

# Getting started with thin-edge.io on a RevPi from Kunbus

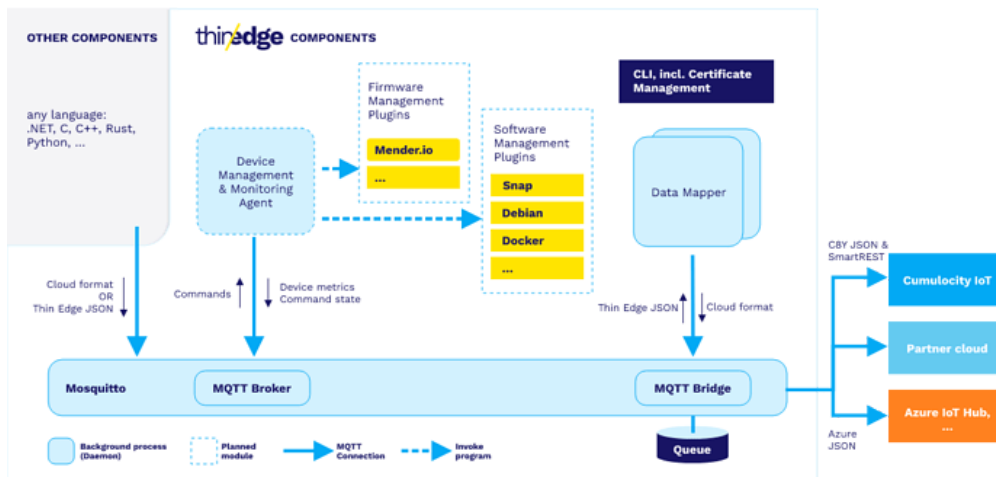
[newsletter](#), [Cumulocity-IoT](#), [IoT-Data-Analytics](#), [IoT](#), [IoT-Device-Connectivity](#), [IoT-Edge](#), [thin-edge.io](#)

[Murat\\_Bayram](#) (Murat Bayram) 1 June 21, 2022, 6:24pm

## Introduction

In this post the installation for [thin-edge.io](#) on a RevPi from Kunbus is described.

[thin-edge.io](#) is an open-source project initiated by Software AG and partners to provide a cloud-agnostic edge framework. It is much more generic than the device management agent, so it can connect to multiple IoT Platforms, and it allows flexible logic executed on the device. It is optimized for a very small footprint and high performance.



The [Revolution PI](#) from Kunbus is an open, modular and cost-effective industrial PC based on the well-known Raspberry Pi. Housed in a slim DIN rail case, the three available base modules can be seamlessly expanded with a variety of matching I/O modules and fieldbus gateways.



Kunbus already got their Cumulocity device management certification and is listed as partner within the [Device Catalog](#) of certified device.

You can get more information about Kunbus and their RevPi on their [website](#).

## Prerequisite

To follow this guide, you only need the following:

- Kunbus RevPI connected
- A Cumulocity [Trial tenant](#)
- Make sure Mosquitto MQTT Broker is installed in Version 2.x on the RevPI

## Install [thin-edge.io](#)

Open a shell of your choice and ssh into the RevPI. The credentials are on the label on the side.

Before installing run

```
sudo apt-get update
```

There are two possibilities to install [thin-edge.io](#), the easiest way is to use the installation script with this command:

```
curl -fsSL https://raw.githubusercontent.com/thin-edge/thin-edge.io/main/get-thir
```

The script will install [thin-edge.io](#) in the latest version. If you want to manually install [thin-edge.io](#) you can also do this via the binaries. See the [documentation](#) for more information.

If successful you can now use [thin-edge.io](#) via tedge commands.

## tege CLI

[thin-edge.io](#) comes with a powerful command line interface

```
tedge 0.7.0
tedge is the cli tool for thin-edge.io

USAGE:
  tedge [OPTIONS] [SUBCOMMAND]

OPTIONS:
  --config-dir <CONFIG_DIR> [default: /etc/tedge]
  -h, --help                  Print help information
  --init                      Initialize the tedge
  -V, --version               Print version information

SUBCOMMANDS:
  cert      Create and manage device certificate
  config   Configure Thin Edge
  connect   Connect to connector provider
  disconnect Remove bridge connection for a provider
  help     Print this message or the help of the given subcommand(s)
  mqtt     Publish a message on a topic and subscribe a topic
```

The usage is via:

```
tedge [OPTIONS] [SUBCOMMAND]
```

and -h can be used to see the help for the latest subcommand.

