



User manual  
**netFIELD Compact X8M**



**Hilscher Gesellschaft für Systemautomation mbH**  
**[www.hilscher.com](http://www.hilscher.com)**

DOC220302UM01EN | Revision 1 | English | 2022-05 | Released | Public

# Table of contents

<b>1</b>	<b>Introduction</b> .....	<b>4</b>
1.1	About this document .....	4
1.1.1	Description of the contents .....	4
1.1.2	List of revisions .....	4
1.1.3	Conventions in this document.....	5
1.2	Terms and abbreviations.....	6
1.3	Brief description .....	7
1.4	netFIELD OS: Industrial IoT Operating System .....	9
<b>2</b>	<b>Safety</b> .....	<b>13</b>
2.1	Intended use .....	13
2.2	General note .....	13
2.3	Personnel qualification .....	13
2.4	Device destruction by exceeding the allowed supply voltage .....	14
2.5	Risk of denial of service due to extensive memory usage close to limits.....	14
2.6	Modification reserved .....	14
<b>3</b>	<b>Hardware description</b> .....	<b>15</b>
3.1	Device overview .....	15
3.2	Dimensions .....	16
3.3	Interfaces .....	17
3.3.1	Ethernet LAN connectors.....	17
3.3.2	Supply voltage .....	17
3.3.3	RS-232 / RS-485 interface .....	18
3.3.4	USB connectors .....	20
3.3.5	Console interface .....	20
3.4	LEDs .....	21
3.4.1	LEDs of the LAN interface .....	21
3.4.2	Edge LED.....	21
<b>4</b>	<b>Commissioning and first steps</b> .....	<b>23</b>
4.1	Overview .....	23
4.1.1	netFIELD Cloud user .....	23
4.1.2	Standard Docker user .....	24
4.2	Installation .....	25
4.3	Establish LAN connection and login to Local Device Manager.....	28
4.3.1	Overview .....	28
4.3.2	Using DHCP server.....	29
4.3.3	Establishing one-to-one connection to device (without DHCP server) .....	31
4.3.4	Login to Local Device Manager .....	32
4.4	Set system time.....	35
4.5	"Onboard" (register) device in netFIELD Cloud .....	37
4.5.1	Overview .....	37
4.5.2	Onboarding using the "Basic" method .....	38
4.5.3	Onboarding using the "Advanced" method .....	40
<b>5</b>	<b>Local Device Manager</b> .....	<b>47</b>
5.1	Overview .....	47

5.2	System .....	49
5.3	Networking .....	52
5.3.1	Overview .....	52
5.3.2	Firewall.....	57
5.3.3	Network Proxy settings .....	66
5.4	Networking Services .....	72
5.4.1	Wi-Fi.....	72
5.4.2	DHCP Server .....	72
5.4.3	Connectivity Check .....	73
5.5	Onboarding (and offboarding) .....	75
5.6	General Settings .....	78
5.6.1	Web Server (Port) Settings .....	78
5.6.2	Default MQTT Client Settings .....	79
5.6.3	Docker Network Settings .....	81
5.6.4	Remote Access .....	85
5.6.5	Login .....	87
5.7	Standard Docker .....	88
5.8	IoT Edge Docker .....	95
5.9	Accounts .....	101
5.10	Certificate .....	104
5.11	Terminal .....	105
5.12	System Update.....	106
5.13	Logs .....	111
5.14	Services .....	112
<b>6</b>	<b>Good to know .....</b>	<b>115</b>
6.1	Device recovery via USB .....	115
6.2	Useful CLI commands and parameters in Terminal.....	119
6.2.1	Network Manager.....	119
6.2.2	Show interface status.....	119
6.2.3	Activate interface .....	119
6.2.4	Docker Compose support for Standard Docker environment .....	119
6.2.5	Manage Standard Docker .....	119
6.2.6	Manage IoT Edge Docker .....	119
6.2.7	External storage support using iSCSI .....	120
6.2.8	Enable/disable SSH Daemon (release port 22) .....	120
6.2.9	Follow the system log via terminal CLI .....	120
6.2.10	Configure operating mode of serial interface .....	120
<b>7</b>	<b>Technical data .....</b>	<b>121</b>
<b>8</b>	<b>Decommissioning, dismantling and disposal .....</b>	<b>123</b>
8.1	Dismantling .....	123
8.2	Disposal .....	124
<b>9</b>	<b>Legal notes .....</b>	<b>127</b>
	<b>List of figures .....</b>	<b>132</b>
	<b>List of tables.....</b>	<b>135</b>
	<b>Contacts.....</b>	<b>136</b>

# 1 Introduction

## 1.1 About this document

### 1.1.1 Description of the contents

This user manual describes the **netFIELD Compact X8M** edge gateway (NFX8M-D2-N32-010) from Hilscher. It contains a description of its hardware, its technical data and a description of its web-based configuration GUI (Local Device Manager).

It also provides instructions on how to commission the gateway “in the field” and how to “onboard” it in the netFIELD Cloud for remote management (optional use case for users/subscribers of the netFIELD Cloud).

Note that for ease of reading, the edge gateway is referred to simply as “device” in this document.

