

# **Edge Insights for Industrial Get Started**

# Contents

<b>Chapter 1: Edge Insights for Industrial</b>	
How it Works .....	3
Device Manageability .....	6
Get Started Guide .....	45
Requirements .....	45
Install Edge Insights for Industrial .....	45
Introduction to the Edge Software CLI .....	59
Tutorials.....	66
Defect Detection Demo .....	66
Release Notes .....	73
Documentation Archive .....	74

---

# Edge Insights for Industrial

---



Edge Insights for Industrial from Intel is a set of pre-validated ingredients for integrating video and time-series data analytics on edge compute nodes.

Edge Insights for Industrial helps to address various industrial and manufacturing usages, which include data collection, storage, and analytics on a variety of hardware nodes across the factory floor. 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 [Tutorials](#) section for step-by-step, hands-on walkthroughs of how to use and configure modules in Edge Insights for Industrial.

## How it Works

---

Edge Insights for Industrial 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.

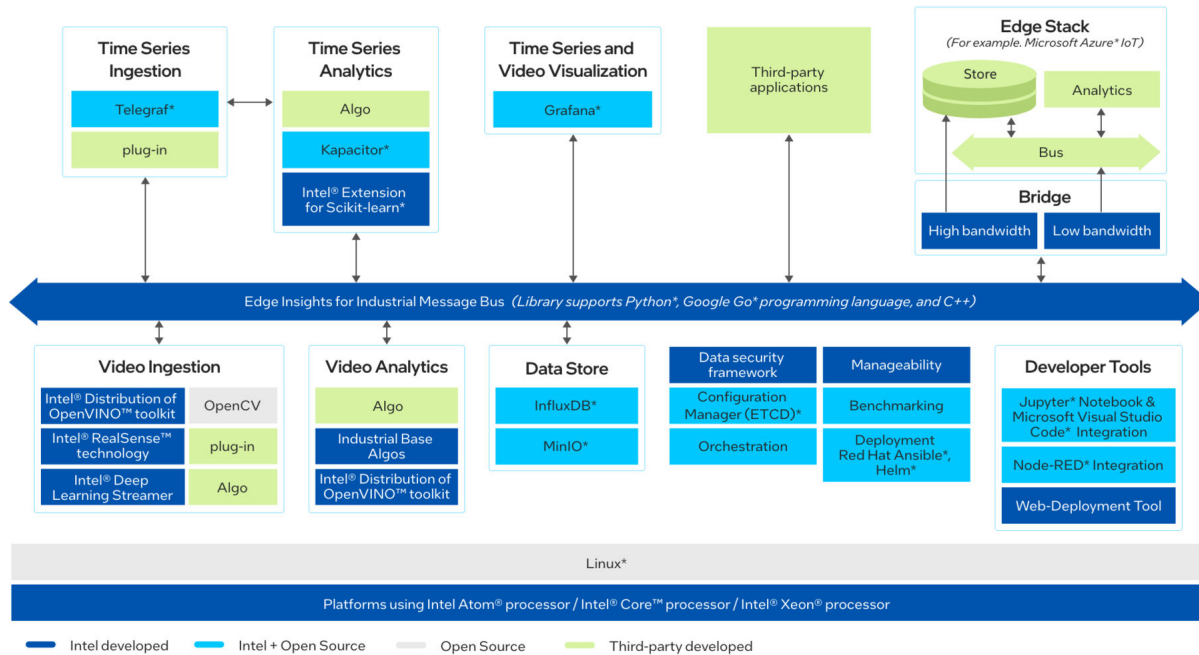
Edge Insights for Industrial is based on [Open Edge Insights for Industrial](#) and offers additional components like reference implementation for various use cases that are accessible via custom download.

---

**NOTE** The Open Edge Insights for Industrial user guide details the capabilities of the Open Edge Insights for Industrial modules and serves as a reference for developers. For more information, see [Open Edge Insights for Industrial User Guide](#).

---

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



## Modules and Services

For an overview of the EII modules, see below.

### Video Ingestion and Analytics

#### Video Ingestion

Edge Insights for Industrial supports ingestion from:

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

Edge Insights for Industrial 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 for Industrial Data Bus. You can create multiple instances of the Video Ingestion container to ingest video from multiple input sources.

Watch a [video](#) on Video Ingestion on the Edge Insights for Industrial [Architecture & Technical Training](#) page.

#### Video Analytics

Classifier algorithms are executed in the Video Analytics container. The container uses the Edge Insights for Industrial Data Bus to subscribe to the input stream published by the Video Ingestion container. 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.

Watch a [video](#) on Video Analytics on the Edge Insights for Industrial [Architecture & Technical Training](#) page.

## Edge Video Analytics Microservice

The Edge Video Analytics Microservice (EVAM) wraps the Intel Video Analytics Serving library to provide an EII visual ingestion and analytics module utilizing the features and functionality of the library. For more details, refer to the [Edge Video Analytics Microservice](#) documentation.

## Web Deployment Tool

Web Deployment Tool (WDT) provides a graphical interface to set up and complete the deployment process for the Open Edge Insights for Industrial services. For more details, refer to the [Web Deployment Tool](#) documentation.

## Image Store

The image store provides storage and retrieval of images as binary blobs. You can store the images persistently by using: In persistent storage with [MinIO\\*](#), an open source, distributed, object storage server.

Watch a [video](#) on Image Store on the Edge Insights for Industrial [Architecture & Technical Training page](#).

## Time Series Analytics

Edge Insights for Industrial leverages [Telegraf\\*](#) for time-series data ingestion into [InfluxDB\\*](#). [Kapacitor\\*](#) reads data from [InfluxDB\\*](#), generates alerts, and takes the appropriate action.

- [Telegraf\\*](#) is an agent for collecting, processing, aggregating and writing performance metrics.
- [InfluxDB\\*](#) is a distributed datastore for metrics, events, and real-time analytics.
- [Kapacitor\\*](#) is a data processing engine that handles both stream and batch data from [InfluxDB](#), acting on this data in real time via its programming language, TICKscript.

Watch a [video](#) on Time Series Data on the Edge Insights for Industrial [Architecture & Technical Training page](#).

## Security and Configuration

The Data Security Framework enables security through a two-stage process involving provisioning and execution on 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 Industrial 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.

## Manageability

Edge Insights for Industrial uses [Azure\\*](#) and [ThingsBoard\\*](#) to support over-the-air updates and to enable cloud-to-edge manageability of application services.

For details, see [Manageability section](#).

## 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 for Industrial Data Bus

Edge Insights for Industrial 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.

## Device Manageability

Device Manageability enables software updates and deployment from cloud to device. This includes the following:

- Software over the air (SOTA)
- Firmware update over the air (FOTA)
- Application over the air (AOTA) and system operations

The **AOTA** update enables cloud to edge manageability of application services running on the Edge Insights for Industrial (EII) enabled systems through Device Manageability.

For EII use case, only **AOTA** features from Device Manageability are validated and is supported through **Azure\*** and **ThingsBoard\*** cloud-based management front-end services. You can select either one of these options depending on preference.

The following section will walk you through the following:

- Setting up **Azure\*** and **ThingsBoard\***
- Establishing connectivity with the target systems
- Updating applications on systems

You can refer to the user guide in **docs** for more detailed reference.

---

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

---

## Install Device Manageability

### Prerequisites

The following are the prerequisites to install Device Manageability:

1. Ubuntu 18.04 and Ubuntu 20.04.
2. Install the latest docker cli/docker daemon. Refer to the following sections:
  - **Install using the repository** and **Install Docker Engine** at <https://docs.docker.com/install/linux/docker-ce/ubuntu/#install-docker-ce>.
  - Also, follow the Manage Docker as a non-root user section at <https://docs.docker.com/install/linux/linux-postinstall/> to run docker without sudo.
3. Follow the below steps only if the node or system on which the docker setup is tried out is running behind a HTTP proxy server otherwise, you can skip this step.
  - a. Configure proxy settings for docker client to connect to internet and for containers to access internet. For more information, see <https://docs.docker.com/network/proxy/>.
  - b. Configure proxy settings for docker daemon. For more information, see <https://docs.docker.com/config/daemon/systemd/#httphttps-proxy>.
4. Install docker-compose tool. For more information, see <https://docs.docker.com/compose/install/#install-compose>

All Device Manageability devices can be controlled using a cloud service application. To install Device Manageability, complete the following:

---

**NOTE** This step is required only if you select cloud service **Azure\*** and **ThingsBoard\***.

---

1. Install Device Manageability agent on the machine.

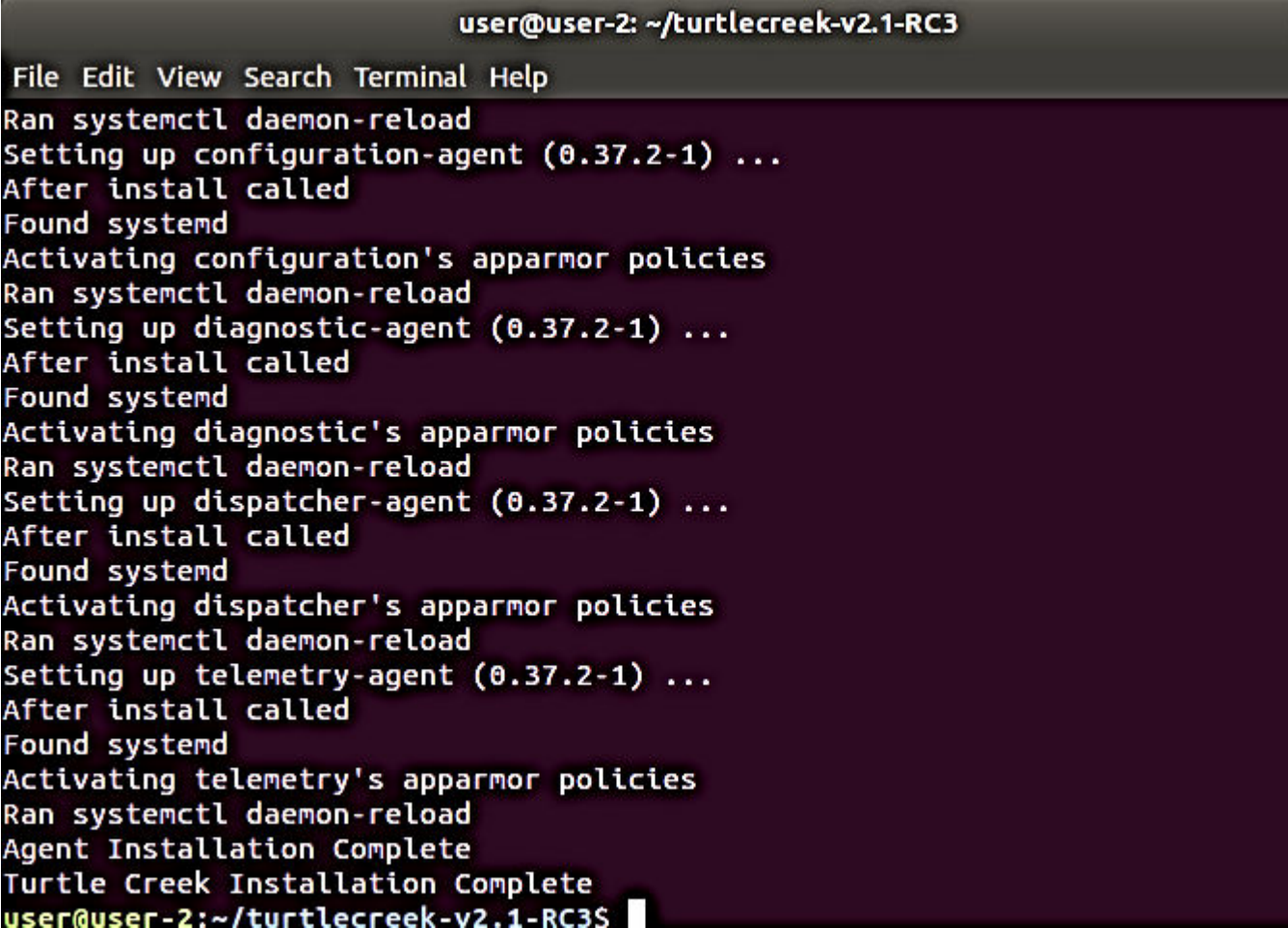
- a. Run below steps for installation from manageability folder:

```
sudo chmod +755 install-tc.sh
sudo ./install-tc.sh
```

**NOTE** Before you install a newer version of Device Manageability, check if the system has old version of Device Manageability. If an older version is installed then, uninstall the older version with the same version of the uninstall script.

- b. After reading all the licenses, press `q` to finish.
- c. Enter `Y` to accept the license.

The following figure shows the terminal output you should expect to see after the installation is complete.

A terminal window screenshot with a dark background and light text. The title bar reads 'user@user-2: ~/turtlecreek-v2.1-RC3'. The terminal output shows a series of commands and their outputs: 'Ran systemctl daemon-reload', 'Setting up configuration-agent (0.37.2-1) ...', 'After install called', 'Found systemd', 'Activating configuration's apparmor policies', 'Ran systemctl daemon-reload', 'Setting up diagnostic-agent (0.37.2-1) ...', 'After install called', 'Found systemd', 'Activating diagnostic's apparmor policies', 'Ran systemctl daemon-reload', 'Setting up dispatcher-agent (0.37.2-1) ...', 'After install called', 'Found systemd', 'Activating dispatcher's apparmor policies', 'Ran systemctl daemon-reload', 'Setting up telemetry-agent (0.37.2-1) ...', 'After install called', 'Found systemd', 'Activating telemetry's apparmor policies', 'Ran systemctl daemon-reload', 'Agent Installation Complete', and 'Turtle Creek Installation Complete'. The prompt 'user@user-2:~/turtlecreek-v2.1-RC3\$' is visible at the bottom.

```
user@user-2: ~/turtlecreek-v2.1-RC3
File Edit View Search Terminal Help
Ran systemctl daemon-reload
Setting up configuration-agent (0.37.2-1) ...
After install called
Found systemd
Activating configuration's apparmor policies
Ran systemctl daemon-reload
Setting up diagnostic-agent (0.37.2-1) ...
After install called
Found systemd
Activating diagnostic's apparmor policies
Ran systemctl daemon-reload
Setting up dispatcher-agent (0.37.2-1) ...
After install called
Found systemd
Activating dispatcher's apparmor policies
Ran systemctl daemon-reload
Setting up telemetry-agent (0.37.2-1) ...
After install called
Found systemd
Activating telemetry's apparmor policies
Ran systemctl daemon-reload
Agent Installation Complete
Turtle Creek Installation Complete
user@user-2:~/turtlecreek-v2.1-RC3$
```

- d. After the Device Manageability has been installed successfully, set the following values in the `/etc/intel_manageability.conf` file.

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

- e. Under the `<all>` `</all>` section, change `db`s ON to OFF. Refer to the following figures. DBS stands for Docker Bench Security, this feature of Device Manageability is not used for EII.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- configuration-agent/fpm-template/etc/intel_manageability
<configurations>
  <all>
    <db>ON</db>
  </all>
  <telemetry>
    <collectionIntervalSeconds>60</collectionIntervalSeconds>
    <publishIntervalSeconds>300</publishIntervalSeconds>
    <maxCacheSize>100</maxCacheSize>
    <containerHealthIntervalSeconds>600</containerHealthIntervalSeconds>
  </telemetry>
```

```
<?xml version="1.0" encoding="utf-8"?>
<!-- configuration-agent/fpm-template/etc/intel_manageability
<configurations>
  <all>
    <db>OFF</db>
  </all>
  <telemetry>
    <collectionIntervalSeconds>60</collectionIntervalSeconds>
    <publishIntervalSeconds>300</publishIntervalSeconds>
    <maxCacheSize>100</maxCacheSize>
    <containerHealthIntervalSeconds>600</containerHealthIntervalSeconds>
  </telemetry>
```

- f. Add the URL endpoint for the developer files to the TrustedRepositories as shown in the following figure

```
<trustedRepositories>
https://api-dev.devicewise.com
</trustedRepositories>
```

- g. Save and exit. Restart the machine to apply the changes.

### EII Provisioning and Deployment via Bundle Generation

EII deployment via bundle generation requires the use of the Docker\* registry. The following section outlines some of the commands to create EII bundle

---

**NOTE** This step is required only if you select cloud service **Azure Portal\*** and **ThingsBoard\***.

---

1. Set up the Docker registry URL, build, and push images.



- a. Update docker registry URL in **DOCKER\_REGISTRY** variable **<ip:port>**. The example uses **10.221.40.65:5000/** as the Docker registry.

```
sudo vi [WORK_DIR]/IEdgeInsights/build/.env
```

```
user@user-1: ~/EIS2.1/IEdgeInsights/docker_setup
File Edit View Search Terminal Help
# video analytics
SOCKET_DIR=/EIS/sockets

# DEV_MODE if set `true` allows one to run EIS in non-secure mode and provide
additional UX/DX etc.,
DEV_MODE=true
# PROFILING_MODE is set 'true' allows to generate profile/performance data
PROFILING_MODE=false

RTSP_CAMERA_IP=localhost
# All server certificates will be generated with below HOST_IP
# If HOST_IP is blank, HOST_IP will be automatically detected while generating
certificate
HOST_IP=
# Please provide docker registry details below for docker-compose push/pull
# Please provide full registry url with port trail by /
# e.g. localhost:5000/
DOCKER_REGISTRY=10.221.40.65:5000/

# EIS dependency image versions
UBUNTU_IMAGE_VERSION=18.04
GRAFANA_VERSION=6.4.3
INFLUXDB_VERSION=1.5.3
EIS_VERSION=2.1
```

- b. In the same build repository, build the image using the following command:

```
docker-compose -f docker-compose-build.yml build
```

If the build is successful, you should see the following terminal output:

```

user@user-1: ~/EIS2.1/IEdgeInsights/docker_setup
File Edit View Search Terminal Help
---> Using cache
---> 98592ed97bac
Step 27/31 : COPY --from=common ${GO_WORK_DIR}/common/cmake ${PY_WORK_DIR}/c
/cmake
---> Using cache
---> b520ec205a87
Step 28/31 : COPY --from=common /usr/local/lib /usr/local/lib
---> Using cache
---> c0b687d8bbb6
Step 29/31 : COPY --from=common /usr/local/lib/python3.6/dist-packages/ /usr
l/lib/python3.6/dist-packages
---> Using cache
---> c27491060d45
Step 30/31 : COPY . .
---> Using cache
---> 85d580eb49cd
Step 31/31 : ENTRYPOINT ["python3.6", "visualize.py", "-l"]
---> Using cache
---> 1f34e088421a

[Warning] One or more build-args [EIS_UID] were not consumed
Successfully built 1f34e088421a
Successfully tagged 10.221.40.65:5000/ia_visualizer:2.1
user@user-1: ~/EIS2.1/IEdgeInsights/docker_setup$

```

- c. Push the image to the registry:

```
docker-compose -f docker-compose-push.yml push
```

## Generating EII Bundle for Deployment

The following steps are used to generate the EII\_bundle. This software will only work on Python\* 3 and onwards.

1. Update config.json file to include the services that you need for the bundle. Ensure to exclude those services that you do not want with the bundle.

```
$ sudo vi build/deploy/config.json
```

```

{
  "docker_compose_file_version": "<docker_compose_file_version which is compose file version
supported by deploying docker engine>",
  "exclude_services": [list of services which you want to exclude from your deployment]
  "include_services": [list of services needed in final deployment docker-compose.yml]
}

```

**Note:** Ensure that you have updated the DOCKER\_REGISTRY in build/.env as mentioned in the **EII deployment** section.

2. Run the following script in the terminal to generate the eii\_bundle.

```
cd [WORK_DIR]/IEdgeInsights/build/deploy/
sudo python3 generate_eii_bundle.py -gb
```

The default value for bundle name is **eii\_bundle.tar.gz**, while the tag name is **eii\_bundle**.

Using `-t` options can give you a custom tag name which will also be used for bundle generation.

The default value for the docker-compose yml path is `../docker-compose.yml`.

3. The file **eii\_bundle.tar.gz** will be created in the same folder. Create the following two bundles for the deployment:
  - Bundle 1: Generate bundle named `cfgmgr_bundle.tar.gz` with only "ia\_configmgr\_agent" as included service.
  - Bundle 2: Generate bundle named `eii_bundle.tar.gz` with all the required included EII services.

## Creating an HTTP Local Server

**NOTE** This step is required only for user that select cloud service Azure\* or ThingsBoard\* cloud services and for development purpose. During actual implementation, users need to have their own cloud http server to upload the EII bundle that being generated and use the URL to fetch the uploaded bundle on the AOTA step later.

The generated EII bundle need to available via http server to perform AOTA fetching from the cloud services.

### Prerequisites

Run `sudo vi /etc/environment`.

Under `no_proxy` variable, add the local http server (example, 10.221.40.65).

1. Run the following command on the ThingsBoard machine to create the HTTP server for sharing the `eii_bundle`:

```
python3 -m http.server [any port number]
```

2. Add the fileserver endpoint to the configuration file on the node that you will access via the local HTTP server.

In the example, the steps below must be done on the new node. The new node will access the local HTTP server to fetch the `eii_bundle` that is being created.

- a. Open the configuration file.

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

- b. Under **trustedRepositories**, add the bundle hosting server endpoint on a new line. <http://xx.xxx.xxx.xx> as used above.