The CLI will be used to configure the [thin-edge.io](#) installation on the RevPI in the next steps.

## Configure and Connect

In order to connect the RevPi to a Cumulocity tenant it need to be configured.

The following configuration parameters are mandatory

### 1. C8Y URL

The tenant url is needed in order to allow the upload of the certificate to the specific tenant and the registration of the device. In can be configured via:

```
sudo tedge config set c8y.url {{YOUR_C8Y_URL}}
```

### 2. Certificate

[thin-edge.io](#) connects via MQTT protocol using a X.509 certificate for authentication. To do so, a certificate must be trusted by Cumulocity. A certificate is trusted when it is added to the trusted certificates and is in activated state.

First we need to create the device certificate locally (If you already have a device certificate uploaded directly via ui to Cumulocity you can skip that step).

```
sudo tedge cert create --device-id {{YOUR_UNIQUE_DEVICE_ID}}
```

The device id is a unique identifier e.g. MAC address that identifies a physical device clearly.

The certificate is uploaded to the Cumulocity Tenant via:

```
sudo tedge cert upload c8y --user {{YOUR_USERNAME}}
```

If the password prompt appears, enter your password.

### 3. Connect

We now are ready to connect RevPi with Cumulocity. This can be achieved via:

```
sudo tedge connect c8y
```

After the connect the device will be created on the platform side and can be found within the device list in the device management.

## Sending Device Data

Once your Thin Edge device is configured and connected to an IoT cloud provider, you can start sending measurements, events or alarms. In the standard configuration you can not connect externally to the mosquito broker and thus the messages have to be send directly from the device itself. If you want to change that you need to configure according to [here](#).

The tedge cli allows you to send payloads via mqtt the following way:

```
tedge mqtt pub {{TOPIC}} {{PAYLOAD}}
```

**Thin-edge.io** comes with a tedge-mapper daemon. This process collects the data von the 'tedge/#' topics and translates them to the tedge payloads on the 'c8y/#' topics which is mapped directly to Cumulocity. The mapper thus translates simple json's to the desired target payload for Cumulocity.

## Sending measurements

**Measurements** within Cumulocity represent regularly acquired readings and statistics from sensors.

A simple single-valued measurement like a temperature measurement, can be represented in Thin Edge JSON as follows:

```
{ "temperature": 25 }
```

with the key-value pair representing the measurement type and the numeric value of the measurement. The endpoint that is supervised by the tedge-mapper for measurements is:

```
tedge/measurements
```

Thus the temperature measurement described above can be sent as follows:

```
tedge mqtt pub tedge/measurements '{ "temperature": 25 }'
```

## Sending events

**Events** are used to pass real-time information through Cumulocity IoT that are not just plain sensor values.

A simple event can be represented in Thin Edge JSON as follows:

```
{  
  "text": "A door was closed",  
  "time": "2022-06-10T05:30:45+00:00"  
}
```

The endpoint that is supervised by the tedge-mapper for events is:

```
tedge/events/{event-type}
```

Thus the door open event described above can be sent as follows:

```
tedge mqtt pub tedge/events/door '{"text": "A door was closed","time": "2022-06-10T05:30:45+00:00"}'
```

| A door was closed    |                            |
|----------------------|----------------------------|
| Time                 | 10 Jun 2022 07:30:45       |
| Server creation time | 20 Jun 2022 17:21:44       |
| Type                 | door                       |
| Last updated         | '2022-06-20T15:21:44.578Z' |

## Device Monitoring

With [thin-edge.io](#) device monitoring, you can collect metrics from the RevPi and forward these device metrics to Cumulocity.