### 1.1.2 List of revisions

Index	Date	Author	Revision
1	2022-05-03	MKE	Document created

Table 1: List of revisions

### 1.1.3 Conventions in this document

Notes, instructions and results of operating steps are marked as follows:

#### Notes



---

**Important:**

<important note you must follow to avoid malfunction>

---



---

**Note:**

<general note>

---



---

<note on further information>

#### Instructions

1. Operational step
  - Instruction
  - Instruction
2. Operational step
  - Instruction
  - Instruction

#### Results

↻ Intermediate result

⇒ Final result

## 1.2 Terms and abbreviations

Term	Description
Container	Executable software package including all components needed to run an application on a Docker engine.
Docker	Software for isolating applications using container virtualization. Docker enables the creation and operation of Linux containers. netFIELD OS provides two Docker runtime environments: <i>IoT Edge Docker</i> and <i>Standard Docker</i> . The <i>IoT Edge Docker</i> environment is managed remotely from the <i>netFIELD Platform</i> .
IIoT	Industrial Internet of Things.
IT network	Information technology network
Microsoft Azure IoT Edge	Features a deployable Docker-based runtime along with a public cloud-hosted backend logic for remote device servicing. It is the basic framework of the evolved netFIELD device-to-cloud communication infrastructure.
netFIELD App	netFIELD application Docker container from Hilscher. Runs in the <i>IoT Edge Docker</i> or <i>Standard Docker</i> of the netFIELD OS on the netFIELD Edge.
netFIELD Cloud	Internet-hosted platform providing APIs for cloud-to-cloud and cloud-to-edge communication. Based on <i>Microsoft Azure IoT Edge</i> . Consists of the netFIELD Platform (backend) and the netFIELD Portal (web-based user interface/frontend). The netFIELD Cloud is also referred to as <i>netFIELD.io</i>
netFIELD Edge	Gateway devices or systems running the netFIELD OS, providing connectivity to the netFIELD Cloud. Cloud connectivity is based on <i>Microsoft Azure IoT Edge</i> .
netFIELD.io	Internet-hosted platform providing APIs for cloud-to-cloud and cloud-to-edge communication. Based on <i>Microsoft Azure IoT Edge</i> . Consists of the netFIELD Platform (backend) and the netFIELD Portal (web-based user interface/frontend). netFIELD.io is also referred to as <i>netFIELD Cloud</i> .
netFIELD OS	Cross-platform capable Linux operating system providing core OS functions plus optional connectivity to the netFIELD Cloud. Cloud connectivity is based on <i>Microsoft Azure IoT Edge</i> .
netFIELD Platform	Backend of the netFIELD Cloud, providing APIs for cloud-to-cloud and cloud-to-edge communication.
netFIELD Portal	Web-based user interface (frontend) of the netFIELD Cloud.
OT network	Operational technology network.

Table 2: Terms and abbreviations

## 1.3 Brief description

### Overview

netFIELD Compact X8M is an edge gateway for running custom software, close-to-machine deployment of workloads and distributing IIoT logic and intelligence.

Connecting your netFIELD Compact over the Internet to the netFIELD Cloud (<https://www.netfield.io>) allows you to manage your device via remote control functions from the netFIELD Portal (which is the web-based user interface of the netFIELD Cloud) and to control the distribution of its IIoT applications from remote over the Internet.

You can also stream MQTT messages from your device to 3<sup>rd</sup> party applications via the RESTful API of the netFIELD Platform (which is the backend of the netFIELD Cloud).



---

**Note:**

Note that connecting the netFIELD Compact device to the netFIELD Cloud requires an account/subscription for the *netFIELD Cloud services* (<https://www.netfield.io>).

Contact your local Hilscher sales representative for information on terms and conditions.

---

### Key features hardware

- 1.8 GHz quad-core ARM Cortex-A53 64Bit CPU (NXP *i.MX8M Mini*)
- 2 GB LPDDR4 RAM
- 32 GB eMMC flash storage
- 2 x Ethernet ports: 1 Gbit/s and 100 Mbit/s (RJ45 connectors)
- 3 x USB 2.0 interfaces
- RS-232 or RS-485 (switchable) serial interface (terminal block connector)
- Serial console interface (UART-to-USB)
- Programmable LED
- Compact aluminum housing
- Protection class IP20
- Removable mounting bracket for DIN top hat rail or wall mounting
- Wide input voltage range of +8 V to +36 V DC (typical: +24 V DC)
- Wide ambient temperature range of –20 °C to +60 °C
- Real-time clock buffering via on-board coin cell battery

### Key features software

- Preinstalled and non-removable **netFIELD OS** (netFIELD Operating System) with **Local Device Manager** (web GUI) for local configuration and administration of the device.  
For more details and information about the netFIELD OS, see section *netFIELD OS: Industrial IoT Operating System* [▶ page 9].
- “High Assurance Boot”: Only Hilscher-authenticated netFIELD OS core software can run on the device at any time.
- Applications for data acquisition, analytics, processing or connectivity (to cloud or other enterprise systems) do not run natively under the netFIELD OS, but as “containers” in a Docker runtime. netFIELD OS provides two Docker runtimes that are running simultaneously on the device:
  - **IoT Edge Docker** for remote and automatic deployment and maintenance of containers. These containers are deployed (“pulled”) and managed over the netFIELD Cloud. This requires your device to be onboarded in the netFIELD Cloud. Note that you need an account/subscription for the netFIELD Cloud (<https://www.netfield.io>) for this.
  - **Standard Docker** for manual and local deployment and maintenance of containers.  
Those containers can be pulled from official registries like Docker Hub (<https://hub.docker.com>) or any locally hosted Docker registry. In case you do not have an account/subscription for the *netFIELD Cloud*, the standard Docker is the only way to pull and run container applications on your device.
- Options for users/subscribers of the netFIELD Cloud (<https://www.netfield.io>):
  - **Remote device management** and container deployment via netFIELD Portal.



