripts/truck_simulator_routes/  
cd /home/intel/driver_behavior_analytics/Driver_Behavior_Analytics_2022.2/  
Driver_Behavior_Analytics/EII-DriverBehavior-UseCase/src/scripts/truck_simulator_routes/  
sudo chown $USER:$USER <exported_file>  
chmod 0775 <exported_file>
```

Use Case Manager

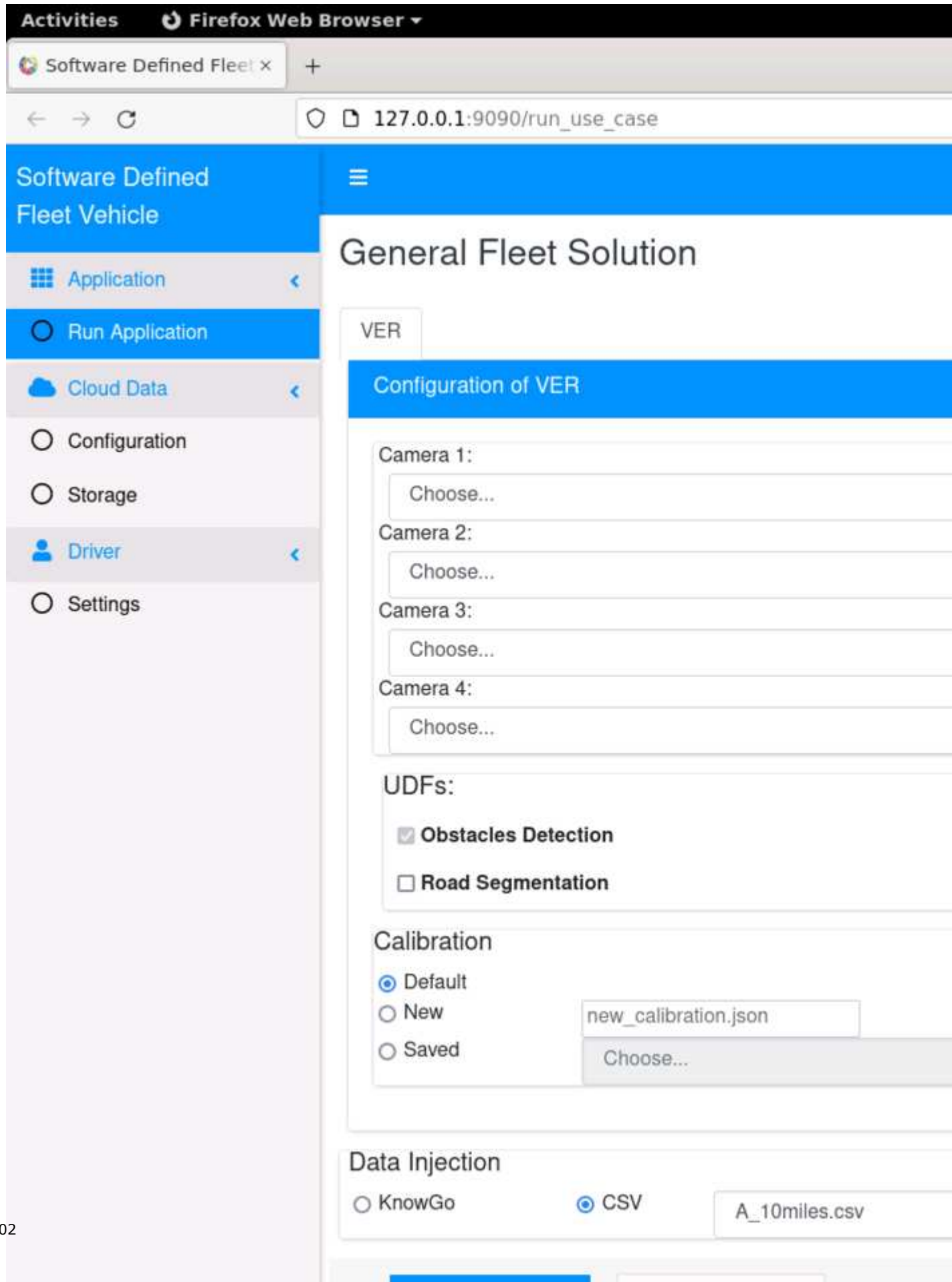
The Use Case Manager allows simplified parallel execution for Edge Insights for Fleet Reference Implementations. This module uses a json file to build and run reference implementations, which are then added to a specific group.

Web Page Structure

The Fleet Manager is able to configure every reference implementation from a single web page: 127.0.0.1:9090. During installation, all reference implementations are added to the default group named "General-Fleet-Solution". After the reference implementation installation is completed, you can navigate to 127.0.0.1:9090 web page and have a sub-menu on the **Run Application** main page with each reference implementation's abbreviated name. (For example, Vehicle Event Recording will appear as VER and Driver Behavior Analytics will appear as DBA.)

The unified web page offers access to Cloud Configuration, Cloud Storage and Driver Settings and those settings will apply for all Reference Implementations that will run.

1. Single Reference Implementation:



2. Multiple Reference Implementations

Activities Firefox Web Browser

Software Defined Fleet x +

127.0.0.1:9090/run_use_case

Software Defined Fleet Vehicle

- Application
- Run Application**
- Cloud Data
- Configuration
- Storage
- Driver**
- Settings

General Fleet Solution

VER DBA

Configuration of VER

Camera 1:
BlindspotFront.mp4

Camera 2:
BlindspotLeft.mp4

Camera 3:
BlindspotRear.mp4

Camera 4:
BlindspotRight.mp4

UDFs:

- ☒ Obstacles Detection
- ☐ Road Segmentation

Calibration

- ☒ Default
- ☐ New
- ☐ Saved

Data Injection

- ☐ KnowGo
- ☒ CSV

NOTE You must configure all the Reference Implementations before you can run the application.

Local Configuration Structure

You can change the configuration of the reference implementations you want to run. If your system doesn't support more than two reference implementations in parallel, you can update the file located at: `/opt/intel/eif/EII-UseCaseManager/scripts/applications.json`

The changes can be made based on `/opt/intel/eif/EII-UseCaseManager/scripts/applications_json_example.txt` that includes all reference implementations recommended for different type of solutions:

```
{
  "General-Fleet-Solution": [
    "EII-EVMSC-UseCase",
    "EII-DriverBehavior-UseCase",
  ],
  "eCommerce-Delivery": [
    "EII-EVMSC-UseCase",
    "EII-DriverBehavior-UseCase",
    "EII-CargoManagement-UseCase",
    "EII-AddressDetection-UseCase"
  ],
  "Public-Transit": [
    "EII-EVMSC-UseCase",
    "EII-DriverBehavior-UseCase",
    "EII-PassengerCounting-UseCase"
  ],
  "Public-Safety": [
    "EII-EVMSC-UseCase",
    "EII-DriverBehavior-UseCase",
    "EII-LicensePlateRecognition-UseCase"
  ],
  "Construction-Work-Zone-Machinery": [
    "EII-EVMSC-UseCase",
    "EII-DriverBehavior-UseCase",
    "EII-WorkZoneDetection-UseCase"
  ]
}
```

Build the defined structure using the following command:

```
cd /opt/intel/eif/EII-UseCaseManager/
make config app="Your_Application_Structure_Name" v=4 pc=true

# Example: make config app=Construction-Work-Zone-Machinery v=4 pc=true.
# Default value is "General Fleet Solution".
```

Launch the newly built application structure using the following command:

```
make webui
```

Launch the application web page. On your preferred browser go to <http://127.0.0.1:9090>

NOTE After you build each new configuration, you need to prepare the cloud configuration as well.

Release Notes

New in this Release

- New Use Case Manager to simplify the parallel execution process.
- Integrated Kuiper* MQTT Server. Kuiper* is an edge lightweight IoT data analytics / streaming software implemented by Golang*, which can be run on all kinds of resource constrained edge devices.
- Kuiper* Service replaced the Rule Engine in the middleware configuration in the architecture diagram in [How it Works](#).
- All reference implementations received the new Edge Insights for Fleet package update.

Edgesoftware CLI Features

Initial Features for Recommended Configuration

- Installs Docker CE*.
- Installs Docker Compose*.
- Installs all the prerequisites and dependencies for the Edge Insights for Fleet in the `/opt/intel/eif` directory.
- Installs and sets up Edge Insights for Fleet and Use Case Manager in the `/opt/intel/eif/` directory.
- Brings up all container images as per the use case selected during Edge Insights for Fleet installation.
- Supports the following reference implementations:
 - Vehicle Event Recording
 - Driver Behavior Analytics
 - Cargo Management
 - Public Transit Analytics
 - Automated License Plate Recognition
 - Address Recognition and Analytics
 - Work Zone Analytics
 - Road Sign Detection and Classification

Edge Insights for Fleet Features:

- Video Analytics
- Video Ingestion
- Training and Learning Suite (TLS)
- Image Store