[Thin-edge.io](#) uses the open source component [collectd](#) to collect the metrics from the device. [Thin-edge.io](#) translates the collected metrics from their native format to the [thin-edge.io JSON](#) format and then into the [cloud-vendor specific format](#).

Enabling monitoring on your device is a 3-steps process:

1. Install collectd
2. Configure collectd
3. Enable [thin-edge.io](#) monitoring

## Install collectd

Since [thin-edge.io](#) uses the MQTT plugin of collectd, one needs to install the mosquitto client library (either libmosquitto1 or mosquitto-clients).

```
sudo apt-get install libmosquitto1
```

To install collectd:

```
sudo apt-get install collectd-core
```

## Configure collectd

[Thin-edge.io](#) provides a [basic collectd configuration](#) that can be used to collect cpu, memory and disk metrics.

Simply copy that file to the main collectd configuration file and restart the daemon.

```
sudo cp /etc/tedge/contrib/collectd/collectd.conf /etc/collectd/collectd.conf
sudo systemctl restart collectd
```

What you should see by now is that data arrives on the 'collectd/#' topics. You can check that via:

```
tedge mqtt sub 'collectd/#'
```

```
[ubuntu@first-jackrabbit:~]$ tedge mqtt sub 'collectd/#'
INFO: Connected
[collectd/first-jackrabbit/cpu/percent-active] 1655731822.113:12.3595505617978
[collectd/first-jackrabbit/memory/percent-used] 1655731822.114:11.2657845447405
[collectd/first-jackrabbit/cpu/percent-active] 1655731823.147:0.961538461538462
[collectd/first-jackrabbit/memory/percent-used] 1655731823.147:11.2649789539403
```

## Enable Collectd

To enable monitoring on your device, you have to launch the tedge-mapper-collectd daemon process. This process collects the data von the 'collectd/#' topics and translates them to the tedge payloads on the 'c8y/#' topics.

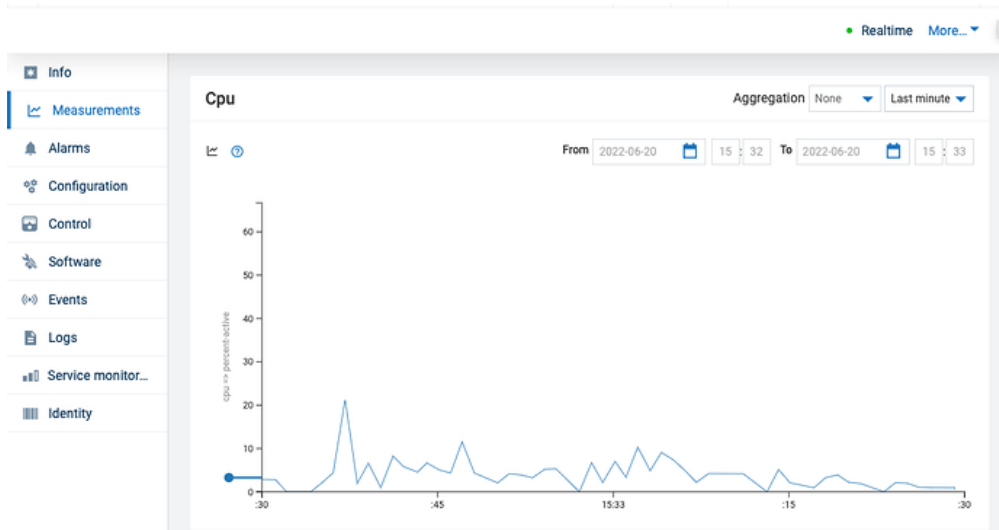
```
sudo systemctl start tedge-mapper-collectd
sudo systemctl enable tedge-mapper-collectd
```