For a detailed description of the netFIELD Portal, see operating instruction manual *netFIELD Portal*, DOC1907010IxxEN.

---

- The **WebSocket** interface of the netFIELD Platform allows you to stream aggregated data for monitoring, long-term analytics or diagnostics with your own applications via **REST API** (see <https://api.netfield.io/v1/documentation>).  
You can easily create your own and personal API key needed for authorization with different levels of permission in the **API Key Manager** of the netFIELD Portal.
- **Software updates** for the device (application containers and netFIELD OS) can be conveniently initialized and managed via the netFIELD Portal.



## 1.4 netFIELD OS: Industrial IoT Operating System

The netFIELD OS supports scalable field device hardware depending on the customer's use case. In order to achieve this, applications do not run directly on the host system but instead as containers in a Docker runtime. Our OS is very lean and only supports the essential services required by the customer's network infrastructure.

### Features

- **Run containers:** Containers are revolutionizing connected IoT devices, and netFIELD OS is the perfect match to run them.
- **Manage device:** Manage your device locally with a web-based interface. It is easy to administer storage, configure networks, and more.
- **Build to last:** Build to survive in harsh environments like unexpected shutdowns with security in mind.
- **Easy to port:** Based on a Yocto project (<https://www.yoctoproject.org>) maintained Linux for easy porting to most capable device types across various CPU architectures.

### Architecture

Hilscher netFIELD OS is a secure operating system that makes it easy to program, deploy, connect and manage Edge Devices. Hilscher netFIELD OS extends the Linux kernel, with software libraries to securely connect operation technology like PLC, MES, Historians, Files or other on-premise systems with IT services like the netFIELD Portal. Our OS lets you innovate faster embracing container technologies managed by the netFIELD Portal from a central point or locally at the edge.

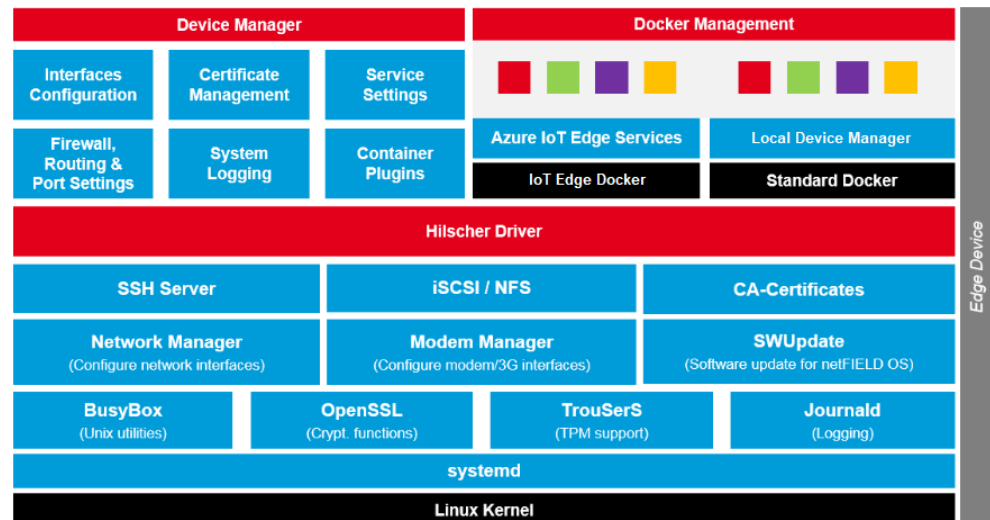


Figure 1: netFIELD OS architecture

## Core services

The netFIELD OS core services include the support of hardware interfaces, the network environment, secure communication and system logging. In order to support the customer in setting up the gateway configuration, the Local Device Manager is coming along with the core services. With the open plug-in mechanism, the functionality of the Local Device Manager can be easily extended with the help of containerized applications.

## Container management

Application containers can run in the IoT Edge Docker or Standard Docker environment and do contain business logic such as for data acquisition, analytics, processing or connectivity to cloud or enterprise systems.

The container management provides the functionality to pull and run containers on the device itself. Before a container can be run, its image needs to be pulled from a certain container registry. After that the container is created, the application can be then controlled by using the start / stop commands or by enabling the autostart option. Also, the deletion of containers and images is a part of container management. In order to enable the field devices for off- and online scenarios, netFIELD OS provides two Docker runtime environments at the same time.