```
<trustedRepositories>
  https://af01p-igk.devtools.intel.com/artifactory/SID-Docker-local/bmp
  https://ubit-artifactory-or.intel.com/artifactory/iotg-bmp-internal-1
  https://af01p-igk.devtools.intel.com/artifactory/iotg-bmp-test-local/
  https://af01p-igk.devtools.intel.com/artifactory/iotg-bmp-igk-local/
  http://ci_nginx:80
  https://api-dev.devicewise:433/file/
  http://10.221.40.65
</trustedRepositories>
```

**NOTE** The above example is running local HTTP server at 10.221.40.65:8000

## Cloud Service: Azure Server Setup

You can connect and control the Device Manageability devices from the Azure\* portal. Create an Azure\* account from the following link: <https://azure.microsoft.com/en-us/free/>

To create an account, select start free, and create the account.

### Setting up an Azure IoT Central Application

1. To generate a premade Device Manageability (Intel Manageability) IoT Central application get the link from `/usr/share/cloudadapter-agent/azure_template_link.txt` file on Device Manageability machine.
2. Copy and paste the link in the browser, you will be redirected to the Azure\* UI as shown below.



Build &gt; New application



# Intel Manageability v1.3



Use this template to interact with devices running Intel Manageability.



## About your app

Application name \* ⓘ

URL \* ⓘ

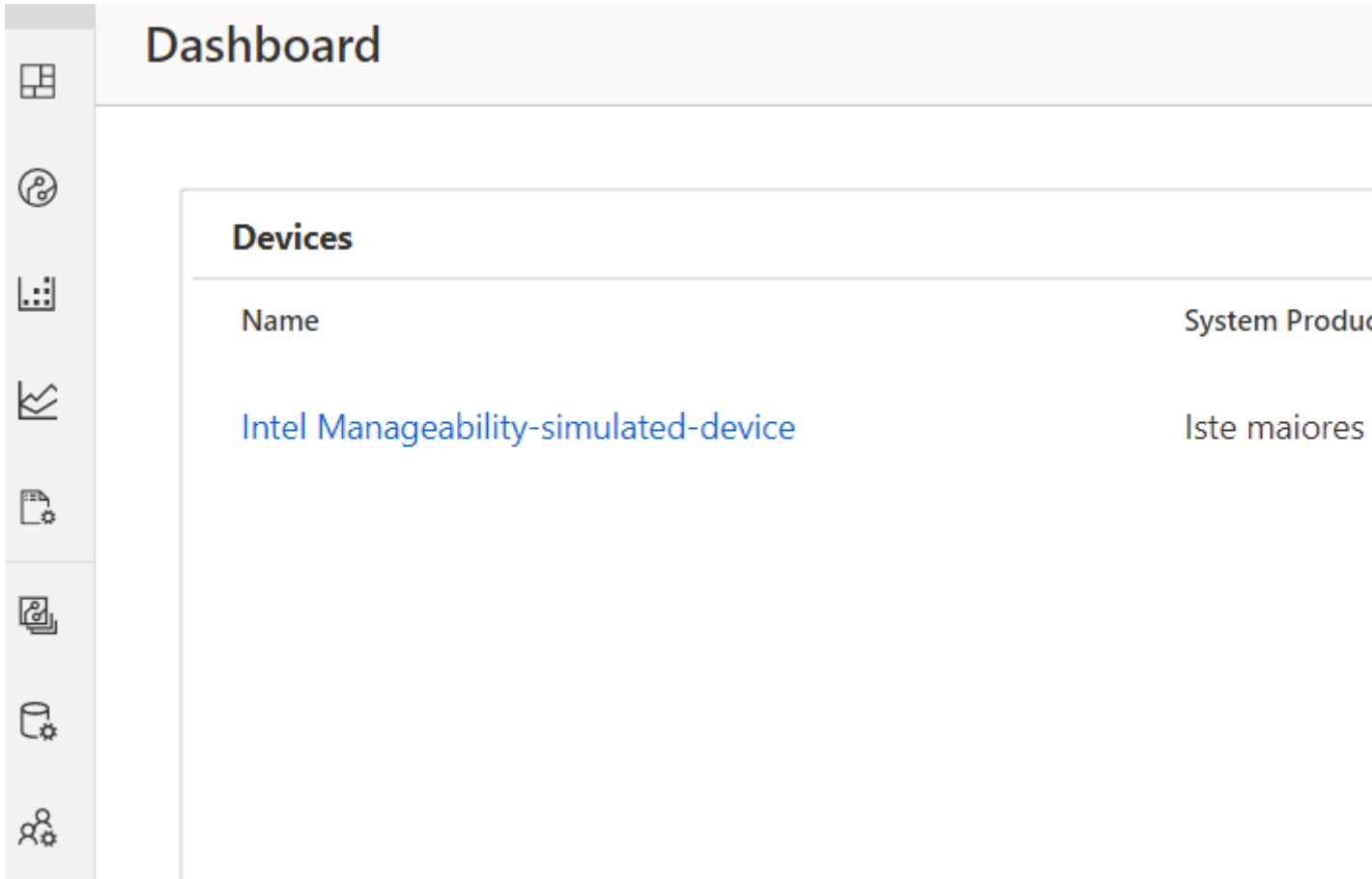
Pricing plan \*

 FreeTry for **7 days** with no commitment

5 free devices

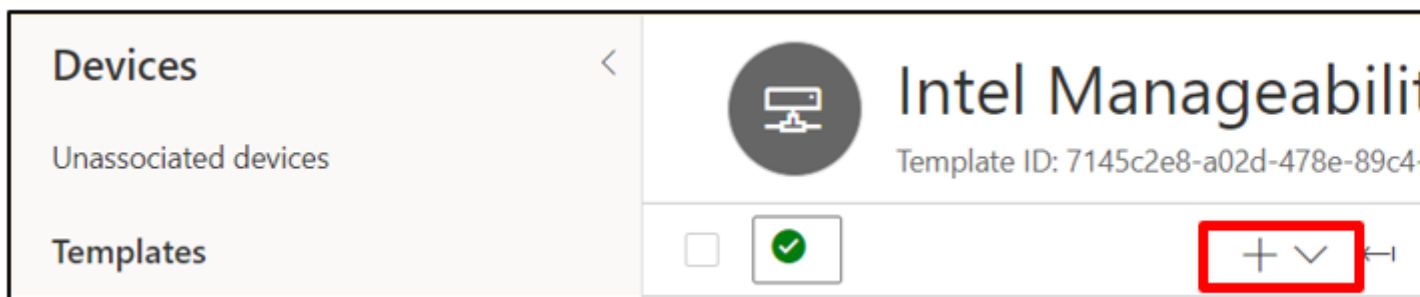
 Standard 1For devices sending a **few messages per hour**2 free devices    **5,000** messages/mo Standard 2 (most popular)For devices sending **messages every few minutes**2 free devices    **30,000** messages/mo

3. For the pricing plan, select **Free** (trial version). Enter the contact information and click create account.
4. After provisioning, the IoT Central application with premade device templates and dashboards will appear. As noted before, this can be accessed at <https://apps.azureiotcentral.com> or through the Azure portal.

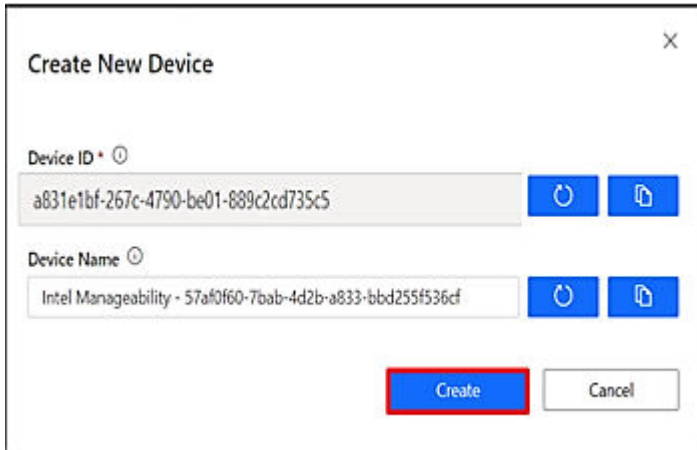


### Obtaining Device Credentials

1. To add a new device, on the side bar click the Devices tab, and then click add new device.



2. Select real on the drop-down list, on the Create New Device form, click Create.



Create New Device

Device ID \* ⓘ  
a831e1bf-267c-4790-be01-889c2cd735c5

Device Name ⓘ  
Intel Manageability - 57af0f60-7bab-4d2b-a833-bbd255f536cf

Create Cancel

3. The new added device will appear on the list of the devices page. To connect the device, click on that specific device and at the top-right bar, click Connect.
4. Device connection information form will appear as follows.

## Device connection

### ID scope ⓘ


0ne000CEFBC

### Device ID ⓘ

a831e1bf-267c-4790-be01-889c2cd735c5

### Credentials

Shared access signature (SAS)    Certificates (X.509)

SAS security tokens are an attestation mechanism for devices to connect to IoT Central. The keys for this device are shown below. Use them to register your device with IoT Central. [more.](#) 

### Primary key ⓘ

V49RzGEoDxUAsP/gRB+VS4llL9jjiQYnSwXJR6QErHs=

### Secondary key ⓘ

Xnh/We2ghEh7A/3YUd+QAuArhSEaXV4+IZOeb7uDyTg=



**Prerequisites** : To perform this step, the Device Manageability device needs to be installed according to the instructions in the Installation Device Manageability section.

1. Provisioning can be done with or without TPM security by setting 'PROVISION\_TPM'. 'PROVISION\_TPM' can be set to the Device Manageability device, users need to run these commands:
  - auto: use TPM if present; disable if not present; do not prompt
  - disable: do not use TPM.
  - enable: use TPM; return error code if TPM not detected.
  - (unset): default behavior; use TPM if present, prompt if not
2. On the Device Manageability device, run the following command:
  - To run provisioning with detecting automatically TPM is present or not:

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

- To run without TPM security:

```
sudo PROVISION_TPM=disable provision-tc
```

3. If the device has been already being configured, it will prompt as follows:

```
Enabling and starting mqtt (this may take some time to generate secrets)
A cloud configuration already exists: "azure"
Replace configuration?
[Y/N] y
```

- Enter 'Y' to replace the existing configuration.
- Select '2' for cloud service option.

```
Please choose a cloud service to use:

1) Telit Device Cloud   3) ThingsBoard
2) Azure IoT Central   4) Custom
#? 2
```

4. A prompt for Device provision type appears; select the type of device authentication preferred: Choose 1 for SAS key authentication. If you choose option 2 - Refer to section [2.5 - Provisioning a Device ] at ./docs/In-Band\_Manageability\_UserGuide\_Azure.pdf for further steps.
5. All the device credentials from obtaining device credentials will be used here. Enter the following information accordingly.

```
Please enter the device Scope ID:
One000CEFBC

Please enter the Device ID:
a831e1bf-267c-4790-be01-889c2cd735c5

Please enter the device SAS Primary Key:
V49RzGEOdxUAsP/gRB+VS411L9jjiQYnSwXJR6QErHs=

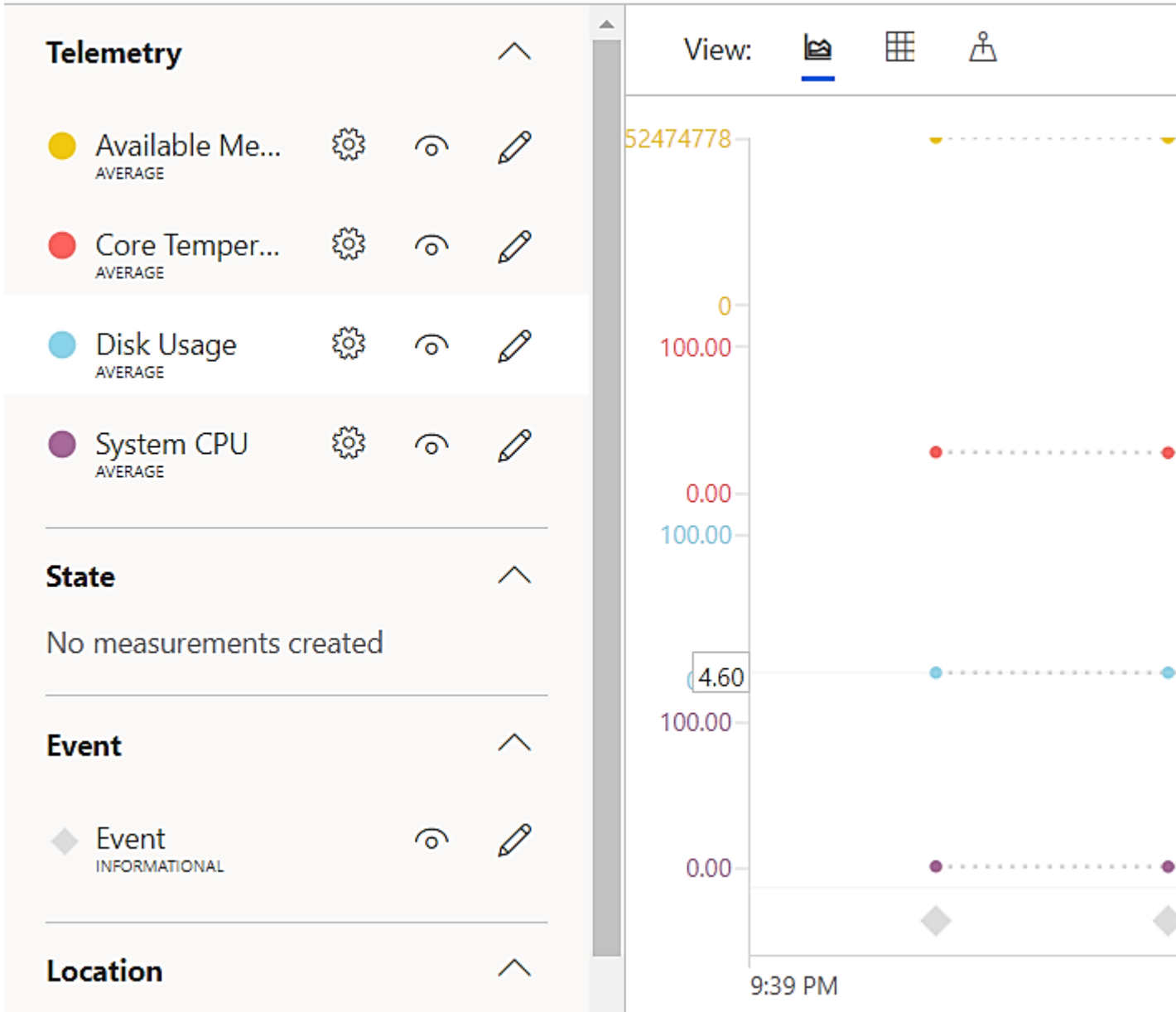
Successfully configured cloud service!

Enabling and starting agents...
Turtle Creek Provisioning Complete
```

6. The script will then start the Intel Manageability services; when the script finishes, the device should be able to interact with its associated IoT Central Application. To verify connectivity:
  - a. Check to see if telemetry/events are appearing. If the information is not seen, use the `systemctl restart cloudadapter telemetry` command to restart the two agents. Allow few seconds for the information to be populated and then refresh the page. b. Alternatively, you can trigger a command like Reboot.
7. Run the provisioning script again, if you want to change or update the cloud service configuration.
8. On the Azure portal, the device status should change from 'Registered' to 'Provisioned' as follows:

<input type="checkbox"/>	Name	Device ID	Device
<input type="checkbox"/>	Intel Manageability - a831e1bf-267c-4790-be01-889c2cd735c5	a831e1bf-267c-4790-be01-889c2cd735c5	No

9. The following is an example of telemetry data of the connected devices that is available to be view on the measurement tab by the user using Azure portal.



- 10. You can get the additional details information for the specific hardware and the event logs on the Properties tab as follows:

**TurtleCreek Azure**

Device **TEST**

Measurements Settings Properties Commands Rules

The dashboard collects your device data and displays it in one centralized location.

**Summary**

System Manufacturer	System Product Name	BIOS Release Date
		<b>2019-05-10 00:00</b>
BIOS Vendor	BIOS Version	CPU
<b>INSYDE Corp.</b>	<b>V25.02.04</b>	<b>Intel(R) Core(TM)</b>
Disk Information	OS Information	Total Memory
<b>[{"NAME": "loop0...</b>	<b>Linux user-2 5.3.0-...</b>	<b>33525768192</b>

**Events**

Event	Time	Category	Value
Event	3/12/2020, 2:25:49 PM	Informational	network-information: {"cards
Event	3/12/2020, 2:25:48 PM	Informational	containers-cpu-percent: {"co
Event	3/12/2020, 2:25:43 PM	Informational	Connected

Azure IoT Central

11. Following is an example of basic information that is shown on the Dashboard tab.

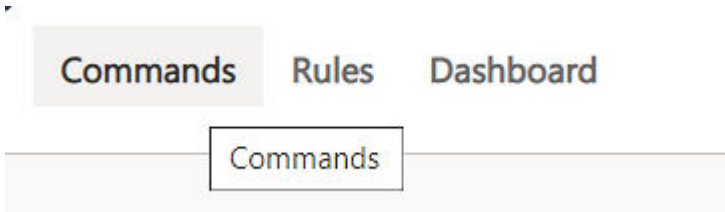
The screenshot displays the TurtleCreek Azure portal interface. On the left is a navigation sidebar with icons and labels for Dashboard, Devices, Device sets, Analytics, Jobs, App settings (Device Templates, Data export, Administration), and a hamburger menu. The main content area is titled 'Device TEST' and includes tabs for Measurements, Settings, Properties (which is selected and highlighted with a dashed box), and Commands. Below the tabs is a 'Save' button. The Properties section contains several fields with their values:

Field Name	Value
System Product Name	[Redacted]
OS Information	Linux user-2 5.3.0-40-generic #32~18.04.1-Ubuntu SMP M
Total Memory (bytes)	33525768192
BIOS Vendor	INSYDE Corp.
BIOS Release Date	2019-05-10 00:00:00

### Performing AOTA Updates Through the Azure Portal

**Prerequisites:** To perform this step, EII bundle need to be generated as mentioned in section Generating EII Bundle for deployment. For testing, you can run the local http server as mentioned in the Creating an HTTP Local Server [ONLY FOR DEVELOPMENT] section. For the actual implementation, you are expected have your own http server or any cloud services that can provide valid URL to fetch the EII bundle.

1. Open the Azure portal, select for the specific device that user will trigger the AOTA.
2. On the menu bar, select **Commands**.



3. On the list command, enter the details in the Trigger AOTA form as follows. To perform this step, generate the EII bundle. For more information, see the **Generating EII Bundle for Deployment** section.

Trigger AOTA ⓘ ↻

App (docker, compose)

Command (down, import, load, pull, snapshot, up, list, stats, remove)

Container Tag

Fetch

4. Above example assumes that the http server is running at **http://10.221.40.65:8000/eii\_bundle.tar.gz** and the EII bundle is generated using the default tag. Leave all other sections empty and click **Run**.

The sample PCB demo example of the EII Visualizer application should appear on the new node. If the visualizer does not appear, run the following command in the new node terminal:

```
xhost +
```

You can also check that EII has been successfully deployed on the new node by checking the list of running containers using the following command:

```
docker ps
```

This action should yield the following terminal output listing the running EII containers

By repeating the AOTA method, you can also stop EII from running on the new node. Instead of passing command up parameter, you can pass command down parameter and the correct eii\_bundle tag. To check if EII has stopped, check the list of running containers.

```
Activities Terminal
File Edit View Search Terminal Help
** (gedit:8403): WARNING **: 16:26:03.110: Set document metadata failed: Set
** (gedit:8403): WARNING **: 16:26:03.110: Set document metadata failed: Set
** (gedit:8403): WARNING **: 16:26:03.360: Set document metadata failed: Set
** (gedit:8403): WARNING **: 16:26:03.360: Set document metadata failed: Set
** (gedit:8403): WARNING **: 16:26:03.630: Set document metadata failed: Set
** (gedit:8403): WARNING **: 16:26:03.630: Set document metadata failed: Set
** (gedit:8403): WARNING **: 16:26:03.837: Set document metadata failed: Set
** (gedit:8403): WARNING **: 16:26:03.837: Set document metadata failed: Set
** (gedit:8403): WARNING **: 16:26:04.400: Set document metadata failed: Set
user@user-2:~$ systemctl restart configuration
user@user-2:~$ systemctl daemon-reload
user@user-2:~$ sudo gedit /etc/intel_manageability.conf
** (gedit:9172): WARNING **: 16:27:22.683: Set document metadata failed: Set
** (gedit:9172): WARNING **: 16:27:22.683: Set document metadata failed: Set
** (gedit:9172): WARNING **: 16:27:22.910: Set document metadata failed: Set
** (gedit:9172): WARNING **: 16:27:22.911: Set document metadata failed: Set
** (gedit:9172): WARNING **: 16:27:23.082: Set document metadata failed: Set
** (gedit:9172): WARNING **: 16:27:23.082: Set document metadata failed: Set
** (gedit:9172): WARNING **: 16:27:23.341: Set document metadata failed: Set
** (gedit:9172): WARNING **: 16:27:23.341: Set document metadata failed: Set
** (gedit:9172): WARNING **: 16:27:24.112: Set document metadata failed: Set
user@user-2:~$ sudo systemctl daemon-reload
user@user-2:~$ sudo systemctl daemon-reload
user@user-2:~$ history | grep xhost
    117  xhost +
    326  xhost +
    473  xhost +
    497  xhost +
    525  history | grep xhost
user@user-2:~$ docker ps
CONTAINER ID        IMAGE                                     COMMAND
ebb9cfff25d71     10.221.40.65:5000/ia_video_analytics:2.1 "VideoAnalyti
a2f523e29f02     10.221.40.65:5000/ia_visualizer:2.1    "python3.6 vi
0980b3c07796     10.221.40.65:5000/ia_video_ingestion:2.1 "VideoIngesti
```

## Verifying Triggered AOTA in Event

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

1. Go to the Device Manageability machine and run the following command to output the log of commands:

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

2. Note the event logs on the Azure portal server showing which commands have been run

---

**NOTE** If the event log does not appear, follow the steps below.