You can inspect the collected and translated metrics, by subscribing to these topics:

```
tedge mqtt sub 'c8y/#'
```

```
ubuntu@first-jackrabbit:~$ tedge mqtt sub 'c8y/#'
INFO: Connected
[c8y/measurement/measurements/create] {"type":"ThinEdgeMeasurement","time":"2022-06-20T13:32:12.148999292Z","memory":{"percent-used":{"value":11.3016333353473},"cpu":{"percent-active":{"value":2.88461538461538}}}
[c8y/measurement/measurements/create] {"type":"ThinEdgeMeasurement","time":"2022-06-20T13:32:13.086999893Z","memory":{"percent-used":{"value":11.3080780617485},"cpu":{"percent-active":{"value":2.1978021978022}}}
[c8y/measurement/measurements/create] {"type":"ThinEdgeMeasurement","time":"2022-06-20T13:32:14.084000117Z","cpu":{"percent-active":{"value":2.53164556962025},"memory":{"percent-used":{"value":11.3048556985479}}}
```

The monitoring data will appear on the device in the measurement section.



## Software management

The [software management](#) takes care of allowing installation and management of any type of software. Since type is generic, any type of software can be managed. In [thin-edge.io](#) this can be extended with plugins. For every software type, a particular plugin is needed.

The following plugins do exist:

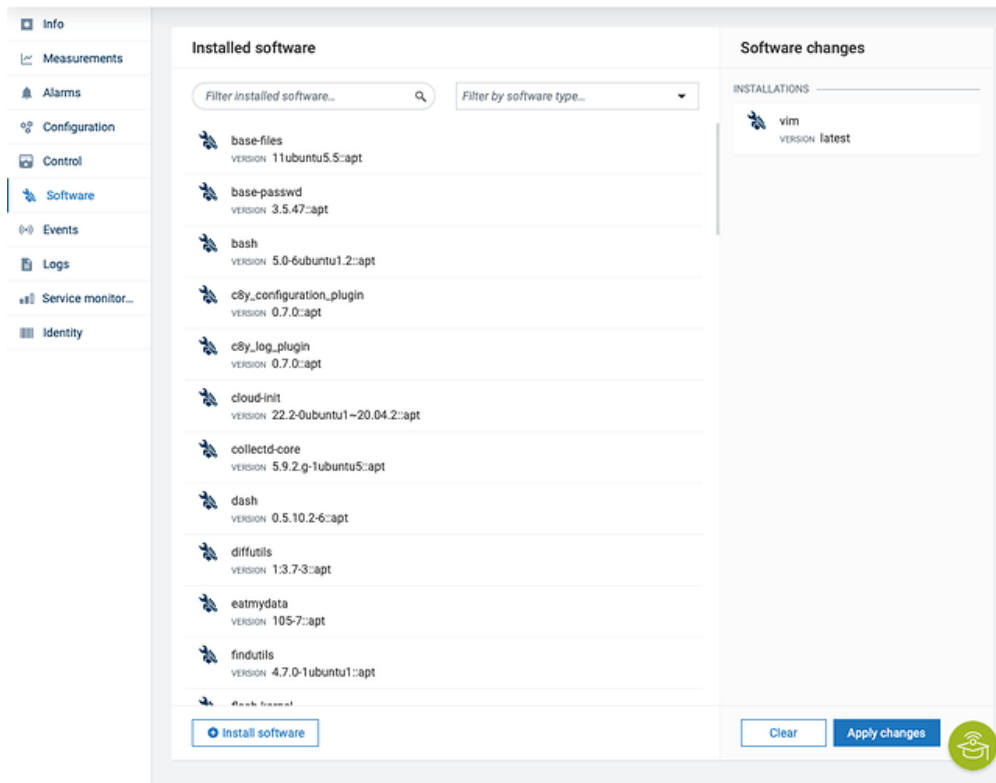
- docker
- apt
- docker-compose
- snap

In order to use those plugins they need to be copied to:

```
/etc/tedge/sm-plugins
```

The apt plugin is installed automatically. You can find the other plugins in the [repository](#). Make sure to disconnect/reconnect the device after adding plugins via

```
sudo tedge disconnect c8y
sudo tedge connect c8y
```



How to develop your own plugins is described [here](#).

## Configuration

With [thin-edge.io](#) you can manage config files on a device by using the [Cumulocity configuration management feature](#) as a part of Device Management.