The IoT Edge Docker environment is managed by the netFIELD.io (netFIELD Platform) remotely. That is why there is no need to have direct access to the netFIELD Edge Device, as long as the device can hold its connection to netFIELD.io.

Administrators can be anywhere and have full management access to the device with the stored images and have the ability to control the application containers remotely. Otherwise, the Standard Docker can be used locally if the netFIELD Edge Device is not connected to netFIELD.io. In this case, the Standard Docker runtime environment can be managed by the Local Device Manager, by the netFIELD OS command line interface or by a web application like *portainer.io*, which can be deployed as container.

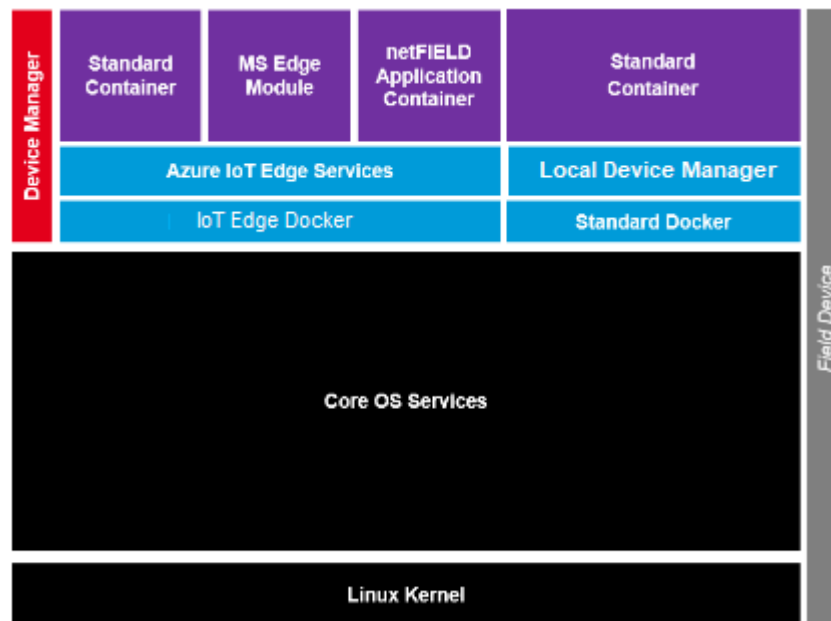


Figure 2: netFIELD OS container management

## Inter-container communication

Application containers usually focus on the dedicated business logic in order to avoid the development of unmaintainable software monoliths. In this scenario, multiple containers need to work together to realize customer use cases. Our powerful message and container-oriented architecture provide the highest level of flexibility and reusability when implementing customer solutions with individual requirements. This reduces IoT solution cost in development and operation.

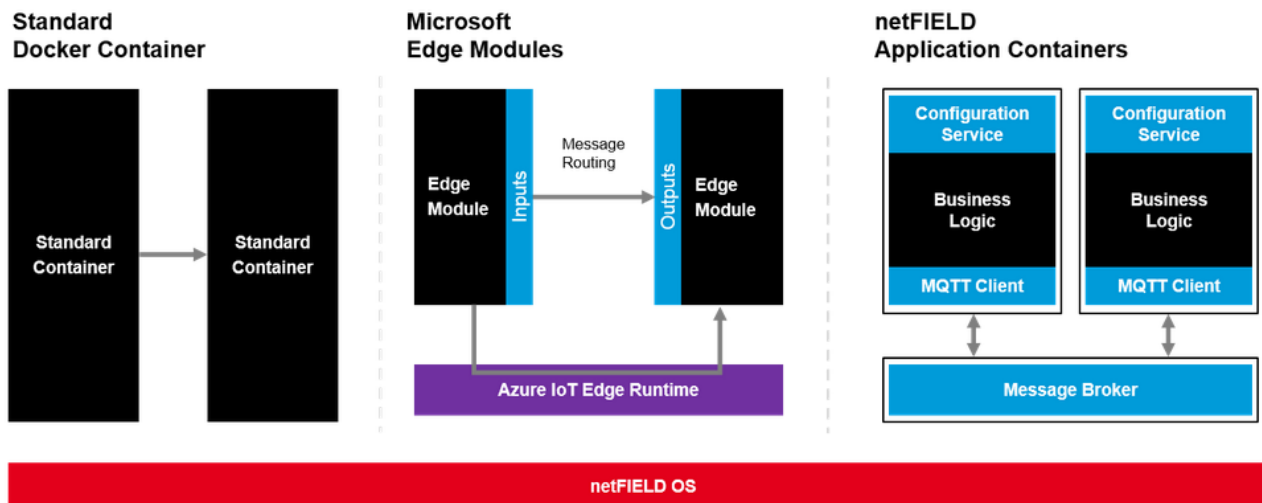


Figure 3: netFIELD OS inter-container communication

**Services supported by netFIELD OS**

- Network interface configuration
- Firewall configuration (NAT, TCP/IP port management)
- HTTP(S) Proxy Server configuration
- Network storage (NFS, iSCSI) support
- Resources monitoring
- Access to netFIELD OS and Docker services via a web-terminal, serial console or SSH
- Standard Docker instance for locally managed containers, including Docker Compose support
- IoT Edge Docker instance for application containers managed via netFIELD Cloud
- netFIELD OS update (local/remote) support
- System and container logging
- Onboarding in netFIELD Cloud
- Secure communication to the netFIELD Cloud services
- Remote device control/access via netFIELD Cloud, protected by “four-eyes principle”: Must be explicitly enabled by the user in the Local Device Manager
- Selection of upstream (device-to-cloud) protocol to the netFIELD Cloud: AMQP, AMQPWS, MQTT or MQTTWS. Note that the protocols use different ports, which is relevant to your firewall configuration
- Management of Linux services in Local Device Manager
- Connectivity check for IoT Edge Docker in Local Device Manager