```
Change settings from ERROR to DEBUG everywhere in below files.  
(Only for Development Purpose)  
/etc/intel-manageability/public/dispatcher-agent/logging.ini  
/etc/intel-manageability/public/cloudadapter-agent/logging.ini
```

---

3. Run the following commands

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

## Cloud Service: ThingsBoard\* Setup

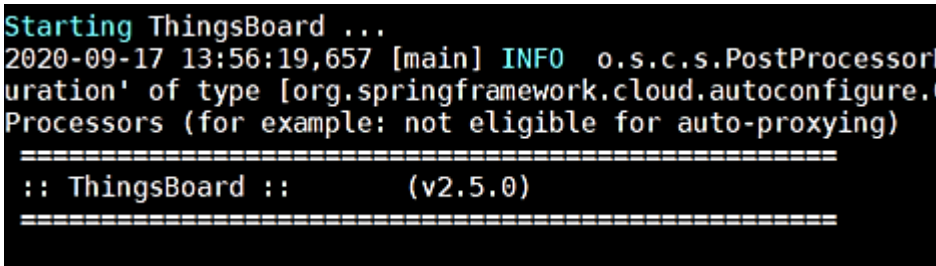
All Device Manageability devices can be added into ThingsBoard to provision or enable the AOTA updates.

### Starting ThingsBoard\* as Docker [macOS\*/Linux\*]

1. Choose an edge device to run as the ThingsBoard server. The edge devices for ThingsBoard machine can be either within the cluster setup or any other edge devices.
2. Run the following command to start ThingsBoard on the ThingsBoard machine.

```
docker run -it -p 9090:9090 -p 1884:2883 -p 5683:5683/udp -v ~/.mytb-data:/data -v ~/.mytb-logs:/var/log/thingsboard -e MQTT_BIND_PORT=2883 -name mytb -restart always thingsboard/tb-postgres:2.5.0
```

Assuming that ThingsBoard has started correctly, you should expect to see the terminal output as shown in figure below.



```
Starting ThingsBoard ...  
2020-09-17 13:56:19,657 [main] INFO o.s.c.s.PostProcessorF  
uration' of type [org.springframework.cloud.autoconfigure.  
Processors (for example: not eligible for auto-proxying)  
=====  
:: ThingsBoard ::      (v2.5.0)  
=====
```

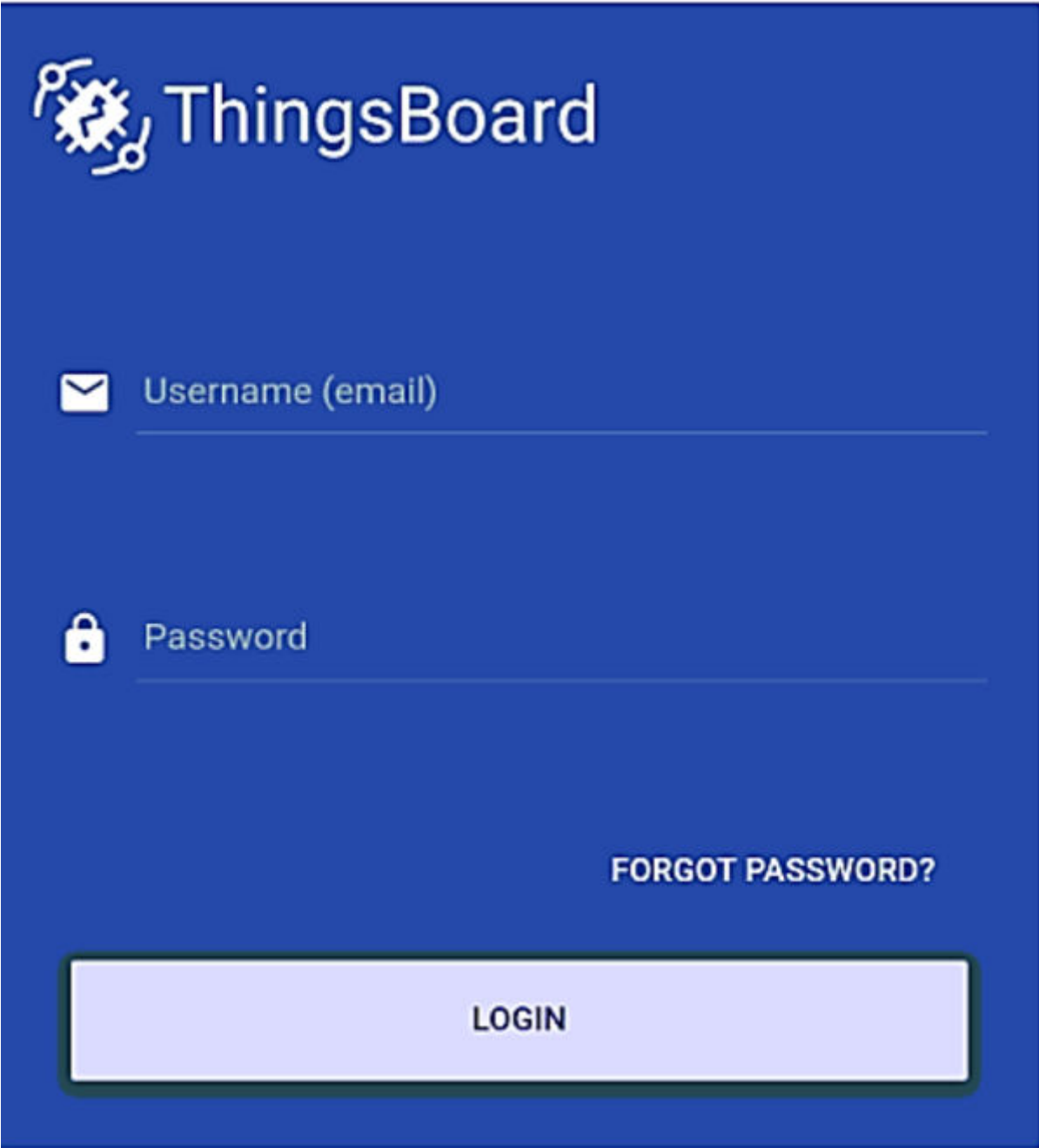
3. After executing the command, the user can browse the ThingsBoard server:

```
<http://<host-ip>:9090>
```

Example: <http://10.221.40.48:9090>

4. ThingsBoard login page should appear, as shown in the following figure:



The image shows the ThingsBoard login page. It has a dark blue background. At the top left is the ThingsBoard logo, which consists of a white gear with a person icon inside, followed by the text "ThingsBoard" in white. Below the logo are two input fields. The first field is labeled "Username (email)" and has an envelope icon to its left. The second field is labeled "Password" and has a lock icon to its left. Below the password field, the text "FORGOT PASSWORD?" is displayed in white. At the bottom of the form is a large, light blue button with the word "LOGIN" in black capital letters.

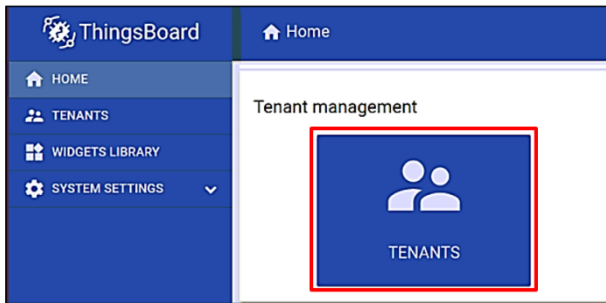
5. Enter the default username and password to access the ThingsBoard home.

Username: `sysadmin@thingsboard.org`

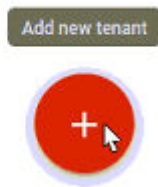
Password: `sysadmin`

#### **Creating a ThingsBoard\* Account**

1. On the homepage, click **Tenant management**, as shown in the following figure:



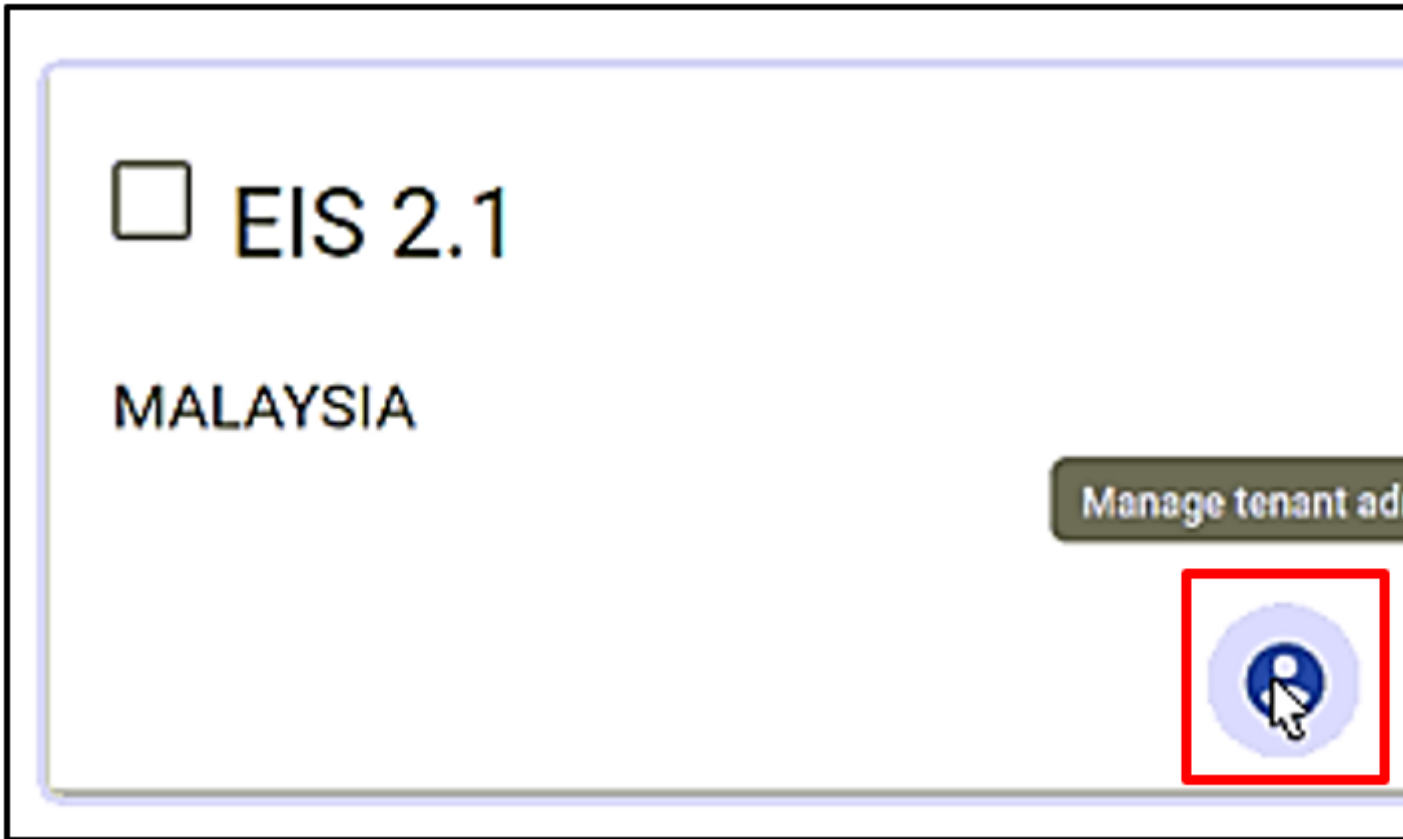
2. At the bottom-right, click the plus button to add a new tenant. Refer to the following figure:



3. The Add Tenant form will appear as shown in the following figure. Complete the form, and then, click **ADD**.

The image shows a web form titled "Add Tenant". The form is contained within a window with a blue header bar. The header bar contains the text "Add Tenant" on the left, a question mark icon in a circle in the center, and a close icon (an 'X') on the right. The form itself is white and contains several input fields: "Title \*" (with an asterisk indicating it is required), "Description", "Country" (with a dropdown arrow), "City", "State / Province", "Zip / Postal Code", "Address", and "Address 2". At the bottom right of the form are two buttons: "ADD" and "CANCEL".

4. New tenant entry will appear in the tenant section as shown in the following figure. Click **Manage tenant admins** to create a new user sign-in username and password.



- 5. Add User form will appear as shown in figure below. Complete the form accordingly. For the Activation Method, click **display activation link**, then click **ADD**.

### User activation link

In order to activate user use the following [activation link](#) :

**<http://10.221.40.48:9090/api/noauth/activate?activateToken=a71J8dEkr0j9UhSWv>**

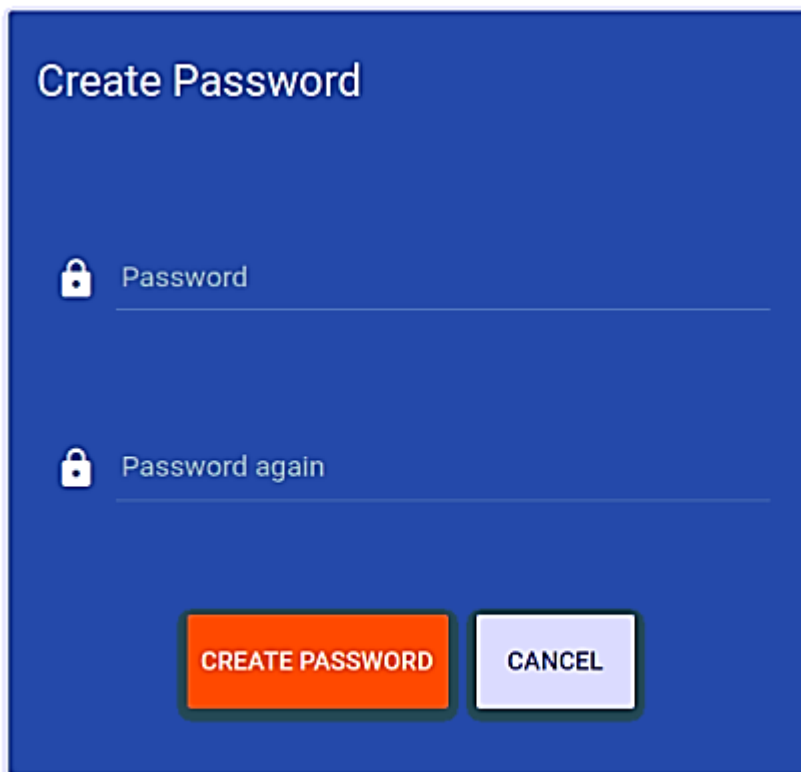
- 6. User activation link box will appear as shown in the following figure below. Click on the activation link.

## User activation link


In order to activate user use the following [activation link](#) :


**`http://10.221.40.48:9090/api/noauth/activate?activateToken=a71J8dEkr0j9UhSWv`**

7. You will be redirected to create a new sign-in password. Click **CREATE PASSWORD** to create the new password as shown in the following figure:



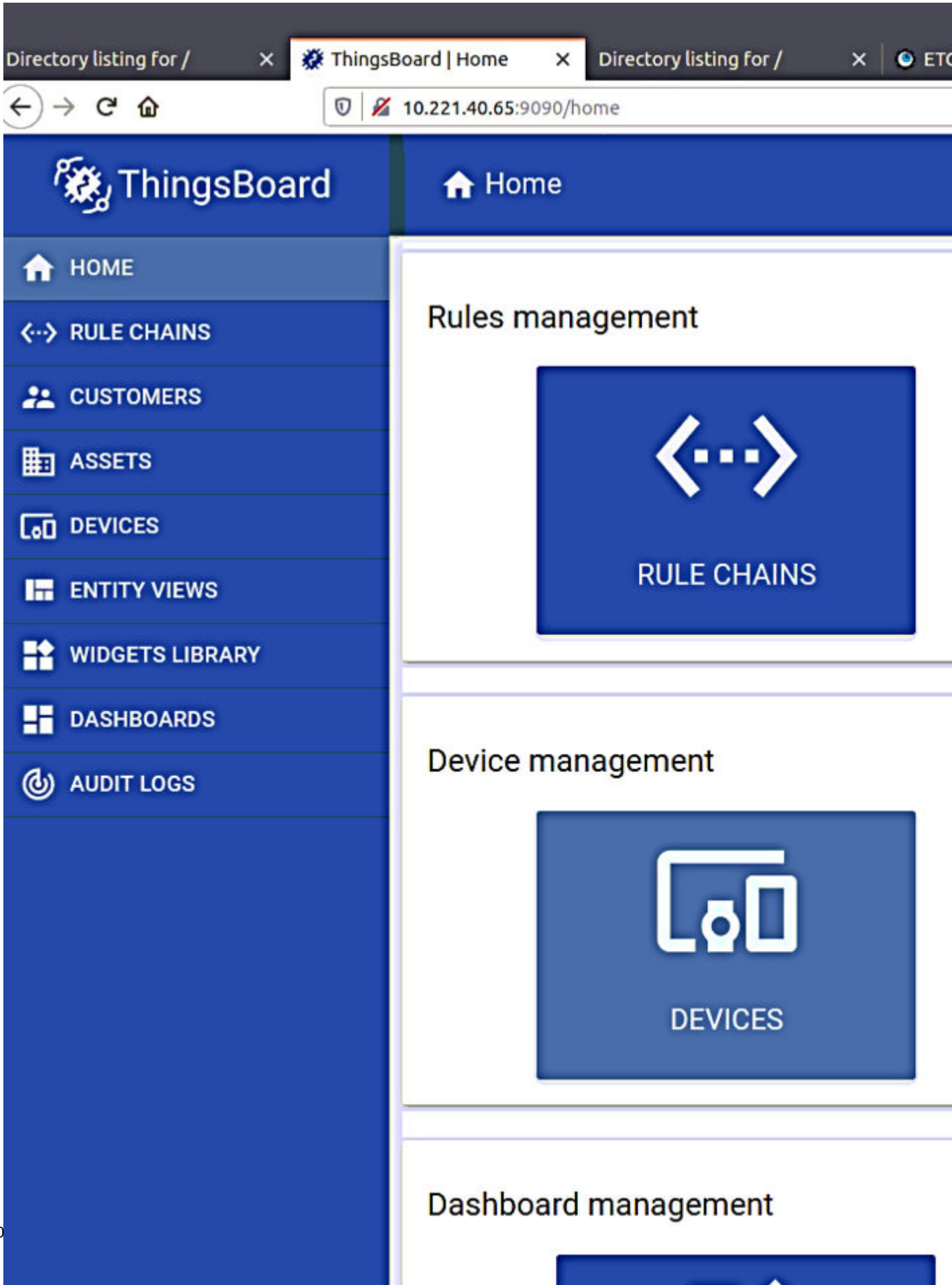
**Create Password**

 Password

 Password again

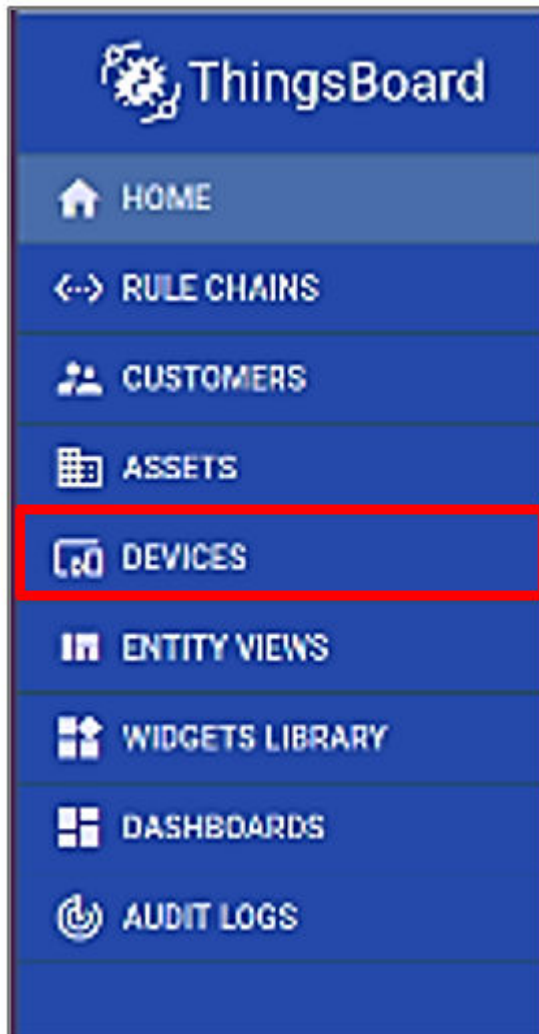
**CREATE PASSWORD** **CANCEL**

8. You can now sign in using the tenant username and password. After which you will be redirected to the ThingsBoard dashboard page as shown in the following figure:

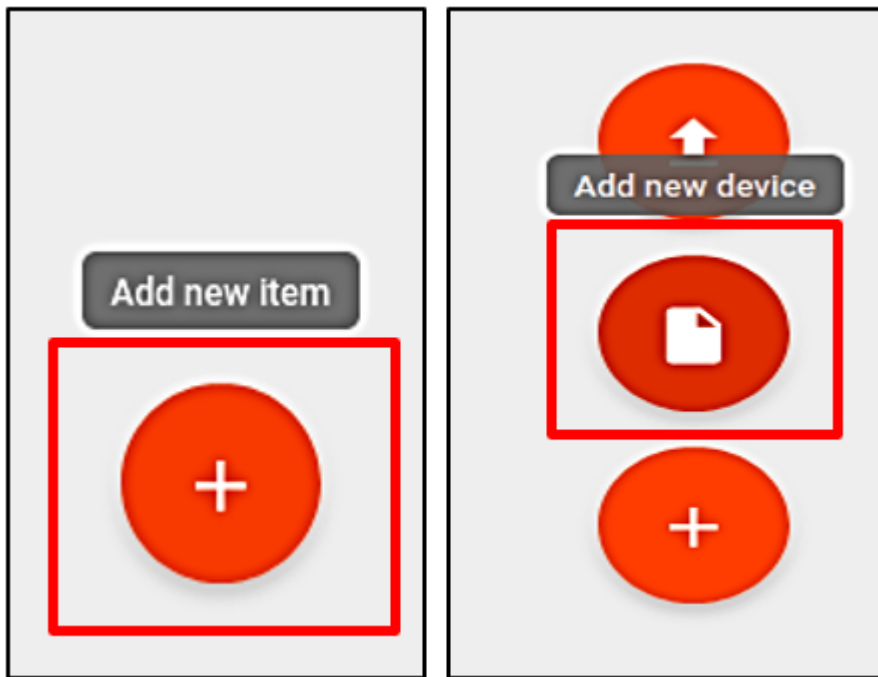


**Adding Device Manageability Device to ThingsBoard\***

1. Add the new Device Manageability device on ThingsBoard.
  - a. Log in to the ThingsBoard page. On the left sidebar, click **Devices** as shown in figure below.



- b. At the bottom-right, click 'plus' and choose to 'Add new device'.

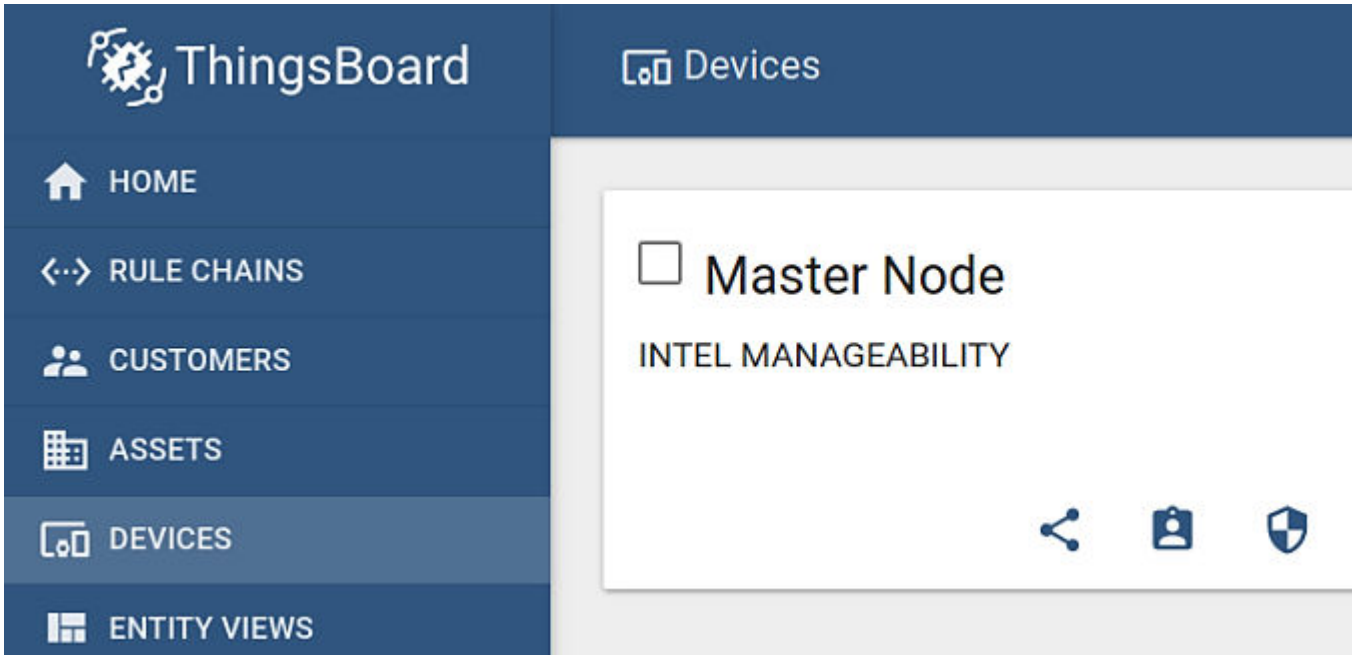


- c. 'Add device' window will appear. Completed the fields, then click **ADD**.

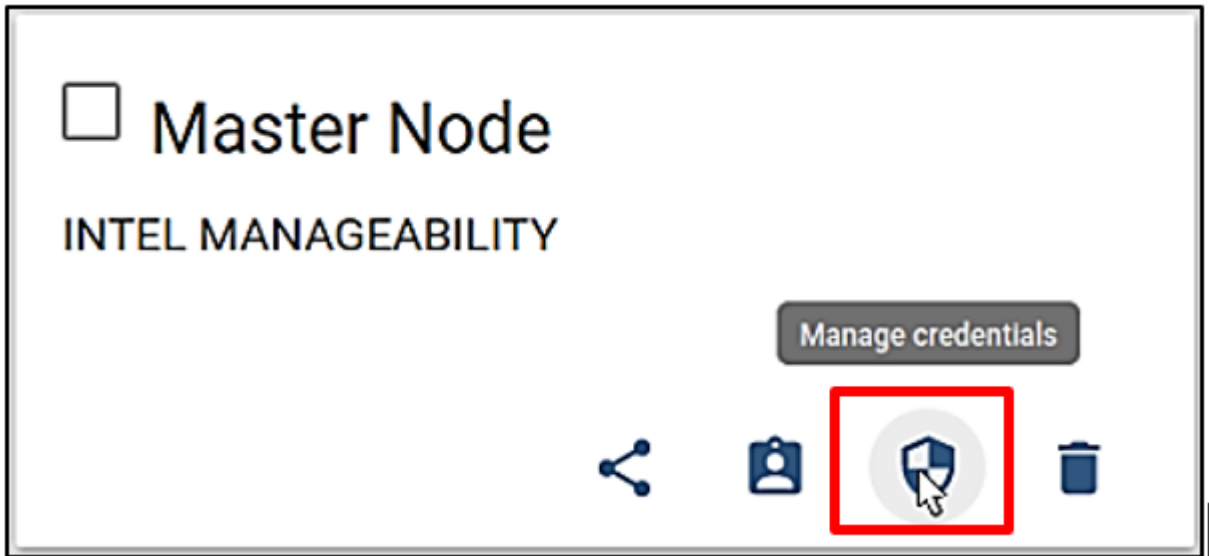


The image displays two side-by-side screenshots of a mobile application's 'Add Device' form. Both screenshots have a dark blue header with the text 'Add Device', a question mark icon, and a close 'X' icon. The form contains several input fields: 'Name \*', 'Device type \*', 'Label', a checkbox for 'Is gateway', and a 'Description' field. In the right-hand screenshot, the 'Name \*' field contains the text 'Master Node'. The 'Device type \*' field contains the text 'Intel Manageability' and is enclosed in a red rectangular box. The 'ADD' button at the bottom right of the form is highlighted in a dark blue color, while in the left-hand screenshot it is greyed out.

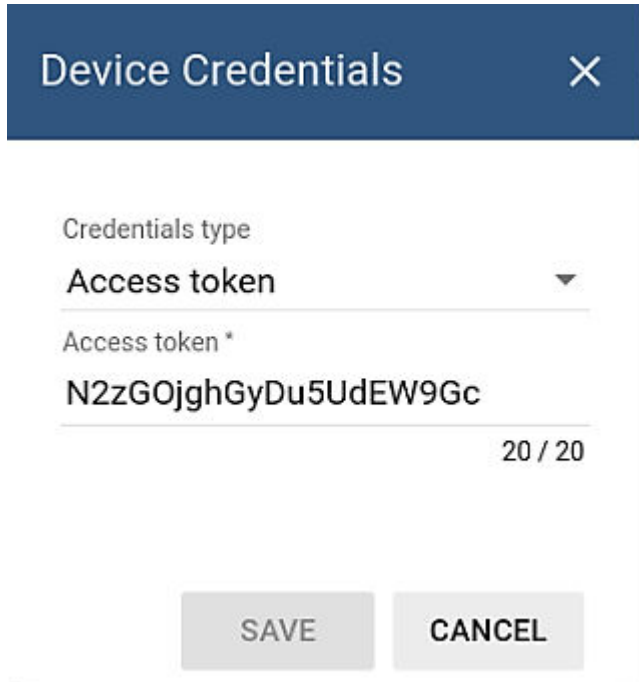
d. Newly added device will be shown on the devices page as shown in the following figure:



- 2. Obtaining device credentials
  - a. Click the manage credentials icon as shown in figure below.



- b. Device credentials window will appear as shown in figure below. Access token is required for provisioning purposes.



Device Credentials

Credentials type

Access token

Access token \*

N2zGOjghGyDu5UdEW9Gc

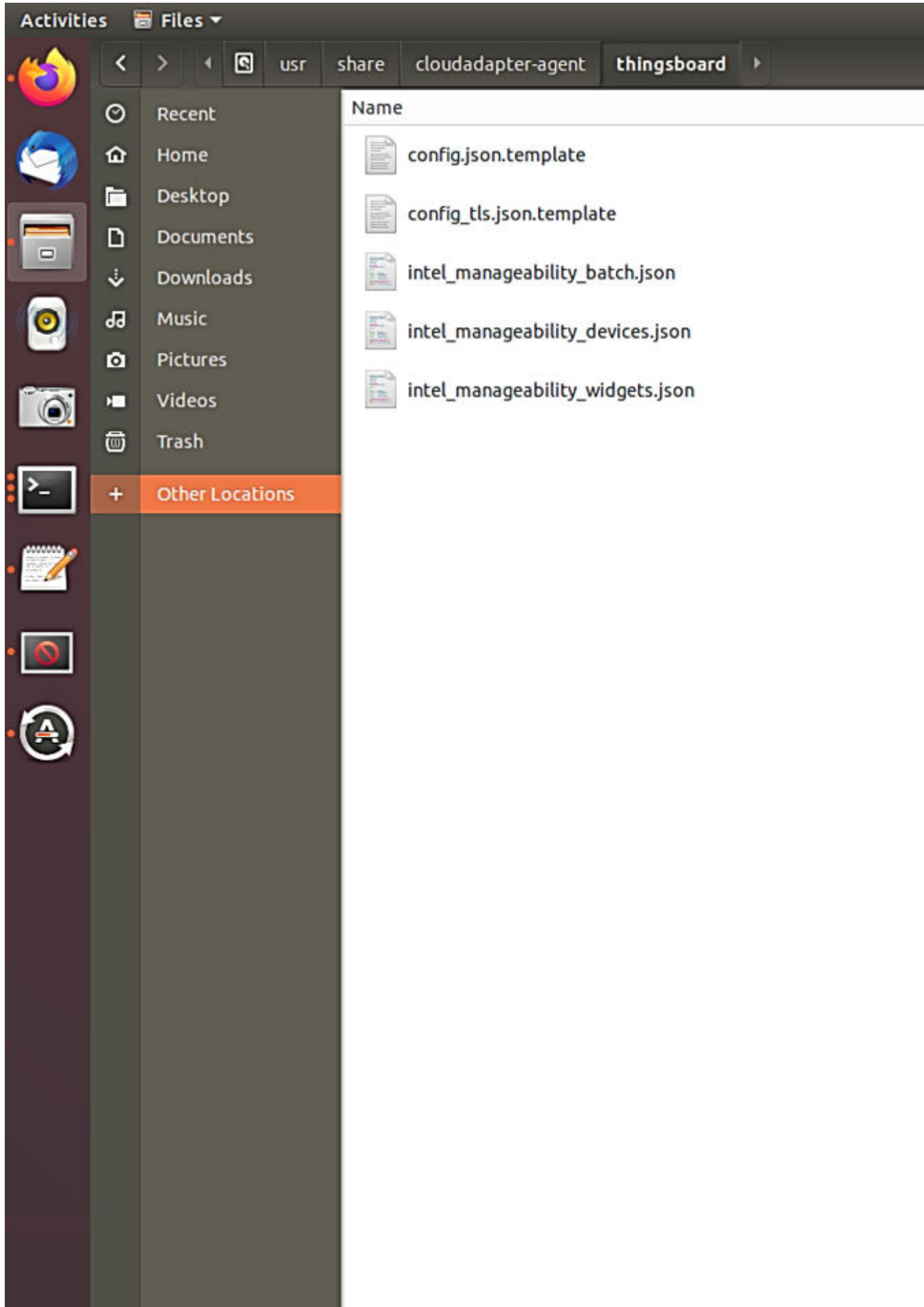
20 / 20

SAVE CANCEL

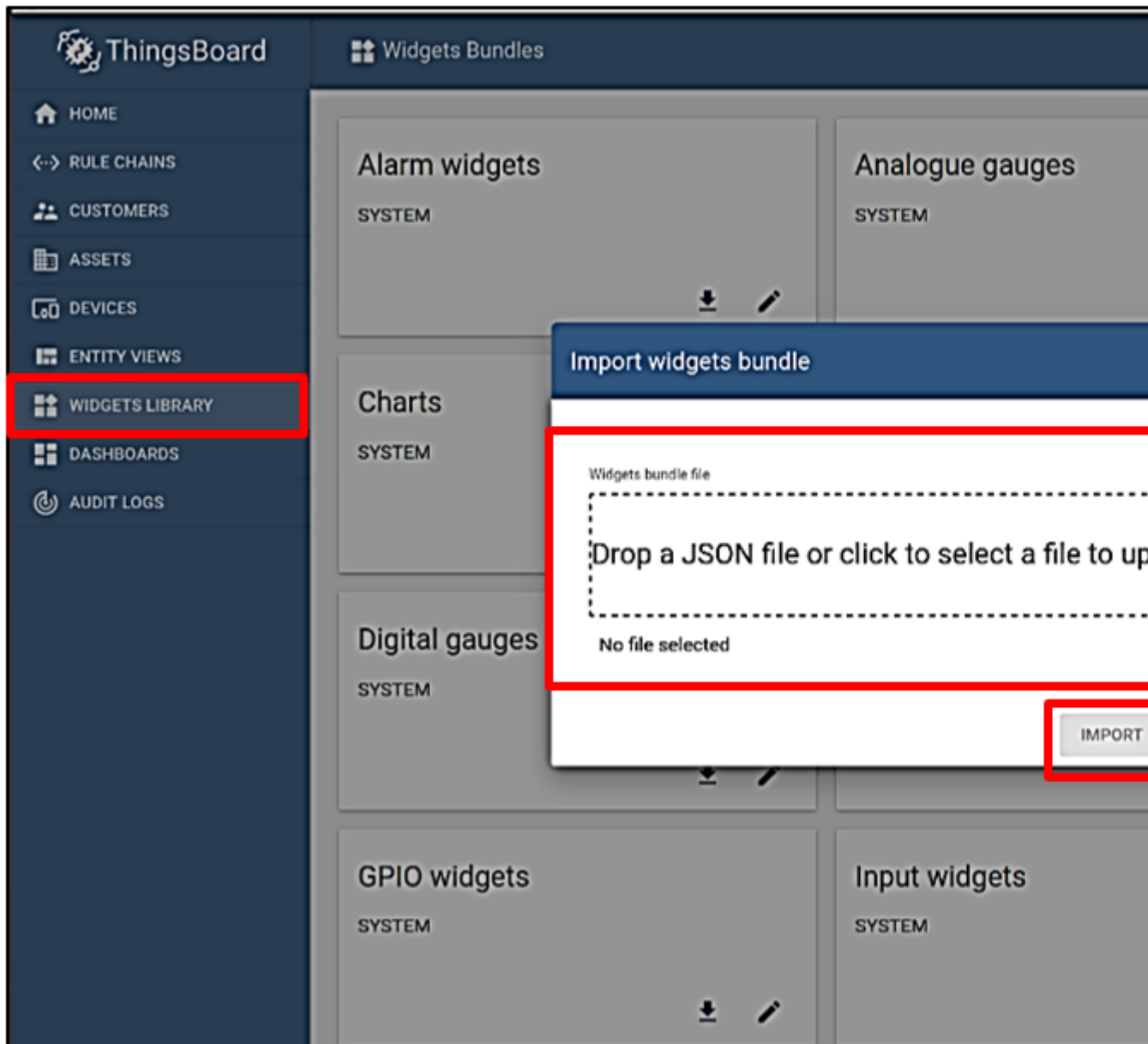
**3.** Setting up the dashboard

- a.** Some files need to be copied from the Device Manageability machine to ThingsBoard page in the next step. These files can be found in the directory below.

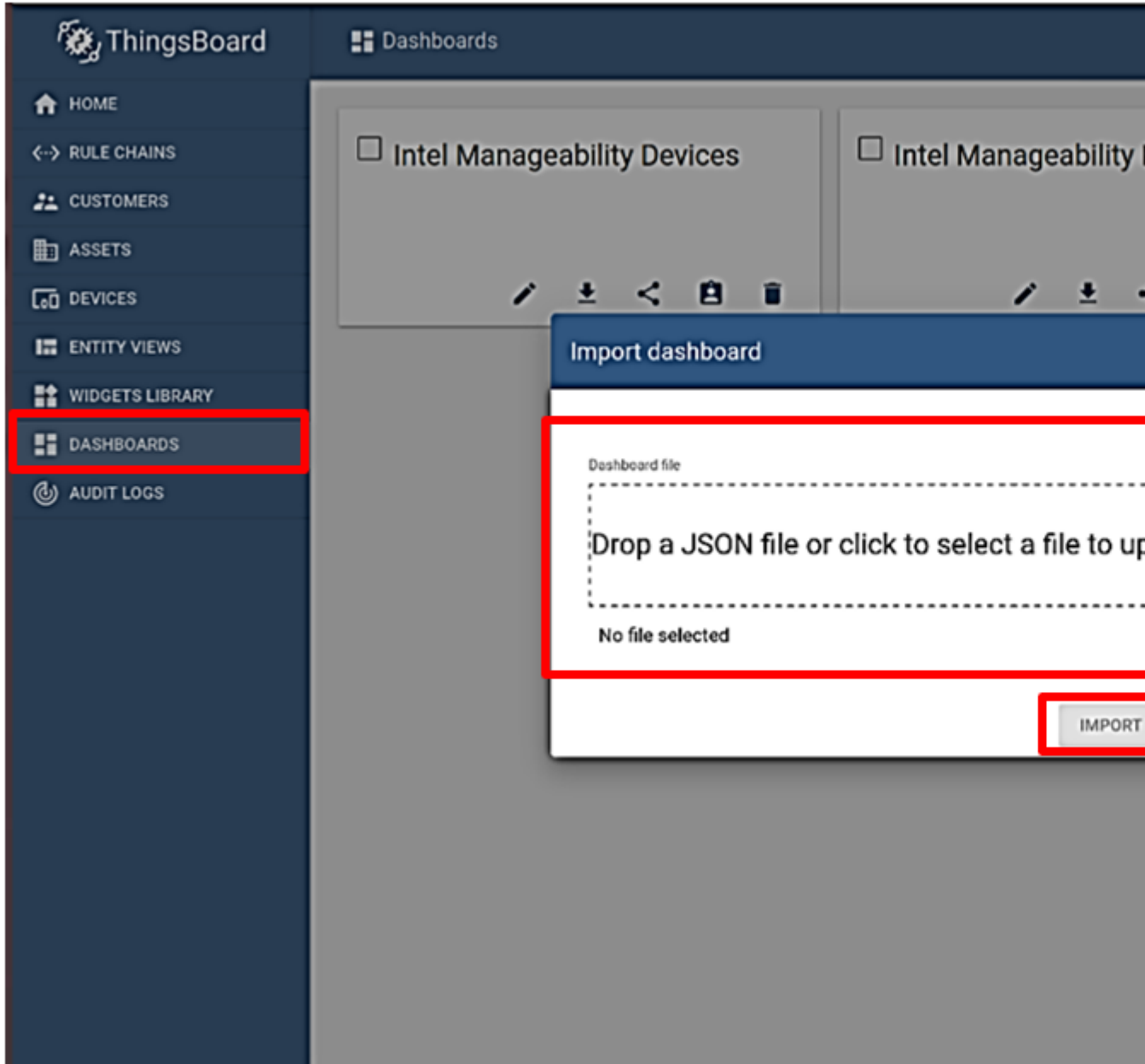
```
cd /usr/share/cloudadapter-agent/thingboard
```



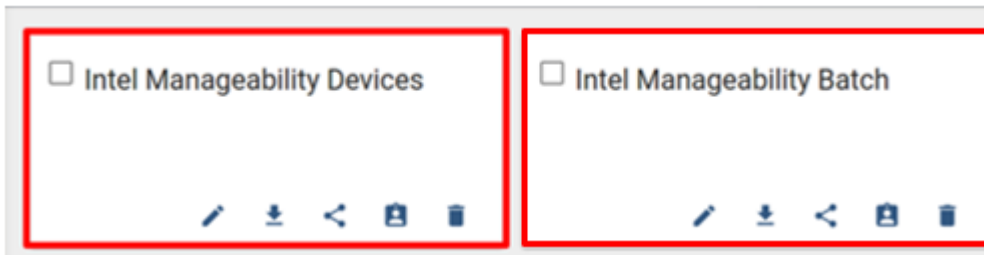
- b. On the left side bar menu, click **Widget Library**, then, click **Add new widget** at the bottom right, and then **Import widgets bundle**. Drag the `intel_manageability_widget.json` file into the box provided and then, click **IMPORT**.



- c. On the left sidebar menu, click **Dashboard**, then at the bottom right click **Add new dashboard**. Drag `intel_manageability_devices.json` into the box provided and then, click **Import**. Repeat this step for `intel_manageability_batch.json`.



- d. Both the newly added dashboards will be displayed.



- e. To access the dashboard, click **Dashboard** on the left sidebar, then on **Intel Manageability Devices**. Basic information about added devices is available here.

If you have more than one device, then you can choose which device to view by clicking the highlighted button on the top-right.

Buttons are also available in the lower part of the screen to view the entire event log, to search for specific event logs, or to expand the window to full screen.

The screenshot shows the ThingsBoard interface for an Intel Manageability device. The left sidebar contains navigation options: HOME, RULE CHAINS, CUSTOMERS, ASSETS, DEVICES, ENTITY VIEWS, WIDGETS LIBRARY, DASHBOARDS, and AUDIT LOGS. The main content area is titled 'Intel Manageability' and shows the status of 'Node 2' as 'Online'. Below this, system information is displayed, including Product, Manufacturer, OS (Linux user-2 5.0.0-37-generic #40~18.04.1-Ubuntu SMP Thu Nov 14 12:06:39 UTC 2019 x86\_64), CPU (Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz), RAM (33576681472 bytes), BIOS Vendor (INSYDE Corp.), BIOS Version (V25 02.04), and BIOS Release Date (2019-05-10 00:00:00). A 'Dynamic Telemetry' section shows various metrics like CPU usage, core temperature, and disk usage. Below the system information are four buttons: SHUTDOWN DEVICE, REBOOT DEVICE, TRIGGER AOTA, and TRIGGER SOTA. At the bottom, an 'Event Log' section shows a table of events with columns for Timestamp and Message. The table shows two events: one at 2020-01-07 09:21:23 with the message 'Disconnected' and another at 2020-01-07 09:21:19 with the message 'Connected'.

Timestamp	Message
2020-01-07 09:21:23	Disconnected
2020-01-07 09:21:19	Connected

## Provisioning a Device

**Prerequisites:** To perform this step, Device Manageability needs to be installed following the instructions in the Installation - Device Manageability section.

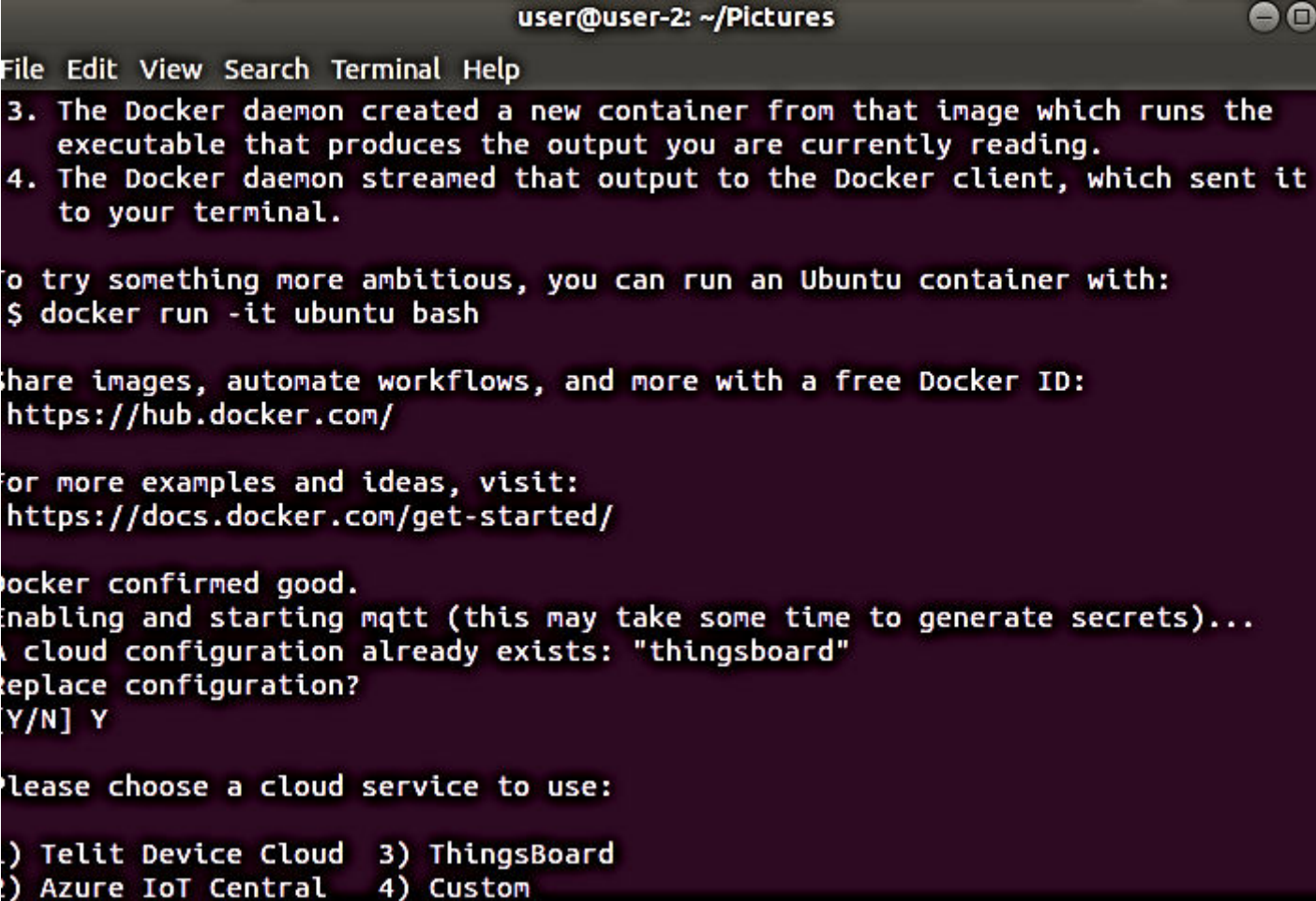
You must provision the Device Manageability device for it to connect to ThingsBoard.

1. Provisioning can be done with or without TPM security by setting 'PROVISION\_TPM'. 'PROVISION\_TPM' can be set to the Device Manageability device, user needs to run these commands:
  - auto: use TPM if present; disable if not present; do not prompt
  - disable: do not use TPM.
  - enable: use TPM; return error code if TPM not detected.
  - (unset): default behavior; use TPM if present, prompt if not
2. On the Device Manageability device, run the following command, and change the bold parameter accordingly as mentioned above.