This functionality is directly installed with the initial script. However, you need to configure the `/etc/tedge/c8y/c8y-configuration-plugin.toml` and add the entries for the configuration files that you want to manage. Just copy the following content to that file:

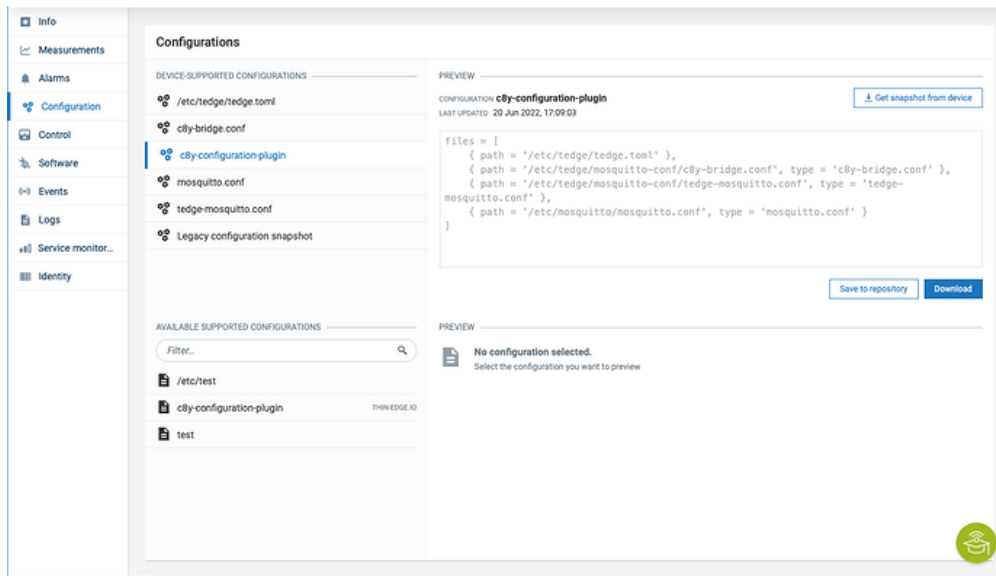
```
files = [
  { path = '/etc/tedge/tedge.toml' },
  { path = '/etc/tedge/mosquitto-conf/c8y-bridge.conf', type = 'c8y-bridge.conf' },
  { path = '/etc/tedge/mosquitto-conf/tedge-mosquitto.conf', type = 'tedge-mosquitto.conf' },
  { path = '/etc/mosquitto/mosquitto.conf', type = 'mosquitto.conf' }
]
```

The daemon is started/enabled via:

```
sudo systemctl start c8y-configuration-plugin
sudo systemctl enable c8y-configuration-plugin
```

However, keep in mind that the daemon has to be restarted every time the `/etc/tedge/c8y/c8y-configuration-plugin.toml` is touched via the command line. If initially started however the `c8y-configuration-plugin.toml` can be managed from the ui directly.





## Log-Files

With [thin-edge.io](https://thin-edge.io) you can request log files from a device by using the [Cumulocity log request feature](#) as a part of Device Management.

This functionality is also directly installed with the initial script. However, you need to configure the `/etc/tedge/c8y/c8y-log-plugin.toml` and add the entries for the log files that can be requested. Just copy the following content to that file:

```
files = [
  { type = "software-management", path = "/var/log/tedge/agent/software-*" },
  { type = "mosquitto", path = "/var/log/mosquitto/mosquitto.log" },
  { type = "daemon", path = "/var/log/daemon.log" },
  { type = "user", path = "/var/log/user.log" },
  { type = "apt-history", path = "/var/log/apt/history.log" },
  { type = "apt-term", path = "/var/log/apt/term.log" },
  { type = "auth", path = "/var/log/auth.log" },
  { type = "dpkg", path = "/var/log/dpkg.log" },
  { type = "kern", path = "/var/log/kern.log" }
]
```