## 2 Safety

### 2.1 Intended use

netFIELD Compact X8M is an edge gateway for running custom software, close-to-machine deployment of workloads and distributing IIoT logic and intelligence. Additionally it allows you to connect it with netFIELD Cloud in order to control the device itself and the distribution of those applications from remote over the Internet.

The device complies with protection class IP20 and is suitable for indoor use, e.g. in industrial automation plants.  
Any use other than specified here is not permitted.

### 2.2 General note

To avoid personal injury or property damage to your system or to this product, you must read and understand all instructions in this manual before using the product.

This manual was written for the use of the product by educated personnel. When using the product, all safety instructions and all valid legal regulations must be obeyed. Technical knowledge is presumed.

Keep this manual for future reference.

### 2.3 Personnel qualification

The device may only be installed, configured, operated and removed by qualified personnel. Job-specific technical skills for people professionally working with electricity must be present concerning the following topics:

- Safety and health at work
- Mounting and attaching of electrical equipment
- Measurement and analysis of electrical functions and systems
- Evaluation of the safety of electrical systems and equipment
- Installing and configuring IT

## 2.4 Device destruction by exceeding the allowed supply voltage

Observe the following notes concerning the voltage supply:

- The device may only be operated with the specified supply voltage of +8 V ... +36 V DC. Make sure that the limits of the allowed range for the supply voltage are not exceeded.
- A supply voltage above the upper limit can cause severe damage to the device!
- A supply voltage below the lower limit can cause malfunction of the device.

## 2.5 Risk of denial of service due to extensive memory usage close to limits

Using applications that exceed the memory resources of the device can cause an out-of-memory situation (OOM) in the Linux kernel leading to temporary delayed application reaction times and limited overall device responsiveness.

You must therefore consider the memory requirements of your Docker containers carefully before deploying them on the device. We also recommend you to configure the logging behavior (e.g. log levels) of your containers accordingly.

For information on the memory resources of the device, see section *Technical data* [► page 121].

## 2.6 Modification reserved

Changes or modifications to hardware or software of the device by the user are not permitted.

The device does not contain any parts that are serviceable by the user. Do not open or damage the housing.