```
sudo PROVISION_TPM=auto provision
```

Read through the license and press **Y** to accept.

3. If the device was previously provisioned, the following message will appear. To override the previous cloud configuration, press **Y**.



```
user@user-2: ~/Pictures
File Edit View Search Terminal Help
3. The Docker daemon created a new container from that image which runs the
executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

Docker confirmed good.
Enabling and starting mqtt (this may take some time to generate secrets)...
A cloud configuration already exists: "thingsboard"
Replace configuration?
[Y/N] Y

Please choose a cloud service to use:

1) Telit Device Cloud  3) ThingsBoard
2) Azure IoT Central  4) Custom
```

4. Select ThingsBoard as the cloud service by pressing **3** and **[ENTER]**.



```
Please choose a cloud service to use:
```

```
1) Telit Device Cloud   3) ThingsBoard
2) Azure IoT Central    4) Custom
#? 3
```

```
Configuring to use ThingsBoard...
```

5. A prompt for Device provision type appears; select the type of device authentication preferred: Choose 1 for Token authentication. If you choose option2 - Refer section [2.5 - Provisioning a Device ] at ./docs/In-Band\_Manageability\_UserGuide\_ThingsBoard.pdf for further steps.
6. Provide the IP address of the server running ThingsBoard.

```
Please enter the server IP:
10.221.40.48
```

7. Provide the device token based on step 3b.

```
Please enter the device token:
N2zGOjghGyDu5UdEW9Gc
```

8. Use 1884s for the port configuration.

```
Please enter the server port (default 1883):
1884
```

9. Prompts for 'Configure TLS' will appear, type 'N' for this.

```
Configure TLS?
[Y/N] N
```

10. The following screen will appear if cloud provisioning has been completed successfully.

```
Successfully configured cloud service!

Enabling and starting agents...
Turtle Creek Provisioning Complete
```

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

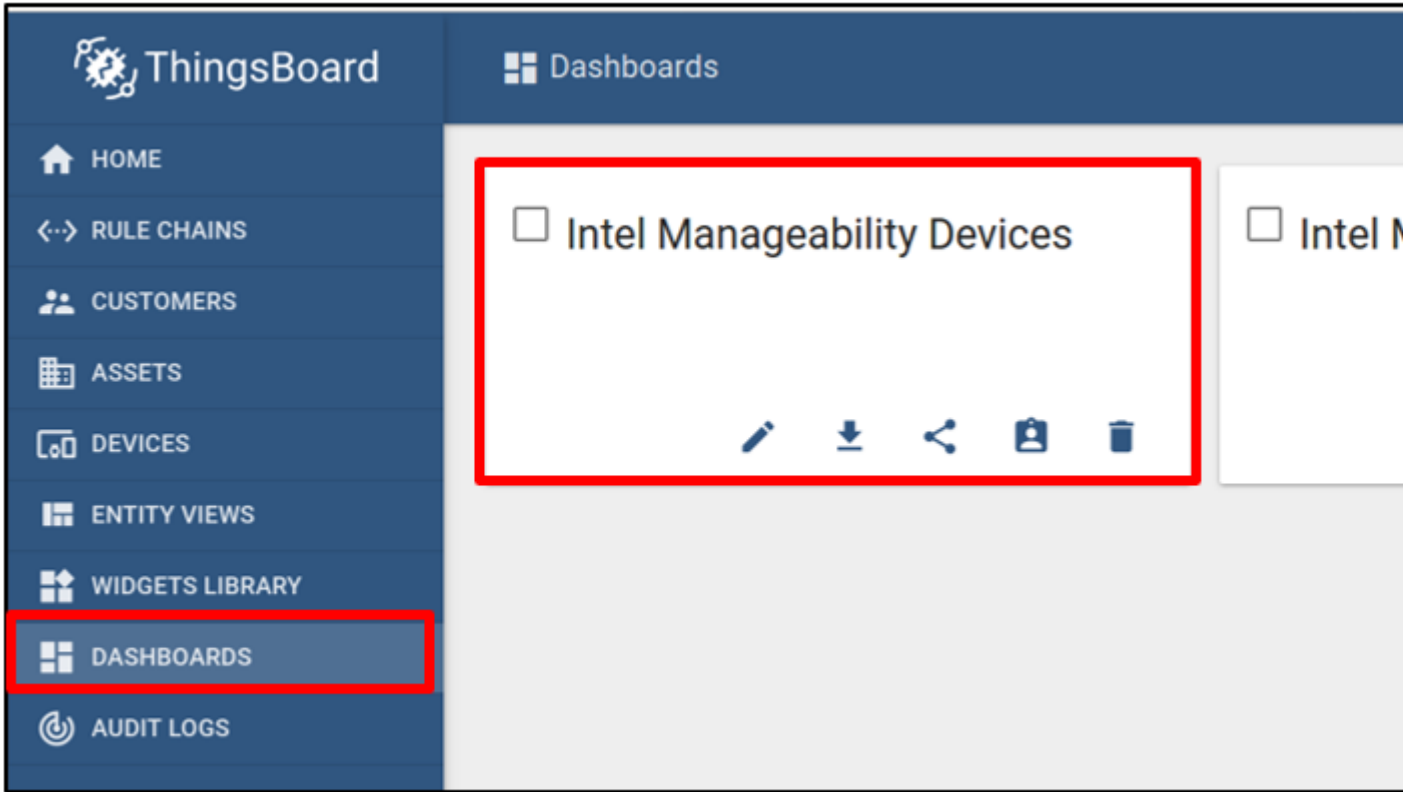
If at any time the cloud service configuration needs to be changed or updated, run this provisioning script again.

To add more than one device, repeat each of the previous steps, except for step 4 (Setting Up the Dashboard).

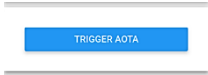
### Performing AOTA Updates Through the ThingsBoard\* Page

**Prerequisites:** To perform this step, generate the EII bundle as mentioned in the Generating EII Bundle for Deployment section and local http server (for development) need to be run as mentioned in Creating an HTTP Local Server [ONLY FOR DEVELOPMENT].

1. Open the ThingsBoard page, then click the "Dashboard" page. Select **Intel Manageability Devices**.



2. Select one of the previously added Device Manageability devices. In this case, you can select the new node (node 2) in which you want to deploy the eii\_bundle.
3. Click **Trigger AOTA**.



4. After the Trigger AOTA window opens, complete each field per the information below:

Trigger AOTA ×

METHOD

RESULTS

App

**docker-compose**



Command

**up**



Container Tag

**eis\_bundle**

Fetch

**http://10.221.40.65:8000/eis\_bundle.tar.gz**

Signature

Version

Server Username

Server Password

For the fetch information, provide the local HTTP server being set up in section Creating an HTTP Local Server [ONLY FOR DEVELOPMENT]. The example uses **http://10.221.40.48:9000/eii\_bundle.tar.gz** in this case.

Leave the other section empty and then, click **Send**.

In the previous step, the new node will access the primary node through the local HTTP server that is being set up to fetch the eii\_bundle.

In the ThingsBoard log, you can see that the eii\_bundle is fetched from the local server and is deployed successfully:

---

```
{'status': 200, 'message': 'COMMAND SUCCESS'}
```

---

```
Package: http://10.221.40.65:8000/eis_bundle.tar.gz Fetch Result: {'status': 200, 'message': 'OK'}
```

---

The sample PCB demo example of the EII Visualizer application should appear on the new node. If the visualizer does not appear, run the following command on the new node terminal.

```
xhost +
```

You can also verify that EII has been successfully deployed on the new node by checking the list of running containers using the following command:

```
docker ps
```

This command will list the EII containers that are running in the terminal output.

```
CONTAINER ID      IMAGE                                     COMMAND
ebb9cff25d71     10.221.40.65:5000/ia_video_analytics:2.1 "VideoAnalytics/va_
a2f523e29f02     10.221.40.65:5000/ia_visualizer:2.1    "python3.6 visualiz
0980b3c07796     10.221.40.65:5000/ia_video_ingestion:2.1 "VideoIngestion/vi_
d8aff3d38637     ia_etcd:2.1                             "./start_etcd.sh"
a0c2e6174211     registry:2                               "/entrypoint.sh /et
```

You can also stop EII from running on the new node by repeating the AOTA method. Instead of passing command up parameter, You can pass the command down parameter and the correct eii\_bundle tag. You can verify that EII has stopped by checking on the list of running containers.

### Verifying Triggered AOTA in Event

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

1. Go to the Device Manageability machine and run the following command to output the log of commands:

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

2. Note the event logs on the ThingsBoard server shows the commands that have been run.

---

**NOTE** If the event log does not appear, follow the steps below.

```
Change settings from ERROR to DEBUG everywhere in below files.
(Only for Development Purpose)
/etc/intel-manageability/public/dispatcher-agent/logging.ini
/etc/intel-manageability/public/cloudadapter-agent/logging.ini
```

---

### 3. Run the following commands:

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

## Get Started Guide

---

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

Before you get started with installation, refer to the [Requirements section](#).

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

---

#### NOTE

##### Important

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

---

Edge Insights for Industrial 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.

If you run into problems installing the software, see [Troubleshooting](#).

After installation, follow the [Introduction to the Edge Software CLI](#) section. 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 Industrial package](#), you must have the following:

### Target System

- 8th generation Intel® Core™ processor onwards with Intel® HD Graphics or Intel® Xeon® processor.
- At least 16 GB RAM for video ingestion and analytics. The recommended RAM for time series analytics pipeline is 4 GB with Intel® Atom processors.
- At least 128 GB hard drive. Recommended 256 GB hard drive.
- An Internet connection.
- Ubuntu\* 20.04 LTS or 18.04

### Knowledge/Experience

You are familiar with executing Linux\* commands.

## Install Edge Insights for Industrial

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. For information on the current release, see the :doc: **Release Notes** </release-notes>.

---

## 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 Industrial that you downloaded. If you need help installing Ubuntu\* Linux, follow these steps:

1. Download [Ubuntu\\* Linux\\* Desktop ISO file](#) to your developer workstation.
2. Use an imaging application to create a bootable flash drive.
3. Power off your target system, insert the USB drive, and power on the system.

If the target system doesn't boot from the USB drive, change the boot priority in the system BIOS.

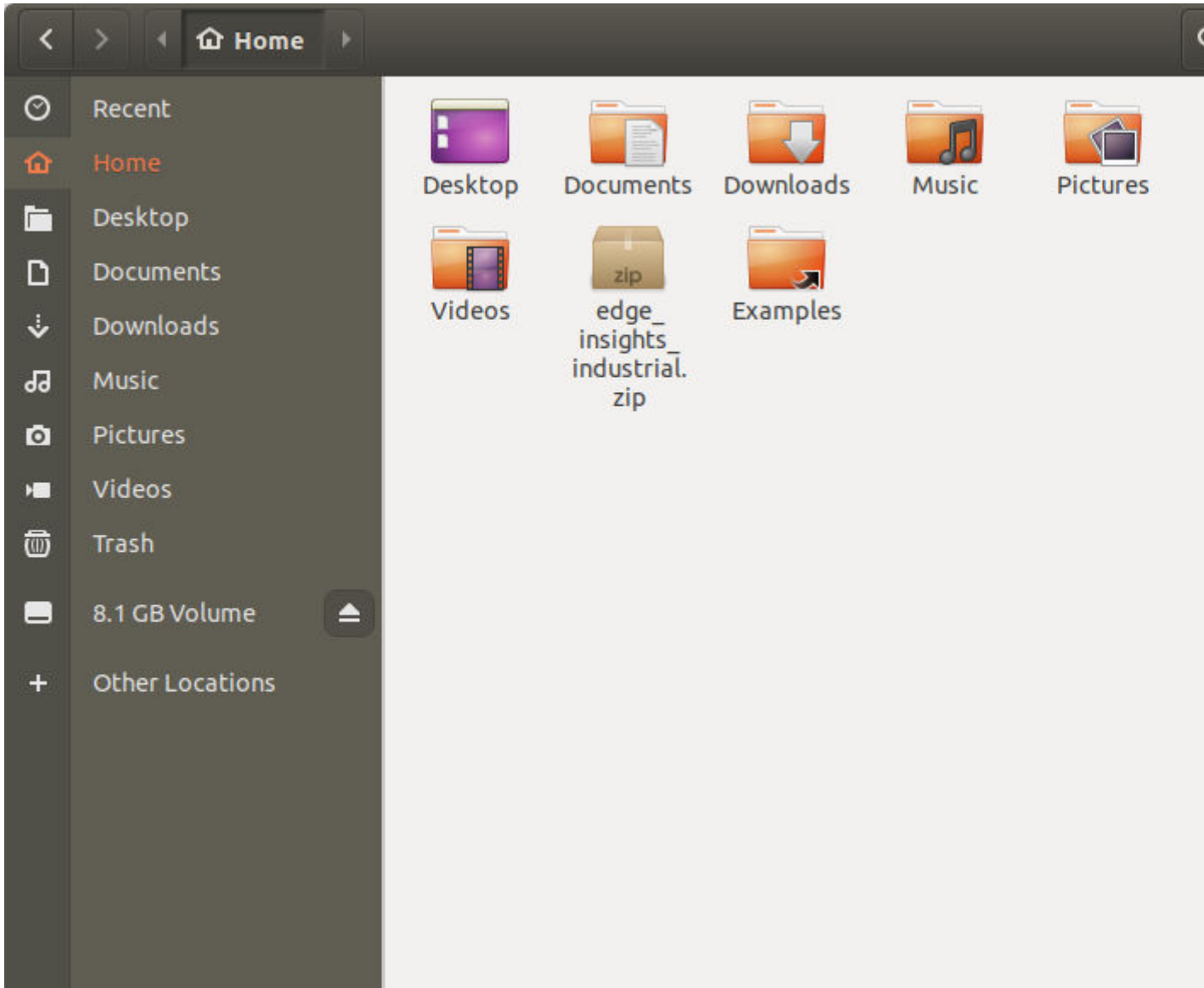
4. Follow the prompts to install Ubuntu\* Linux with the default configurations. For detailed instructions, see the [Ubuntu guide](#).
5. Power down your target system and remove the USB drive.
6. Power up the target system. You will see Ubuntu\* Linux is successfully installed.
7. Log on as the root user:

```
su root
```

## Step 2: Download and Copy the Edge Insights for Industrial .zip File to the Target System

In this step you download the package and copy Edge Insights for Industrial to your target system.

1. Download the [Edge Insights for Industrial package](#).
2. Copy `edge_insights_industrial.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 Industrial Software

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

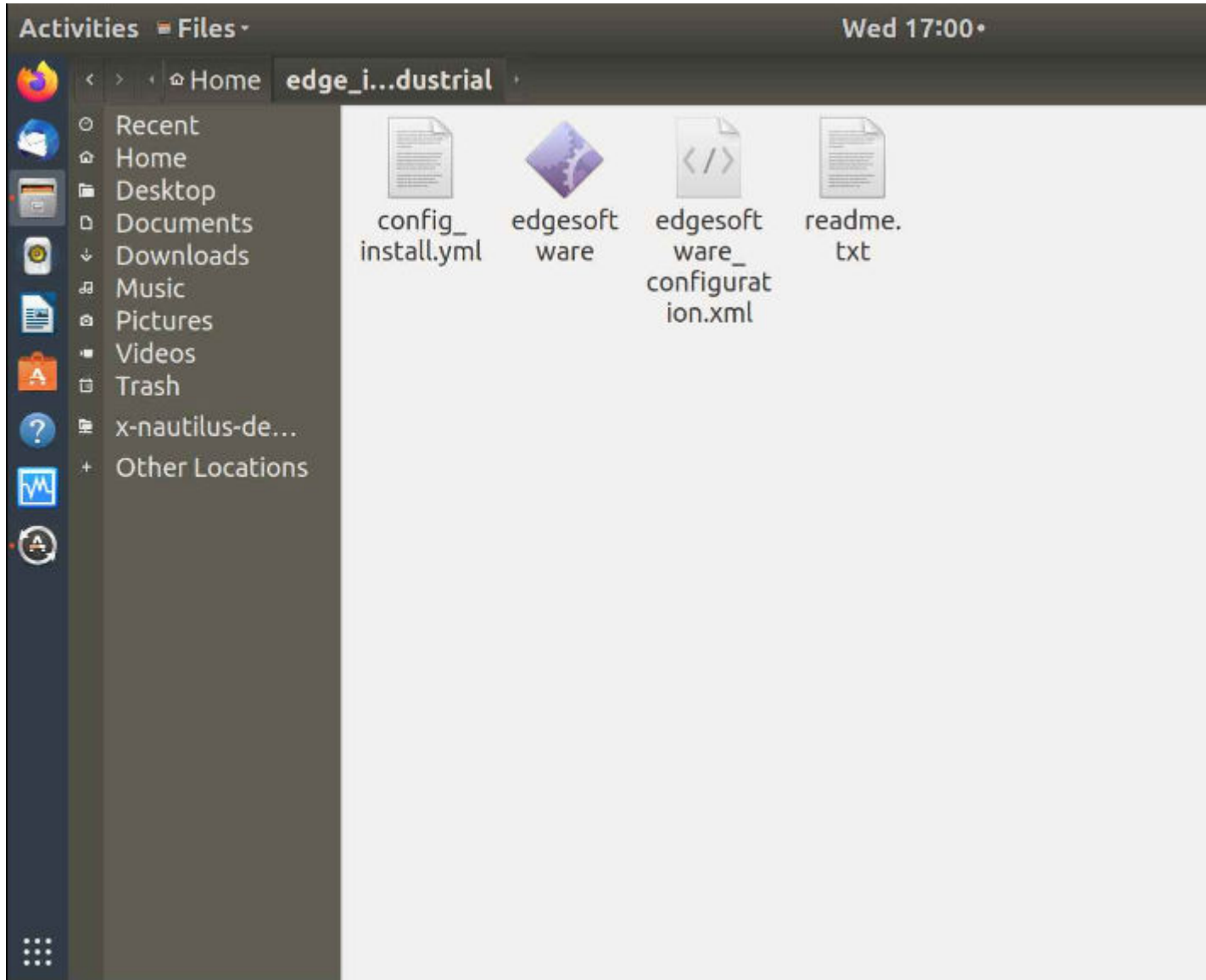
1. Ensure that you have a working internet connection.
2. Open a new terminal.
3. Extract the package:

```
unzip edge_insights_industrial.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: Configure your Installation**

This is a mandatory step. You can use the `config_install.yml` file to configure the installation behavior of the components in your target system.





e  
:  
T  
h  
i  
s  
k  
e  
y  
s  
e  
t  
s  
y  
o  
u  
r  
i  
n  
s  
t  
a  
l  
l  
a  
t  
i  
o  
n  
m  
o  
d  
e  
. S  
e  
t  
t  
h  
e  
k  
e  
y  
v  
a  
l  
u  
e  
a  
s  
f  
o  
l  
l

are enabled. This mode is commonly used for deployment.