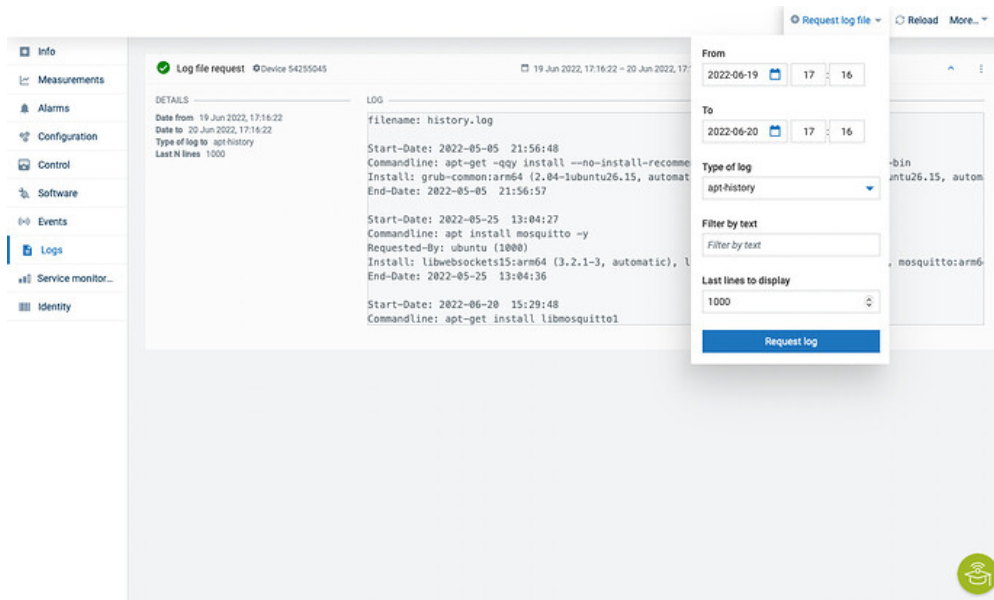
The daemon is started/enabled via:

```
sudo systemctl start c8y-log-plugin
sudo systemctl enable c8y-log-plugin
```

If you add the `c8y-log-plugin.toml` into the `c8y-configuration-plugin.toml` you can to the administration from there.

However, keep in mind that the daemon has to be restarted every time the `/etc/tedge/c8y/c8y-log-plugin.toml` is touched via the command line.





## Outlook & Summary

In this guide, we installed [thin-edge.io](#) on a Revolution Pi from Kunbus. Installation and configuration were straight forward and thanks to the device management capabilities of Cumulocity I have a fully managed RevPi connected to my tenant. I can change configurations, check the device status, request for log files and even install or update software components.

However, the journey does not stop there. [thin-edge.io](#) offers many possibilities such as writing your own [software management plugins](#) or even your own [operation plugins](#). You can find many examples on the official [repository](#) in the examples section.

The Revolution Pi itself has many connectivity modules for various fieldbus protocols and is thus very powerful gateway hardware. In terms of installation, there will be also a new firmware version where [thin-edge.io](#) is already preinstalled and only needs to be configured within the local web service delivered by Kunbus. So stay tuned.

---

This article is part of the TECHniques newsletter blog - technical tips and tricks for the Software AG community. [Subscribe](#) to receive our quarterly updates or read the [latest issue](#).

7 Likes

---

[Top content and updates – H1, 2022](#)

---

[TECHniques - Issue 3, 2022](#)

---

[ThinEdge Client can't connect to Cumulocity](#)

---

[Send data from raspberry\\_pi to platform cumulocity iot](#)

---

[anon31993359](#) 2 August 18, 2022, 9:40am

great article [@Murat\\_Bayram](#) , its worth noting that this tutorial would also work for a normal RaspberryPi 3 or 4.

---

[Albin\\_Suresh](#) (Albin Suresh) 3 October 19, 2022, 8:45am

Impressed to see pretty much all the features of thin-edge covered in this single article and keeping it brief at the same time. Great work [@Murat\\_Bayram](#) .

---

[Reuben\\_Miller2](#) (Reuben Miller) 4 January 26, 2023, 9:54am

This page was so well received we decided to take most of the details from here and put it in the [thin-edge.io](#) website docs under the getting started section.

You can checkout the docs here:

<https://thin-edge.github.io/thin-edge.io/html/tutorials/getting-started.html>

2 Likes