## 3 Hardware description

### 3.1 Device overview

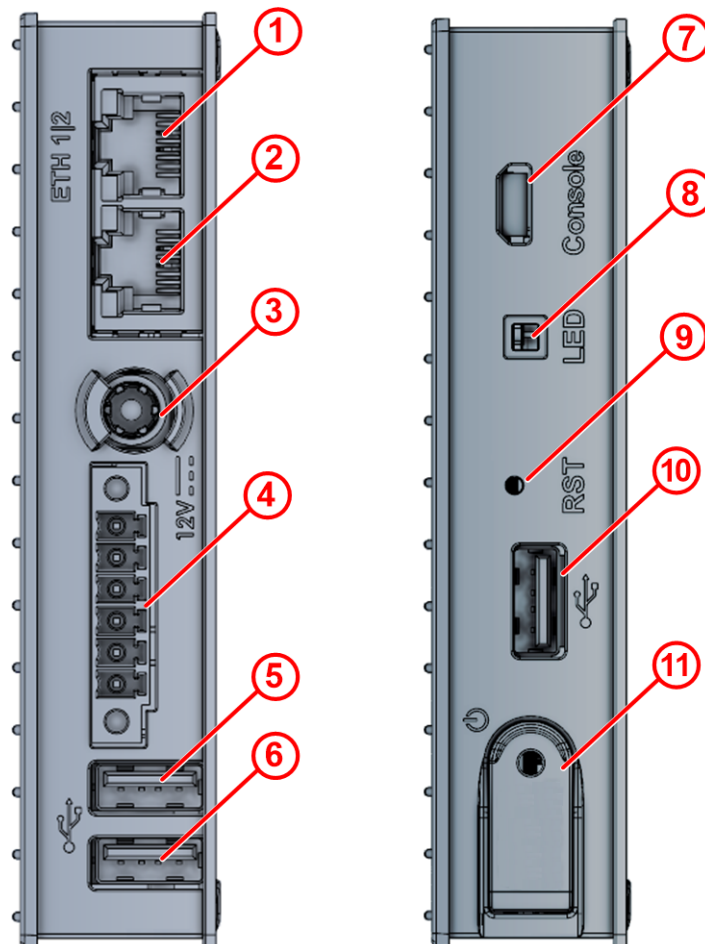


Figure 4: Positions on netFIELD Compact

Pos.	Label/name	Description	For details see
(1)	ETH2	100 Mbit/s RJ45 Ethernet port	Ethernet LAN connectors [▶ page 17]
(2)	ETH1	1 Gbit/s RJ45 Ethernet port	
(3)	12V	Supply voltage input connector (+8 ... +36 V DC, typical: 24 V)	Supply voltage [▶ page 17]
(4)	-	RS-232 and RS-485 (2-wire) interface connector	RS-232 / RS-485 interface [▶ page 18]
(5)	-	USB 2.0 type A connector	USB connectors [▶ page 20]
(6)	-	USB 2.0 type A connector	
(7)	Console	Serial debug console connector (UART-to-USB bridge over Micro USB)	Console interface [▶ page 20]
(8)	LED	User-programmable LED	EDG LED
(9)	RST	Button for resetting the netFIELD OS. <b>Note:</b> For proper shut-down of the netFIELD OS and the device, use the power button (11).	-
(10)	-	USB 2.0 type A connector	USB connectors [▶ page 20]
(11)	-	Power button with LED indicator.	-

Table 3: Positions on netFIELD Compact device

## 3.2 Dimensions

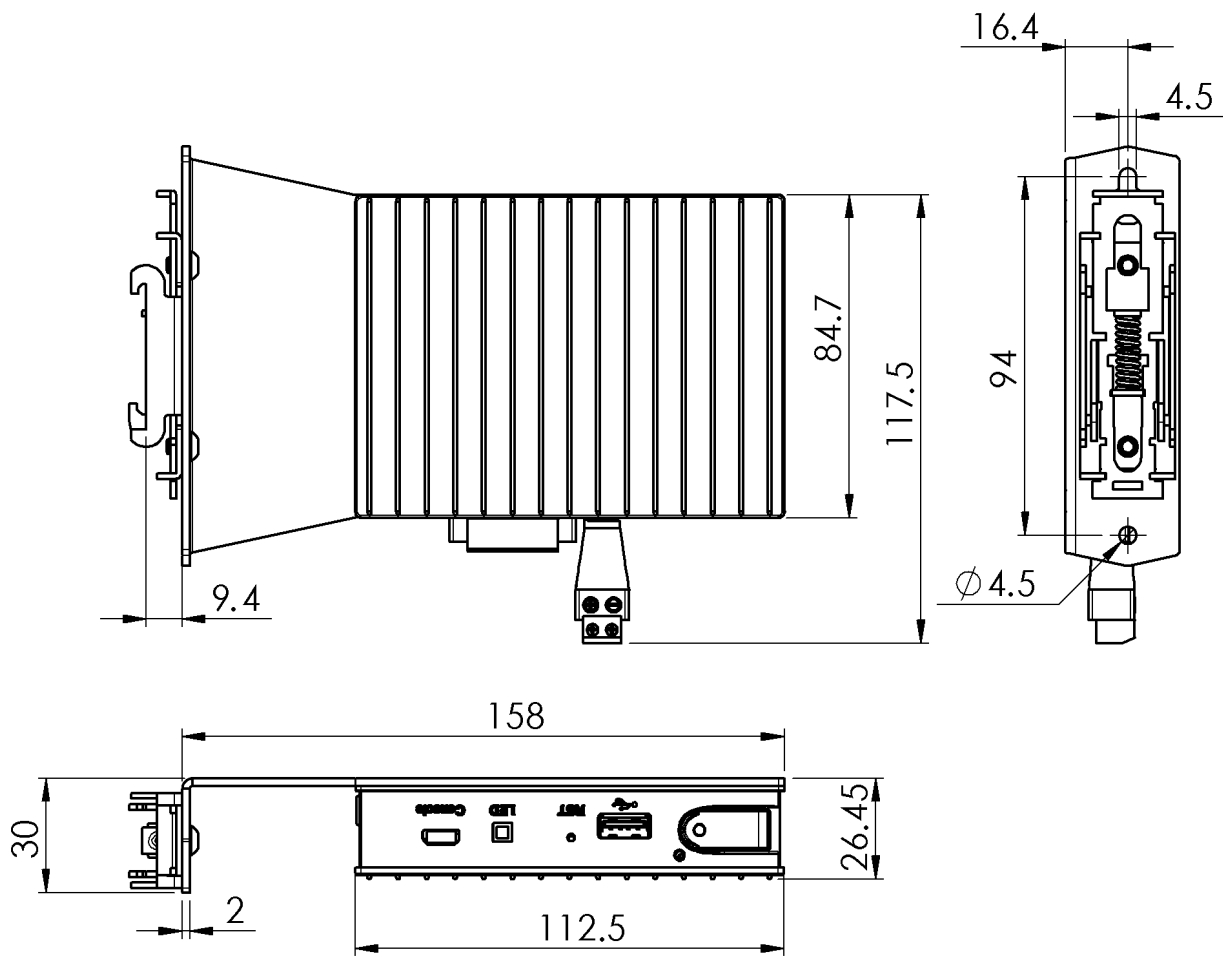


Figure 5: Dimensions in millimeters



## 3.3 Interfaces

### 3.3.1 Ethernet LAN connectors

The two RJ45 Ethernet connectors allow you to connect your device to your local IT network and/or to the Internet (e.g. the netFIELD Cloud). The MAC addresses of the ports are printed on the device label. Note that you can change the IP address configuration of these ports in the Local Device Manager (see section *Networking* [▶ page 52]).