- **false:** This value enables the Development mode. In this mode, security and encryption are disabled. This mode is commonly used during evaluation and development as it eases the integration and testing of User Defined Applications.

o  
w  
s  
:  
  
•  
  
•  
  
d  
e  
f  
a  
u  
l  
t  
\_  
u  
s  
e  
\_  
c  
a  
s  
e  
:  
I  
f  
y  
o  
u  
h  
a  
v  
e  
d  
o  
w  
n  
l  
o  
a  
d  
e  
d  
t  
h  
e  
f  
u  
l  
l  
V  
i  
d  
e

- **remove\_previous\_eii:** Set this key value as 'true' if you want to remove the already existing containers of Edge Insights for Industrial previous installation.
- **video-timeseries:** Includes modules for video and timeseries, i.e., the full use-case.
- **video:** Includes only video modules and adds databases for storing images and metadata of analytics results.
- **time-series:** Includes modules for time-series data ingestion, storage and analytics.

o  
A  
n  
a  
l  
y  
t  
i  
c  
s  
+  
T  
i  
m  
e  
S  
e  
r  
i  
e  
s  
E  
d  
g  
e  
I  
n  
s  
i  
g  
h  
t  
s  
f  
o  
r  
I  
n  
d  
u  
s  
t  
r  
i  
a  
l  
p  
a  
c  
k  
a  
g  
e  
,

Y  
O  
U  
C  
A  
N  
S  
E  
T  
T  
H  
E  
T  
Y  
P  
E  
O  
F  
U  
S  
E  
C  
A  
S  
E  
Y  
O  
U  
W  
A  
N  
T  
T  
O  
I  
N  
S  
T  
A  
L  
L  
U  
S  
I  
N  
G  
T  
H  
I  
S  
K  
E  
Y  
V

```
a  
l  
u  
e  
a  
s  
b  
e  
l  
o  
w  
:
```

- **etcdroot\_password:** Updates the etcdroot password credential.
- **influxdb\_username:** If the downloaded edgesoftware\_configuration.xml file contains the InfluxDBConnector module. Set the username credential value. The following special characters are not allowed for the **influxdb\_username**: `'+[@^{\%(-"*|, &<` } . _ =} !>;?#\$) \`
- **influxdb\_password:** If the downloaded edgesoftware\_configuration.xml file contains the InfluxDBConnector module. Set the password credential value. The following special characters are not allowed for the **influxdb\_password**: `'+[@^{\%(-"*|, &<` } . _ =} !>;?#\$) \`
- **webvisualizer\_username:** If the downloaded edgesoftware\_configuration.xml file contains WebVisualizer module. Set the username credential value.
- **webvisualizer\_password:** If the downloaded edgesoftware\_configuration.xml file contains WebVisualizer module. Set the password credential value.
- **minio\_access\_key:** If the downloaded edgesoftware\_configuration.xml file contains ImageStore module. Set the access key credential value. The minio\_access\_key length must be a minimum of 8 characters.
- **minio\_secret\_key:** If the downloaded edgesoftware\_configuration.xml file contains ImageStore module. Set the secret key credential value. The minio\_secret\_key length must be a minimum of 8 characters.

## Step 5: Install the Edge Insights for Industrial Software

---

**NOTE** If you are running behind a proxy server, ensure 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 Industrial on your target system.

The software installation can take 1 to 3 hours depending on the package you chose. The completion time depends on your target system and Internet connection.

**1.** Run these commands:

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

**NOTE** If you encounter any Docker\* pull-related issues during the installation process, refer to the :doc: **Troubleshooting** </install-edge-insights-for-industrial> section at the end of this document.

2. Type the product key at the prompt:

```
intel@edgesoftware: ~/edge_insights_industrial
File Edit View Search Terminal Help
intel@edgesoftware:~/edge_insights_industrial$ ./edgesoftware install
Please enter the Product Key. The Product Key is contained in the email you received from Intel confi
[REDACTED]
Starting the setup...
ESB CLI version: 2020.3
Target OS: Ubuntu
-----SOFTWARE INFO-----
Operating System: Ubuntu 18.04.4 LTS
Kernel: 5.4.0-42-generic
-----HARDWARE INFO-----
Hardware Architecture: x86_64
Processor: Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz
Memory size: 15G
Total disk space (GB): 468
Free disk space (GB): 432
-----HARDWARE ACCELERATOR-----
No hardware accelerator found
Python version: 3.6.9
Checking Internet connection
Connected to the Internet
Checking for prerequisites
All dependencies met
Validating product key
Successfully validated Product Key
Starting installation
Downloading modules...
Downloading component esb_common
ZIP file for module 5e8c4742e02f17002a2a6976 already exists. Validating it...
Module validation passed for 5e8c4742e02f17002a2a6976
Skipping download...
Downloading component Docker_Community_Edition_CE
ZIP file for module 5f21392e9e63c9002a6fd88d already exists. Validating it...
Module validation passed for 5f21392e9e63c9002a6fd88d
Skipping download...
Downloading component Docker_Compose
ZIP file for module 5f213aae9e63c9002a6fd88e already exists. Validating it...
Module validation passed for 5f213aae9e63c9002a6fd88e
Skipping download...
Downloading component IEdgeInsights
ZIP file for module 5ef1c13e1c1b39002a68448b already exists. Validating it...
Module validation passed for 5ef1c13e1c1b39002a68448b
Skipping download...
Downloading component IEdgeInsights/Samples
ZIP file for module 5ef2e867fa5828002aa646da already exists. Validating it...
```

**NOTE** See the Edge Insights for Industrial based Reference Implementation (RI) tutorial requirements in the table below.

---

S.No	RI Tutorial	Applicable Use Case	Requirement
1	Defect Detection Demo	<b>Video Analytics</b> or <b>Video Analytics and Time series</b>	Download the installer package through the <b>Download Recommended Configuration</b> option by choosing one of the use cases (see Applicable Use Case column) in the <a href="#">Edge Insights for Industrial package</a> .

Based on components selected and system configuration, you might be prompted for additional actions if you have not configured the `config_install.yml` file correctly.

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



```

Activities  Terminal  Thu 15:34
intel@edgesoftware: ~/edge_insights_industrial

File Edit View Search Terminal Help
Creating edgeinsightssoftware_ia_azure_simple_subscriber_1 ... done
Creating ia_grafana ... done
Creating ia_video_ingestion ... done
Verifying Docker images
SUCCESS : Verified openedgeinsights/ia_azure_bridge
SUCCESS : Verified openedgeinsights/ia_azure_simple_subscriber
SUCCESS : Verified openedgeinsights/ia_factoryctrl_app
SUCCESS : Verified openedgeinsights/ia_grafana
SUCCESS : Verified openedgeinsights/ia_imagestore
SUCCESS : Verified openedgeinsights/ia_influxdbconnector
SUCCESS : Verified ia_kapacitor
SUCCESS : Verified openedgeinsights/ia_opcua_export
SUCCESS : Verified openedgeinsights/ia_rest_export
SUCCESS : Verified ia_tls_remoteagent
SUCCESS : Verified openedgeinsights/ia_telegraf
SUCCESS : Verified openedgeinsights/ia_video_analytics
SUCCESS : Verified openedgeinsights/ia_video_ingestion
SUCCESS : Verified openedgeinsights/ia_visualizer
SUCCESS : Verified openedgeinsights/ia_web_visualizer
SUCCESS : Verified openedgeinsights/ia_zmq_broker
Building Insights Images [.....]
Building Insights Images [.....]
100%
Edge Insights Visualizer [
Edge Insights Visualizer [.....]
Edge Insights Visualizer [.....]
100%
Successfully installed eii_installer took 5 minutes 39.83 seconds
Installation of package complete
***Recommended to reboot system after installation***
+-----+-----+
|          Id          |          Module          |          Status          |
+-----+-----+
| 5f21392e9e63c9002a6fd88d | Docker Community Edition CE | SUCCESS |
| 5f213aae9e63c9002a6fd88e | Docker Compose           | SUCCESS |
| 60c727ad4b40e5002ad9b795 | eii installer             | SUCCESS |
+-----+-----+
intel@edgesoftware:~/edge_insights_industrial$

```

**NOTE** For Time series or Video Analytics and Time series use cases, the demo for time series does not start automatically, unlike the video use case. Follow the step below to see the time series demonstration:

- Launch Grafana to see the time series visualization. For details, refer to [Open Edge Insights Grafana](#) information.
- 

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

## Steps for Manual Installation of Manageability Module

**NOTE** Manageability is downloaded by default, but not auto-installed.

---

- Go to `edge_insights_industrial/Edge_Insights_for_Industrial_<version>/manageability/` where `<version>` indicates the downloaded version of Edge Insights for Industrial.
- Refer to the steps in `README.md` for manual installation of manageability component.

## 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.
- If the `eii installer` component fails to install after running the `./edgesoftware install` command, check the running containers list using `docker ps` command and check the logs of the failing containers to find out why they are failing. Mostly services like InfluxDBConnector, Grafana, Imagestore, etc., fail if the port is already used by other process on the host m/c as EII uses `docker host` network. Please make sure to free up that port and re-run the `./edgesoftware install` command or follow the steps below:
  - Go to `edge_insights_industrial/Edge_Insights_for_Industrial_<version>/IEdgeInsights/build` where `<version>` indicates the downloaded version of Edge Insights for Industrial.
  - Run `sudo sg docker -c docker-compose up -d` command to relaunch services.

## Restart Mode

The following services may be in **restart mode** after successful installation. Refer to the respective `README.md` files to troubleshoot, where `<version>` indicates the downloaded version of Edge Insights for Industrial:

- **Visualizer:**`edge_insights_industrial/Edge_Insights_for_Industrial_<version>/IEdgeInsights/Visualizer/README.md`
- **FactoryControlApp:**`edge_insights_industrial/Edge_Insights_for_Industrial_<version>/IEdgeInsights/FactoryControlApp/README.md`
- **RestDataExport:**`edge_insights_industrial/Edge_Insights_for_Industrial_<version>/IEdgeInsights/RestDataExport/README.md`

## Docker Image Pull Issue

This issue can be because of the latest pull rate limitations introduced by Docker hub.

- Please check this [Docker site article](#) to help determine the exact pull limit that's applicable on the system where you are trying to pull the publicly available Docker hub images, such as Docker, Python, etc.

---

**NOTE** This limit is only applicable for the 6-hour window.

---

- If you see this issue with an anonymous user (pull limit of 100), i.e., without Docker login, you can [create an account](#) and try to do the build after running the command: `docker login -u <username> -p <password>`. The alternative is to use a [paid subscription](#).

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

## Introduction to the Edge Software 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.

---

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

---

To begin:

1. Open a terminal window.
2. Go to the `edge_insights_industrial/` 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.
```

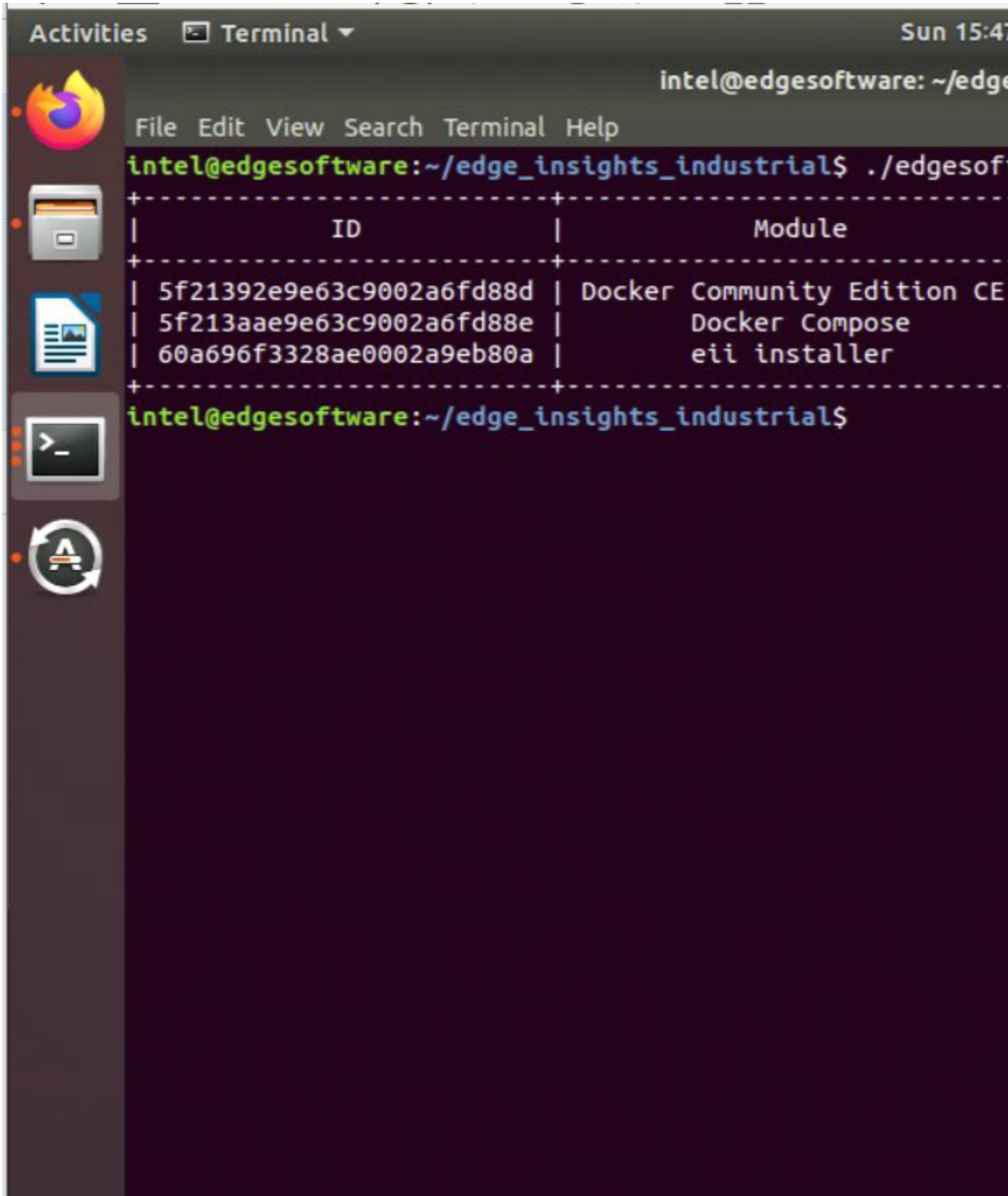
Commands:

```
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.



Terminal window showing the execution of a command and its output. The prompt is `intel@edgesoftware:~/edge_insights_industrial$`. The command `./edgesof` is executed, resulting in a table with two columns: ID and Module.

```
File Edit View Search Terminal Help
intel@edgesoftware:~/edge_insights_industrial$ ./edgesof
+-----+-----+
|          ID          |          Module          |
+-----+-----+
| 5f21392e9e63c9002a6fd88d | Docker Community Edition CE |
| 5f213aae9e63c9002a6fd88e | Docker Compose             |
| 60a696f3328ae0002a9eb80a | eii installer              |
+-----+-----+
intel@edgesoftware:~/edge_insights_industrial$
```

## View the Software Version

- Command:

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

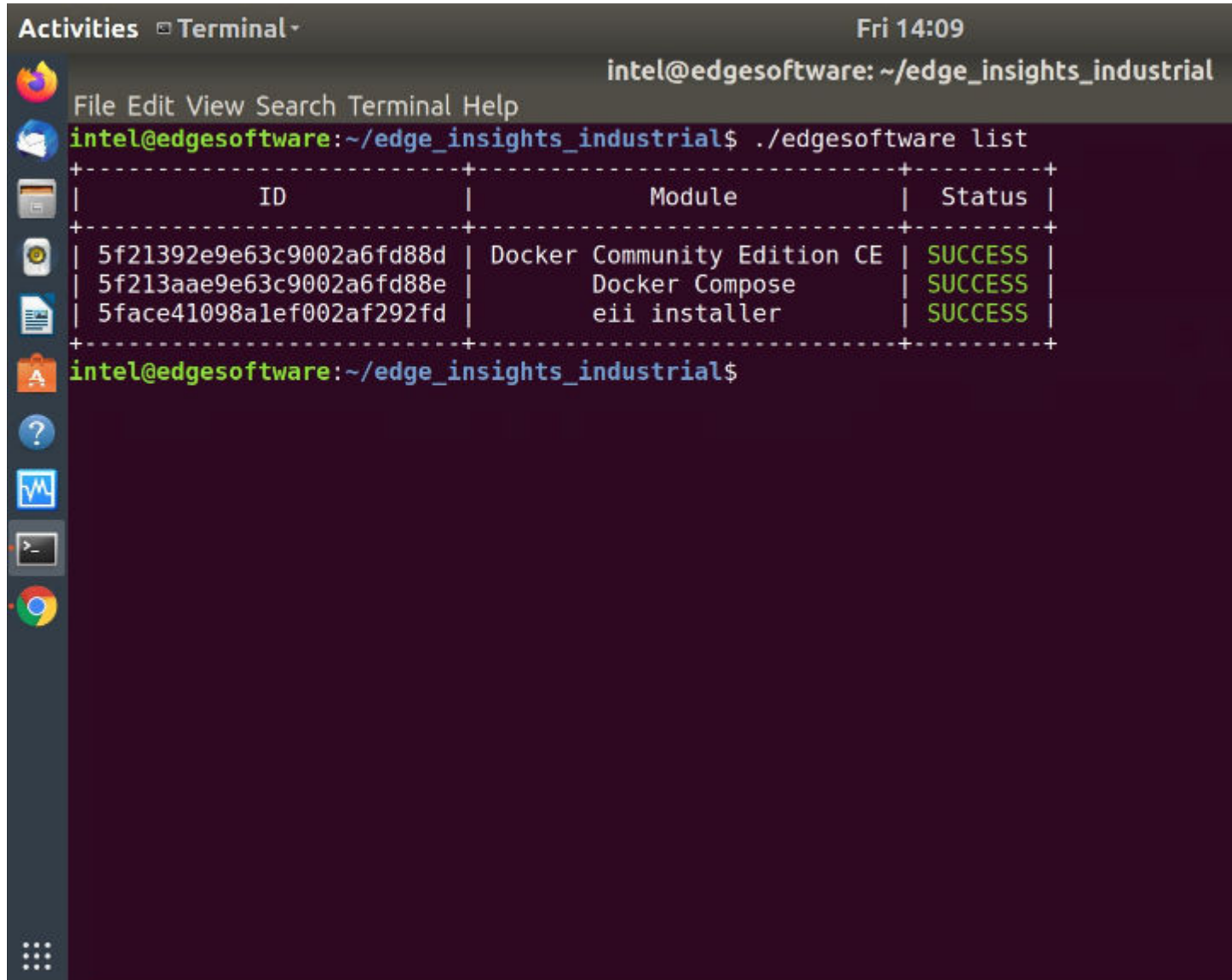
- Response: The edgesoftware version, build date, and target OS.

## List the Package Modules

- Command:

```
./edgesoftware list
```

- Response: The modules installed and status.



```
Activities Terminal Fri 14:09 intel@edgesoftware: ~/edge_insights_industrial
File Edit View Search Terminal Help
intel@edgesoftware:~/edge_insights_industrial$ ./edgesoftware list
+-----+-----+-----+
| ID | Module | Status |
+-----+-----+-----+
| 5f21392e9e63c9002a6fd88d | Docker Community Edition CE | SUCCESS |
| 5f213aae9e63c9002a6fd88e | Docker Compose | SUCCESS |
| 5face41098a1ef002af292fd | eii installer | SUCCESS |
+-----+-----+-----+
intel@edgesoftware:~/edge_insights_industrial$
```

## List Modules Available for Download

- Command:

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

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

## Display the CLI Event Log

- Command:

```
./edgesoftware log
```

- Response: CLI event log information, such as:

- target system information (hardware and software)
- system health
- installation status
- modules you can install

```

intel@edgesoftware: ~/edge_insights_industrial
File Edit View Search Terminal Help
intel@edgesoftware:~/edge_insights_industrial$ ./edgesoftware log
=====Start of installer log=====
Tue Aug 18 11:02:29 IST 2020 - INFO - ESB CLI version: 2020.2.08.17
Tue Aug 18 11:02:30 IST 2020 - INFO - -----SOFTWARE INFO-----
Tue Aug 18 11:02:30 IST 2020 - INFO - Operating System: Ubuntu 18.04.4 LTS
Tue Aug 18 11:02:30 IST 2020 - INFO - Kernel: 5.3.0-28-generic

Tue Aug 18 11:02:30 IST 2020 - INFO - -----HARDWARE INFO-----
Tue Aug 18 11:02:30 IST 2020 - INFO - Hardware Architecture: x86_64
Tue Aug 18 11:02:30 IST 2020 - INFO - Processor: Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz
Tue Aug 18 11:02:30 IST 2020 - INFO - Memory size: 15G
Tue Aug 18 11:02:30 IST 2020 - INFO - Total disk space (GB): 468
Tue Aug 18 11:02:30 IST 2020 - INFO - Free disk space (GB): 431

Tue Aug 18 11:02:30 IST 2020 - INFO - -----HARDWARE ACCELERATOR-----
Tue Aug 18 11:02:30 IST 2020 - INFO - No hardware accelerator found
Tue Aug 18 11:02:30 IST 2020 - INFO - Python version: 3.6.9
Tue Aug 18 11:02:30 IST 2020 - INFO - Checking Internet connection
Tue Aug 18 11:02:31 IST 2020 - INFO - Connected to the Internet
Tue Aug 18 11:02:31 IST 2020 - INFO - Checking for prerequisites
Tue Aug 18 11:02:34 IST 2020 - INFO - Installing prerequisites. This may take some time...
Tue Aug 18 11:04:39 IST 2020 - INFO - -----Successfully installed prerequisites-----
Tue Aug 18 11:04:39 IST 2020 - INFO - All dependencies met
Tue Aug 18 11:04:39 IST 2020 - INFO - Validating product key
Tue Aug 18 11:04:40 IST 2020 - INFO - Successfully validated Product Key
Tue Aug 18 11:04:40 IST 2020 - INFO - Starting installation
Tue Aug 18 11:04:40 IST 2020 - INFO - Downloading modules...
Tue Aug 18 11:04:40 IST 2020 - INFO - Downloading component esb_common
Tue Aug 18 11:04:40 IST 2020 - INFO - Sending request to download module 5e8c4742e02f17002a2a6976
Tue Aug 18 11:04:42 IST 2020 - INFO - Sending request to validate module 5e8c4742e02f17002a2a6976 with hash value
f25be6a391a4322e56ce2216fe4
Tue Aug 18 11:04:42 IST 2020 - INFO - Module validation passed for 5e8c4742e02f17002a2a6976
Tue Aug 18 11:04:42 IST 2020 - INFO - Successfully downloaded module esb_common
Tue Aug 18 11:04:42 IST 2020 - INFO - Downloading component Docker_Community_Edition_CE
Tue Aug 18 11:04:42 IST 2020 - INFO - Sending request to download module 5f21392e9e63c9002a6fd88d
Tue Aug 18 11:04:44 IST 2020 - INFO - Sending request to validate module 5f21392e9e63c9002a6fd88d with hash value
cb30328bcec83af377fe239d5f3
Tue Aug 18 11:04:45 IST 2020 - INFO - Module validation passed for 5f21392e9e63c9002a6fd88d
Tue Aug 18 11:04:45 IST 2020 - INFO - Successfully downloaded module Docker_Community_Edition_CE
Tue Aug 18 11:04:45 IST 2020 - INFO - Downloading component Docker_Compose
Tue Aug 18 11:04:45 IST 2020 - INFO - Sending request to download module 5f21392e9e63c9002a6fd88d

```

### 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.

## 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 Industrial 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 Industrial download.

---

### NOTE

#### Important

Do not manually edit `edgesoftware_configuration.xml`.

---

1. Open a terminal window.
2. Go to the `edge_insights_industrial/` 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 Industrial download.

---

**NOTE** To find the module ID, use

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

---

## 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:

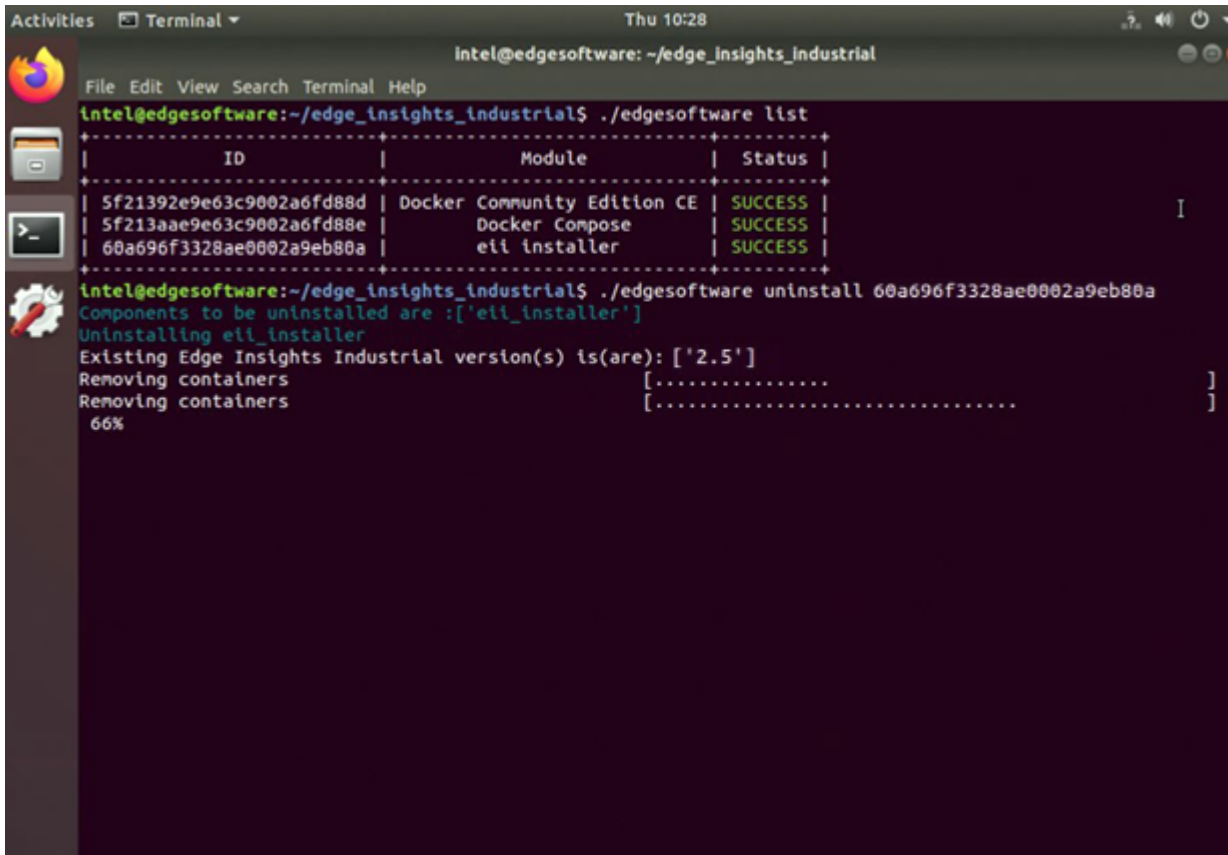
```
./edgesoftware export
```

## Uninstall the Packages

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

To uninstall individual package, run the following command:

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

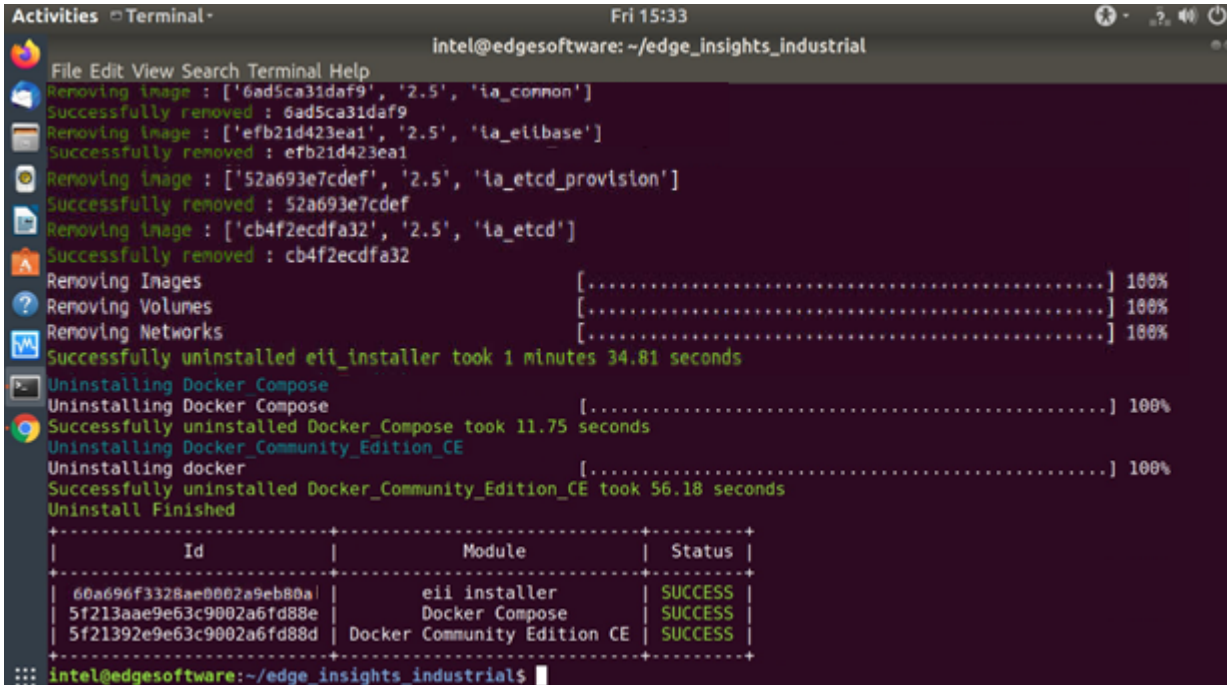


The screenshot shows a terminal window with the following output:

```
intel@edgesoftware: ~/edge_insights_industrial
File Edit View Search Terminal Help
intel@edgesoftware:~/edge_insights_industrial$ ./edgesoftware list
+-----+-----+-----+
| ID | Module | Status |
+-----+-----+-----+
| 5f21392e9e63c9002a6fd88d | Docker Community Edition CE | SUCCESS |
| 5f213aae9e63c9002a6fd88e | Docker Compose | SUCCESS |
| 60a696f3328ae0002a9eb80a | eii installer | SUCCESS |
+-----+-----+-----+
intel@edgesoftware:~/edge_insights_industrial$ ./edgesoftware uninstall 60a696f3328ae0002a9eb80a
Components to be uninstalled are :['eii_installer']
Uninstalling eii_installer
Existing Edge Insights Industrial version(s) is(are): ['2.5']
Removing containers [.....]
Removing containers [.....]
66%
```

To uninstall all the packages, run the following command:

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



```
Activities Terminal Fri 15:33 intel@edgesoftware: ~/edge_insights_industrial
File Edit View Search Terminal Help
Removing image : ['6ad5ca31daf9', '2.5', 'ia_connon']
Successfully removed : 6ad5ca31daf9
Removing image : ['efb21d423ea1', '2.5', 'ia_etibase']
Successfully removed : efb21d423ea1
Removing image : ['52a693e7cdef', '2.5', 'ia_etcd_provision']
Successfully removed : 52a693e7cdef
Removing image : ['cb4f2ecd3a32', '2.5', 'ia_etcd']
Successfully removed : cb4f2ecd3a32
Removing Images [.....] 100%
Removing Volumes [.....] 100%
Removing Networks [.....] 100%
Successfully uninstalled eii_installer took 1 minutes 34.81 seconds
Uninstalling Docker Compose [.....] 100%
Successfully uninstalled Docker Compose took 11.75 seconds
Uninstalling Docker Community Edition CE [.....] 100%
Successfully uninstalled Docker Community Edition CE took 56.18 seconds
Uninstall Finished
+-----+-----+-----+
| Id | Module | Status |
+-----+-----+-----+
| 60a696f3328ae0002a9eb80a | eii installer | SUCCESS |
| 5f213aae9e63c9002a6fd88e | Docker Compose | SUCCESS |
| 5f21392e9e63c9002a6fd88d | Docker Community Edition CE | SUCCESS |
+-----+-----+-----+
```

## Summary and Next Steps

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

To start using Edge Insights for Industrial, go to [Tutorials](#).

## Tutorials

---

Follow the tutorial in this section to learn how to use and configure Edge Insights for Industrial for the Defect Detection Demo.

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

Get started in your learning journey:

- [Defect Detection Demo](#)

---

## Defect Detection Demo

**Prerequisite:** The Defect Detection Demo tutorial requires video pipeline services, which are available in the **Video Analytics** or **Video Analytics and Time series** use cases installer package downloaded through the **Download** option for the [Edge Insights for Industrial](#) package.

In this tutorial, you'll run the Defect Detection Demo to verify that Edge Insights for Industrial was installed successfully and to start getting familiar with its modules and structure. By following this tutorial, you will learn:

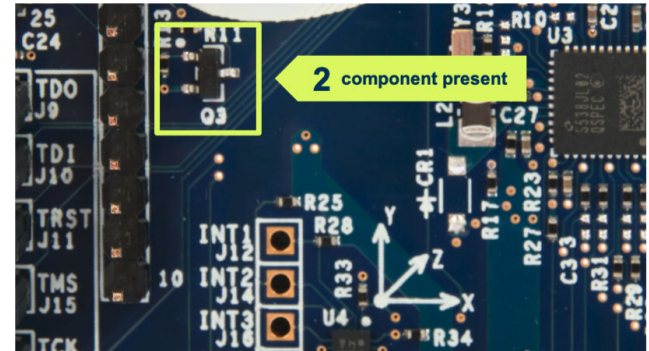
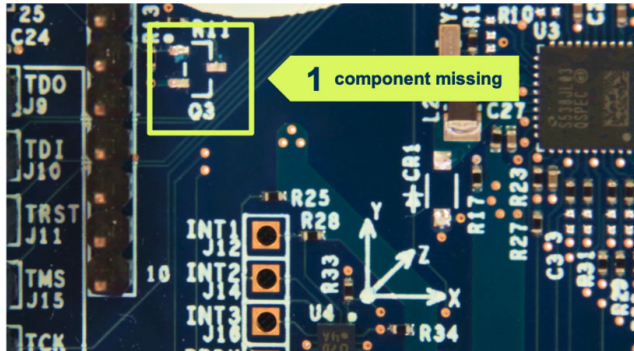
- How to start Edge Insights for Industrial
- How to visualize the results of the demo application
- How the application works at a high level

**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.

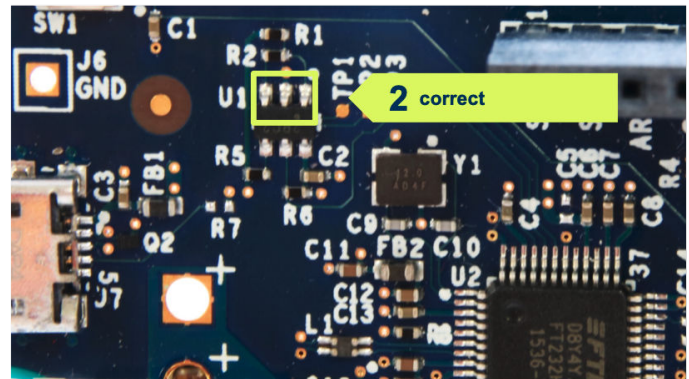
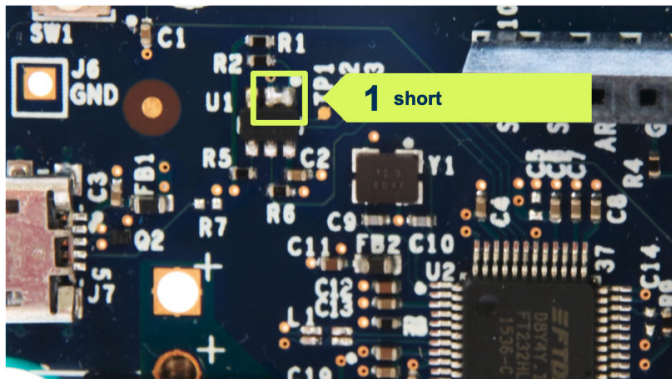
## How it Works

The Defect Detection Demo performs a quality control inspection on a video of the printed circuit boards (PCBs). In this scenario, the PCBs are inspected for quality control and detects two types of defects:

- A missing component. In the following illustration, PCB #1, on the left, is missing a component. PCB #2, on the right, shows the component in place:



- A component short: In the following illustration, PCB #1, on the left, has two solder joints connected that should not be connected, resulting in a short. PCB #2, on the right, shows the correct solder:



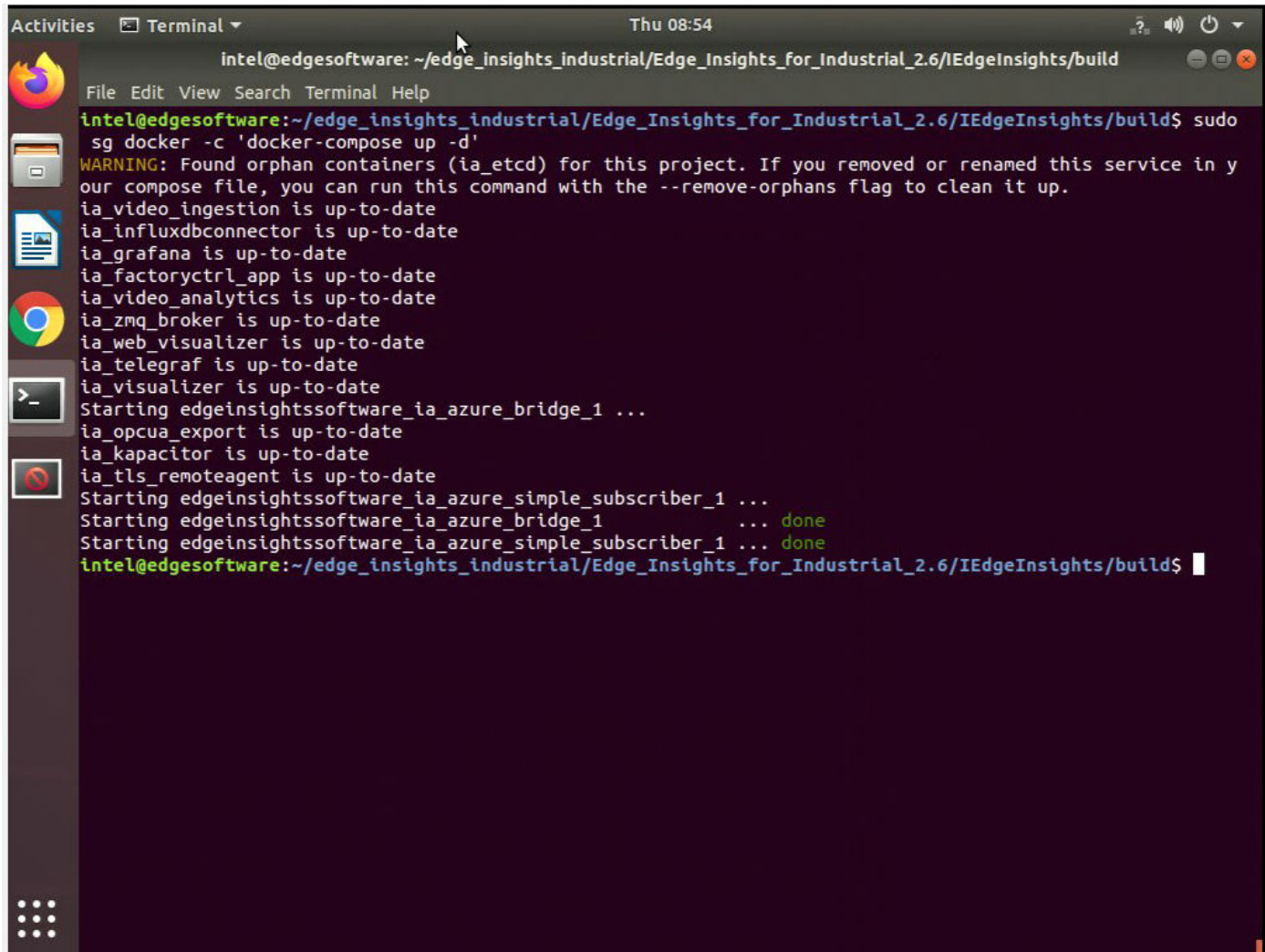
## Step 1: Start the Edge Insights for Industrial Containers

1. Run the Edge Insights for Industrial:

```
xhost +
cd $HOME/edge_insights_industrial/Edge_Insights_for_Industrial_<version>/IEdgeInsights/build
sudo ./eii_start.sh
```

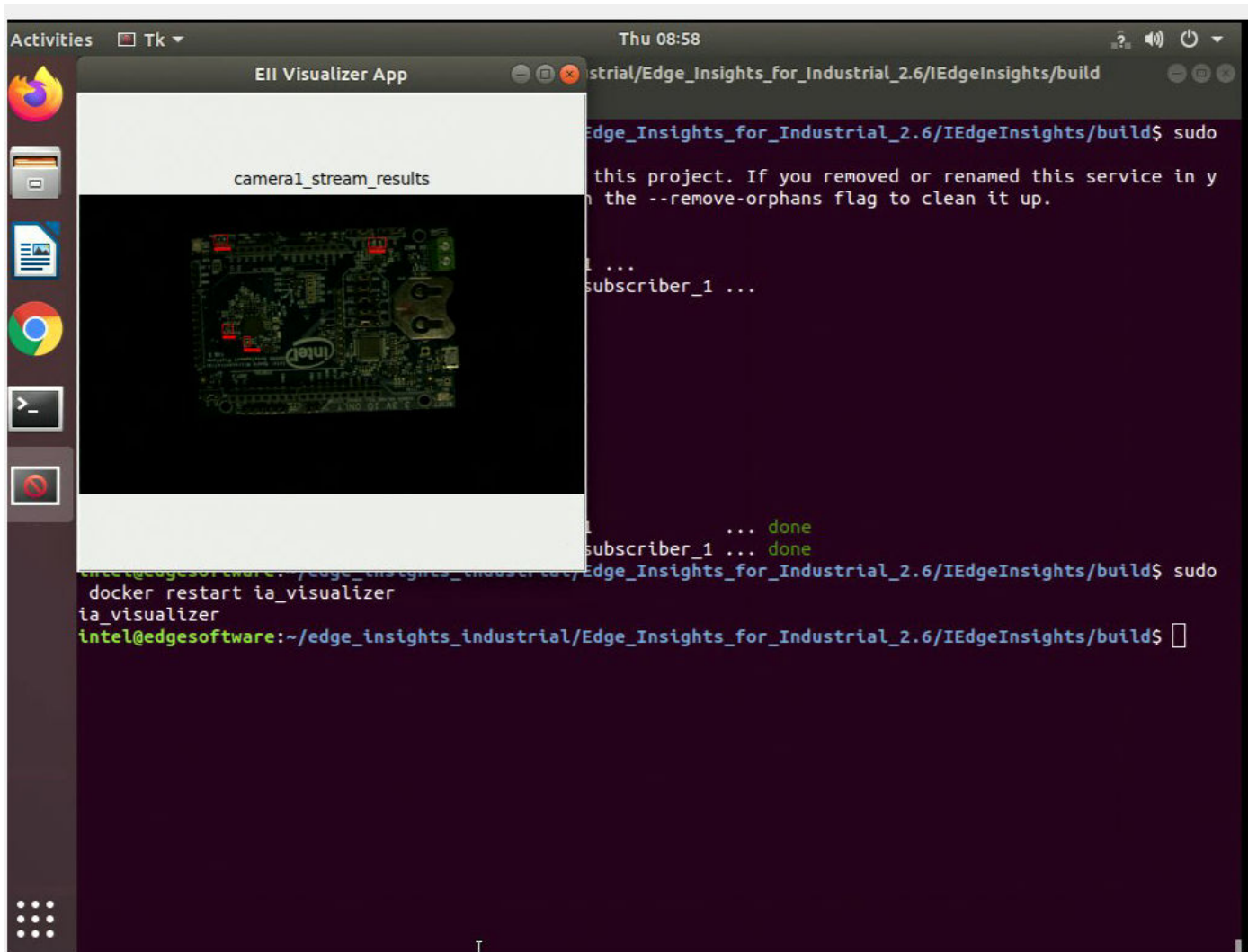
where <version> indicates the downloaded version of Edge Insights for Industrial.