#### ETH1

Primary 1 Gbit/s Ethernet port (see position (2) in section *Device overview* [▶ page 15]).

The logical name of ETH1 in netFIELD OS is **eth0**. The “factory setting” of the IP address of this port is DHCP mode (“fallback” is *link-local*, i.e. address block 169.254.0.0/16).

#### ETH2

Secondary 100 Mbit/s Ethernet port (see position (1)). The logical name of ETH2 in netFIELD OS is **eth1**. The “factory setting” of the IP address of this port is 192.168.253.1/24.

### 3.3.2 Supply voltage

The power input jack (see position (3) in section *Device overview* [▶ page 15]) allows you to connect supply voltage in the range of +8 V to +36 V DC (typical: 24 V).

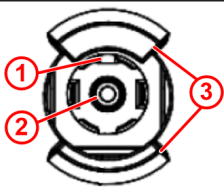
DC jack	Pin	Signal
	1	GND
	2	+8 V ... +36 V DC IN (typical: +24 V DC)
	3	“Bajonet” brackets for locking the plug to the device

Table 4: Supply voltage connector

The input jack mates with a 5.5 x 2.5 mm coaxial (“barrel”) plug. The device is shipped with an adapter plug (5.5 x 2.5 mm coaxial to 2-pin terminal block) for easy wiring:



Figure 6: Adapter plug

When connecting, turn the plug clockwise to lock it to the device (“bajonet”-type mounting).

### 3.3.3 RS-232 / RS-485 interface

The terminal block connector (see position (4) in section *Device overview* [▶ page 15]) is a 16550 UART-compatible serial port that can be used either as RS-232 or RS-485 (2-wire) interface.

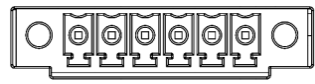
Serial interface	Pin	RS-232 signal	RS-485 signal
	1	RS232_TXD	RS485_NEG
	2	RS232_RTS	RS485_POS
	3	GND	GND
	4	RS232_CTS	N.C.
	5	RS232_RXD	N.C.
	6	GND	GND

Table 5: Serial interface

The device is shipped with a matching COMBICON connector for easy wiring:

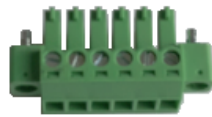


Figure 7: COMBICON plug

#### Configuring the operating mode

The operating mode of the interface (`uart1` in Linux) can be configured in the Linux terminal with the following commands:

Set remanently to RS-485:

```
sudo sh -c "echo 'mode=rs485' > /etc/default/uart1"
systemctl restart uart1-mode
```

Set remanently to RS-232:

```
sudo sh -c "echo 'mode=rs232' > /etc/default/uart1"
systemctl restart uart1-mode
```



#### Note:

In case the `/etc/default/uart1` file is missing, the UART is configured to RS-485 mode by default.

#### Standard Linux device support

netFIELD OS supports the serial interface as standard Linux device `/dev/ttymx0`

Note that the `uart1-mode` service must be enabled and running for this (the `uart1-mode` service is enabled and running by default).

If you stop the `uart1-mode` service (e.g. with the `systemctl stop uart1-mode` command), the `ttymxc0` interface will not be available any longer.

## Enabling access for application containers

A Docker application container accessing the `ttymxc0` serial interface must be either running as `root` user or as a member of the `dialout` group in Linux. If your container is not a `root` user, you can add the container to the `dialout` group with the `--group-add dialout` parameter in your `docker run` command during container deployment. Note that you must also map the `/dev/ttymxc0` interface into the container.

The following example shows a `docker run` command for a Node-RED container that would allow the container to access the interface:

```
docker run -d -p 1880:1880 --device=/dev/ttymxc0:/dev/ttymxc0 --group-add dialout nodered/node-red
```

If the container is deployed via Docker Compose, you would have to add the following lines in the `*.yaml` file:

```
devices:
  - "/dev/ttymxc0:/dev/ttymxc0"
group_add:
  - dialout
```

### 3.3.4 USB connectors

The three USB 2.0 type-A connectors (see positions **(5)**, **(6)** and **(10)** in section *Device overview* [▶ page 15]) allow you to connect peripheral devices such as Wi-Fi or LTE adapters, scanners, mass storage drives etc.



**Note:**

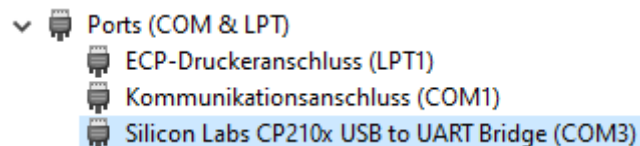
The maximum allowed output current over all USB ports is 1 A.

The USB ports can also be used for the device recovery procedure (see section *Device recovery via USB* [▶ page 115]).

### 3.3.5 Console interface

The micro USB jack (see position **(7)** in section *Device overview* [▶ page 15]) is linked to an internal UART-to-USB bridge allowing you to access the netFIELD OS from a terminal program on your engineering PC (like e.g. *PuTTY*) via USB connection.