Success is indicated by a screen similar to:



```
intel@edgesoftware: ~/edge_insights_industrial/Edge_Insights_for_Industrial_2.6/IEdgeInsights/build
File Edit View Search Terminal Help
intel@edgesoftware:~/edge_insights_industrial/Edge_Insights_for_Industrial_2.6/IEdgeInsights/build$ sudo
sg docker -c 'docker-compose up -d'
WARNING: Found orphan containers (ia_etcd) for this project. If you removed or renamed this service in y
our compose file, you can run this command with the --remove-orphans flag to clean it up.
ia_video_ingestion is up-to-date
ia_influxdbconnector is up-to-date
ia_grafana is up-to-date
ia_factoryctrl_app is up-to-date
ia_video_analytics is up-to-date
ia_zmq_broker is up-to-date
ia_web_visualizer is up-to-date
ia_telegraf is up-to-date
ia_visualizer is up-to-date
Starting edgeinsightssoftware_ia_azure_bridge_1 ...
ia_opcua_export is up-to-date
ia_kapacitor is up-to-date
ia_tls_remoteagent is up-to-date
Starting edgeinsightssoftware_ia_azure_simple_subscriber_1 ...
Starting edgeinsightssoftware_ia_azure_bridge_1 ... done
Starting edgeinsightssoftware_ia_azure_simple_subscriber_1 ... done
intel@edgesoftware:~/edge_insights_industrial/Edge_Insights_for_Industrial_2.6/IEdgeInsights/build$
```

In addition, the visualizer displays an image that displays metadata and a separate window that shows the PCB image with the defects outlined by red boxes:



**NOTE** If the Visualizer UI does not show up and you notice the error `couldn't connect to display ":0"` after running the command `docker logs -f ia_visualizer`, check the value of the `DISPLAY` environment variable on the host machine. Use the command:

```
env | grep DISPLAY
```

Set the value for the `DISPLAY` environment variable in the `ia_visualizer` service in the `HOME/edge_insights_industrial/Edge_Insights_for_Industrial_<version>/IEdgeInsights/build/docker-compose.yml` file. Then bring up the `ia_visualizer` UI with the command:

```
docker-compose up ia_visualizer
```

For example,

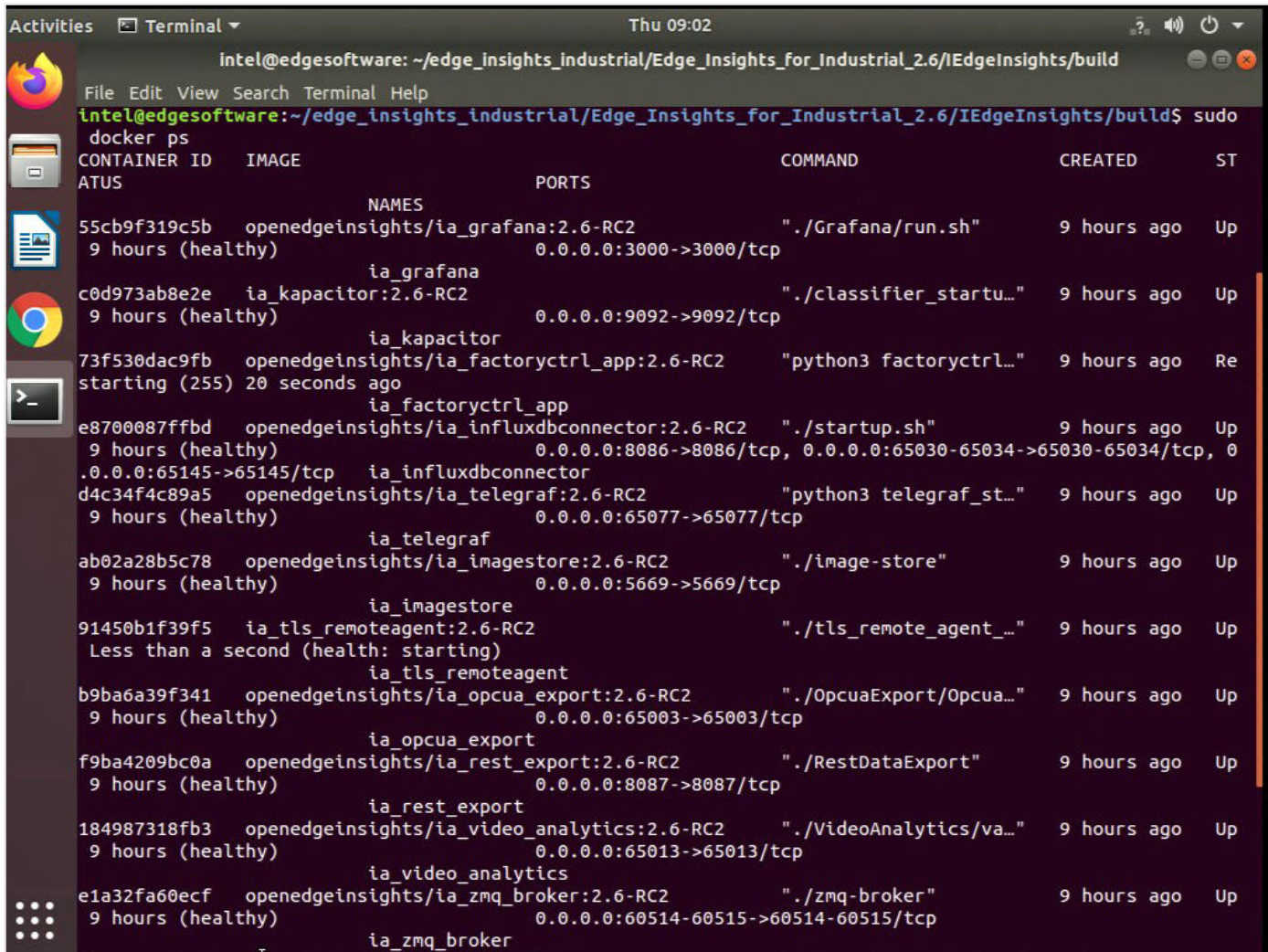
```
$ env | grep DISPLAY
DISPLAY=:1
```

Set `=:1` as the `DISPLAY` environment value in the `ia_visualizer` service in the `docker-compose.yml` file.

## 2. Verify the containers are running:

```
docker ps
```

The result looks similar to:



The following table describes the contents of the screen.

Note that <version> indicates the downloaded version of Edge Insights for Industrial.

Image Column Content	Description
ia_video_ingestion:<version>	Ingests video frames from a video source, like a video file or camera, using the GStreamer* pipeline. Data, consisting of a frame and metadata, is published to the message bus.
ia_video_analytics:<version>	Use OpenVINO™ on the data to perform inference. The data are received from the video ingestion and new data is published to the message bus.
ia_visualizer:<version>	Use a Python-based visualizer to display the frame sent by video analytics.

Image Column Content	Description
ia_etcd:<version>	etcd* provides endpoint configurations to establish the message bus and configuration of Edge Insights for Industrial containers.
ia_etcd_ui:<version>	Web user interface for etcd* configurations.

3. Check the log files to verify the data pipeline in Edge Insights for Industrial is working correctly:

```
sudo docker logs -f ia_video_analytics
```

```

intel@edgesoftware: ~/edge_insights_industrial/Edge_Insights_for_Industrial_2.6/EdgeInsights/build
File Edit View Search Terminal Help
intel@edgesoftware:~/edge_insights_industrial/Edge_Insights_for_Industrial_2.6/EdgeInsights/build
docker logs -f ia_video_analytics
[setupvars.sh] OpenVINO environment initialized
(root): extra property 'queue_size' found.
[Wed Jun 23 19:01:19 2021] INFO:validate_json:49: JSON schema validation passed !
[Wed Jun 23 19:01:19 2021] WARN:get_config_value:202: JSON does not contain key: zmq_recv_hwm
[Wed Jun 23 19:01:19 2021] WARN:get_config_value:202: JSON does not contain key: brokered
[Wed Jun 23 19:01:19 2021] WARN:get_config_value:202: JSON does not contain key: BrokerAppName
[Wed Jun 23 19:01:19 2021] WARN:get_config_value:202: JSON does not contain key: zmq_recv_hwm
[Wed Jun 23 19:01:19 2021] WARN:get_config_value:202: JSON does not contain key: zmq_connect_r
[Wed Jun 23 19:01:19 2021] WARN:get_config_value:202: JSON does not contain key: brokered
[Wed Jun 23 19:01:19 2021] INFO:UdfManager:105: max_workers: 4
[Wed Jun 23 19:01:19 2021] INFO:run:228: UDFManager thread started
[Wed Jun 23 19:01:19 2021] INFO:run:228: UDFManager thread started
[Wed Jun 23 19:01:19 2021] INFO:run:228: UDFManager thread started
[Wed Jun 23 19:01:19 2021] INFO:run:228: UDFManager thread started
[Wed Jun 23 19:01:30 2021] WARN:get_config_value:202: JSON does not contain key: zmq_connect_r
[Wed Jun 23 19:01:30 2021] INFO:start:184: Publisher thread started...
[Wed Jun 23 19:01:30 2021] INFO:start:189: Started udf manager
[Wed Jun 23 19:01:30 2021] INFO:start:194: Subscriber thread started...

```

1. Press Ctrl+C on your keyboard to stop the log file.

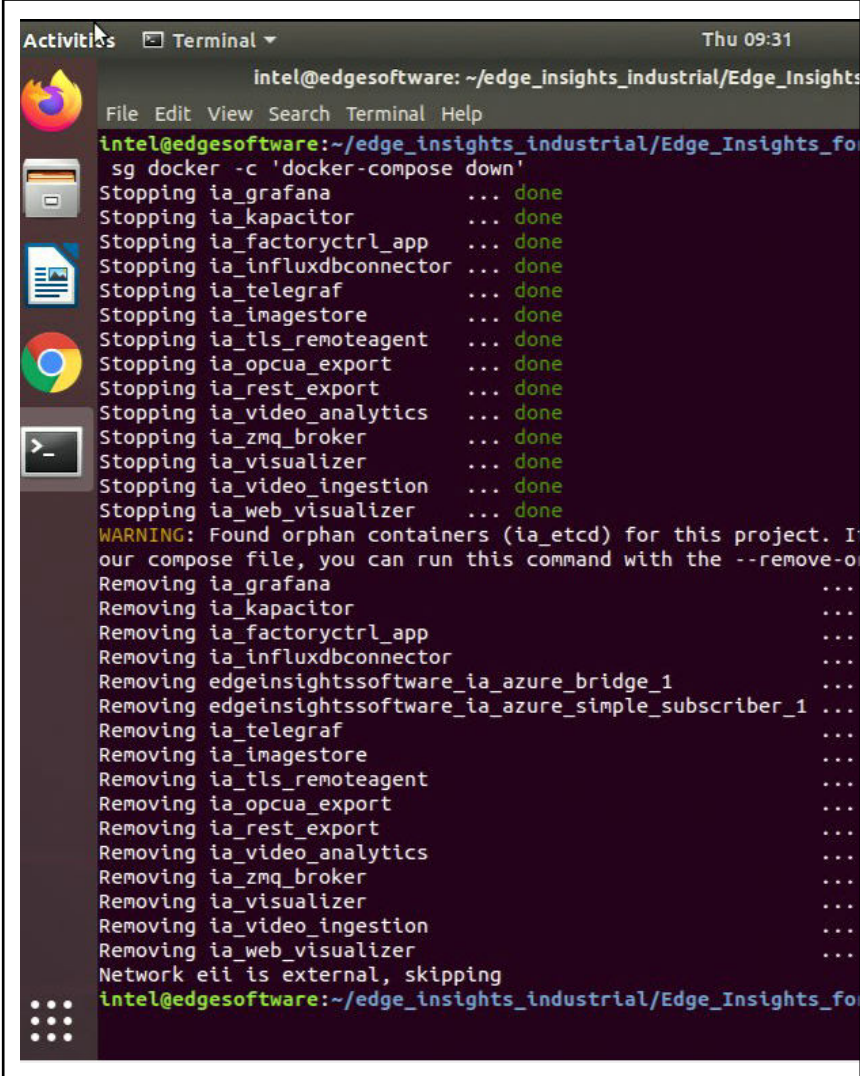
**NOTE** You must stop the containers to close the visualizer.

## Step 2: Stop the Edge Insights for Industrial Containers

1. Stop the Edge Insights for Industrial containers:

```
cd $HOME/edge_insights_industrial/Edge_Insights_for_Industrial_<version>/IEdgeInsights/build
sudo sg docker -c `docker-compose down`
```

where <version> indicates the downloaded version of Edge Insights for Industrial.



```
intel@edgesoftware: ~/edge_insights_industrial/Edge_Insights_for_Industrial_2.6/IEdgeInsights/build
File Edit View Search Terminal Help
intel@edgesoftware:~/edge_insights_industrial/Edge_Insights_for_Industrial_2.6/IEdgeInsights/bu
sg docker -c 'docker-compose down'
Stopping ia_grafana ... done
Stopping ia_kapacitor ... done
Stopping ia_factoryctrl_app ... done
Stopping ia_influxdbconnector ... done
Stopping ia_telegraf ... done
Stopping ia_imagestore ... done
Stopping ia_tls_remoteagent ... done
Stopping ia_opcua_export ... done
Stopping ia_rest_export ... done
Stopping ia_video_analytics ... done
Stopping ia_zmq_broker ... done
Stopping ia_visualizer ... done
Stopping ia_video_ingestion ... done
Stopping ia_web_visualizer ... done
WARNING: Found orphan containers (ia_etcd) for this project. If you removed or renamed this ser
our compose file, you can run this command with the --remove-orphans flag to clean it up.
Removing ia_grafana ... done
Removing ia_kapacitor ... done
Removing ia_factoryctrl_app ... done
Removing ia_influxdbconnector ... done
Removing edgeinsightssoftware_ia_azure_bridge_1 ... done
Removing edgeinsightssoftware_ia_azure_simple_subscriber_1 ... done
Removing ia_telegraf ... done
Removing ia_imagestore ... done
Removing ia_tls_remoteagent ... done
Removing ia_opcua_export ... done
Removing ia_rest_export ... done
Removing ia_video_analytics ... done
Removing ia_zmq_broker ... done
Removing ia_visualizer ... done
Removing ia_video_ingestion ... done
Removing ia_web_visualizer ... done
Network eii is external, skipping
intel@edgesoftware:~/edge_insights_industrial/Edge_Insights_for_Industrial_2.6/IEdgeInsights/bu
```

### How the PCB Defect Detection Demo Works

Video frames are sent to a Python\* application named **filter** in the Video Ingestion container where unwanted frames are filtered out and frames of interest are passed to a Python application named **classifier**. The classifier application is also in the Video Analytics container. This container is for deep learning inference via the data bus.

Results are saved to a database and passed to the data bus, which is used by the Visualizer container that runs a sample Python application to display the images on the target system.



## Summary and Next Steps

In this tutorial, you learned to start Edge Insights for Industrial and see the results of the demo application as it performed a quality control inspection on a video of PCBs. You also learned how the application works at a high level and are more familiar with its modules and structure. For more Edge Insights for Industrial features/configurations, refer to `edge_insights_industrial/Edge_Insights_for_Industrial_<version>/IEdgeInsights/README.md`, where `<version>` indicates the downloaded version of Edge Insights for Industrial.

## Release Notes

---

### Edgesoftware CLI Features

Initial features for recommended configuration include:

- Installs Docker CE\*
- Installs Docker Compose\*
- Installs all the prerequisites and dependencies for Edge Insights for Industrial.
- Installs and sets up the Edge Insights for Industrial.
- Provisions Edge Insights for Industrial based on user input 1) Production or 2) Developer mode.
- Supports below use cases:
  - Video Analytics + Time Series
  - Video Analytics
  - Time Series
- Brings up all container images as per the use case selected during Edge Insights for Industrial installation.
- Downloads Manageability module, which can be installed manually with TPM and non-TPM capability.
- Custom flow enabled users to select or deselect different modules as per their choice.

### EII v3.0.1

EII v3.0.1 is a bugfix release on top of EII v3.0 release.

#### Fixes

- General security fixes have been applied. Use the EII v3.0.1 docker images available at [Docker Hub](#).
- Added the prerequisite step in README.md of having the openssh-server installed for the Web Deployment Tool backend service
- Fixed OpcuaExport docker build issue (fixed the mbedtls path issue in the Dockerfile). To avoid this issue, use the `openedgeinsights/ia_opcua_export:3.0` from the Docker Hub.
- Fixed Kapacitor launch issue by updating the `conda_requirements.txt`. To avoid this issue, use the `openedgeinsights/ia_kapacitor:3.0` from the Docker Hub.

#### Known Issues

- While bringing up the **ia\_azure\_bridge** service, protobuf related issue can occur. In this scenario, add `protobuf==3.20.2` in the `[WORKDIR]/IEdgeInsights/AzureBridge/modules/AzureBridge/requirements.txt` file and rebuild the `ia_azure_bridge` docker image. To avoid this issue, use the `openedgeinsights/ia_azure_bridge:3.0` from the Docker Hub.

- While executing on the Alder Lake 12th Gen Intel(R) Core Processors, the VideoIngestion service can crash with the Gstreamer VA-API element error. In this scenario, to install the required GPU drivers, make the following changes from line 155 through 157 in the [WORKDIR]/IEdgeInsights/VideoIngestion/Dockerfile and rebuild the ia\_video\_ingestion image:

```
# Installing Intel® Graphics Compute Runtime for OpenCL™
# RUN /bin/bash -c "source /opt/intel/opencvino/bin/setupvars.sh && \
#         cd /opt/intel/opencvino/install_dependencies && \
#         yes | ./install_NEO_OCL_driver.sh --auto || true"
RUN apt update && apt-get install -y gpg-agent wget
RUN wget -qO - https://repositories.intel.com/graphics/intel-graphics.key | gpg --dearmor --
output /usr/share/keyrings/intel-graphics.gpg
RUN echo 'deb [arch=amd64 signed-by=/usr/share/keyrings/intel-graphics.gpg] https://
repositories.intel.com/graphics/ubuntu focal-legacy main' | \
tee /etc/apt/sources.list.d/intel.gpu.focal.list
RUN apt-get update && apt-get install \
yes | intel-opencl-icd --auto || true \
intel-level-zero-gpu level-zero \
intel-media-va-driver-non-free libmfx1 libmfxgen1 libvpl2
```

## EII 3.0 Changes

The details of the changes for EII 3.0 is available [here \(Open Edge Insights for Industrial v3.0\)](#).

---

**NOTE** The Training and Learning Suite (TLS) is not supported in EII v3.0.

---

## Release Content

Subproject (component)	Location	Revision
IEdgeInsights	edge_insights_industrial/ Edge_Insights_for_Industria l_3.0/IEdgeInsights	v3.0
Manageability	edge_insights_industrial/ Edge_Insights_for_Industria l_3.0/manageability	v3.0

## Documentation Archive

---

View earlier version of Edge Insights for Industrial documentation:

- [2.6.3](#)

---

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

---