After connecting the interface to your PC, the console port is displayed in the Windows Device Manager under *Ports (COM & LPT)* as “Silicon Labs CP210x USB to UART Bridge (COMx)”:



Use the following serial port settings in your terminal program to access the console:

Parameter	Value
Baud rate	115200
Data bits	8
Stop bits	1
Parity	none
Flow control	none

Table 6: Serial port settings of console

## 3.4 LEDs

### 3.4.1 LEDs of the LAN interface

The LEDs of the RJ45 Ethernet jacks (see positions (1) and (2) in section *Device overview* [▶ page 15]) indicate the state of the Ethernet connection.






LED	Color	State	Meaning
LINK	<b>Duo LED green/orange</b>		
	 (green)	On	1 Gbit or 100 Mbit network connection
	 (off)	Off	10 Mbit or no network connection
ACT	<b>LED yellow</b>		
	 (yellow)	On	The device does not send/receive Ethernet frames.
	 (yellow)	Flickering (load dependent)	The device sends/receives frames.
	 (off)	Off	The device does not send/receive Ethernet frames.

Table 7: LEDs LAN interface

### 3.4.2 Edge LED

The Edge LED (labelled **LED**, see position (8) in section *Device overview* [▶ page 15]) is a user-programmable yellow/green Duo LED.

It can be controlled via Linux in the directories:

```
/sys/class/leds/edge_green/brightness
/sys/class/leds/edge_yellow/brightness
```

For example, you can switch the LED to steady green by entering the following command in the terminal:

```
sudo sh -c "echo '1' >> /sys/class/leds/edge_green/brightness"
```

The green light can be switched off again by entering:

```
sudo sh -c "echo '0' >> /sys/class/leds/edge_green/brightness"
```



For more information on how to control LEDs under Linux, please refer to *LED handling under Linux* on

<https://www.kernel.org/doc/html/latest/leds/leds-class.html>.

## Handling via netFIELD App Platform Connector

If you onboard your netFIELD Compact in the netFIELD Cloud and deploy the *netFIELD App Platform Connector* from the netFIELD Cloud, the Edge LED will be automatically controlled by the *Platform Connector*. Note that your custom Linux programming of the LED may thus be overridden.

The Platform Connector controls the EDG LED by subscribing to the MQTT topic

netfield/internal/netfield-app-platform-connector/edge-led/state

Your application(s) can set the LED state by publishing the following JSON object to this topic:

```
{
  "schemaVersion": 1,
  "nodeId": "state",
  "messageType": "event",
  "dataType": "object",
  "data": [
    {
      "state": <value>
    }
  ]
}
```

Replace <value> with one of the following numbers:

Value	Sets state
0	LED static off
1	LED fast blinking yellow at 5 Hz
2	LED slow blinking green at 1 Hz
3	LED static green on

Table 8: Setting EDG LED state in MQTT topic

## 4 Commissioning and first steps

### 4.1 Overview

#### 4.1.1 netFIELD Cloud user

The following table shows the steps that you must perform in order to commission your netFIELD Compact device if you are a netFIELD Cloud user/subscriber and want to manage your device from the netFIELD Cloud (<https://www.netfield.io>).

#	Step	For details see
0	Requirement: You have a netFIELD Cloud account ( <a href="https://www.netfield.io">https://www.netfield.io</a> )	Contact your local Hilscher sales representative for information on terms and conditions.
1	Install the device.	Section <i>Installation</i> [▶ page 25]
2	Establish LAN connection and login to Local Device Manager.	<i>Establish LAN connection and login to Local Device Manager</i> [▶ page 28]
3	Make sure that the Ethernet network to which you connect the device provides Internet access.	-
4	If applicable (if your LAN uses HTTP/HTTPS/FTP proxy servers): Configure the netFIELD OS for using proxy server.	Section <i>Network Proxy settings</i> [▶ page 66]
5	If applicable (if the default Docker IP addresses are not compatible with your LAN): Customize Docker Network Settings.	Section <i>Docker Network Settings</i> [▶ page 81]
6	Set local system time.	Section <i>Set system time</i> [▶ page 35]
7	Optional: Configure netFIELD OS firewall. <b>Note:</b> By default, the internal netFIELD OS firewall allows all traffic ("trusted zone"). When you assign an interface or subnet to the drop or block zone, make sure that you open the ports that are used by your application containers.	Section <i>Firewall</i> [▶ page 57]
8	"Onboard" (register) device in the netFIELD Cloud <b>Note:</b> Make sure that your company's firewall does not block the TCP port (outgoing) of the upstream protocol (device-to-cloud communication) that you intend to use. MQTT: 8883 MQTT over WebSocket: 443 AMQP: 5671 (default) AMQP over WebSocket: 443	Section <i>"Onboard" (register) device in netFIELD Cloud</i> [▶ page 37]
9	Optional: Deploy application container(s) from netFIELD Portal (if not already deployed through Deployment Manifest).	Section <i>Deploying containers on your device</i> in the operating instruction manual <i>netFIELD Portal</i> , DOC190701O1xxEN

Table 9: Tasks for commissioning the netFIELD Compact for netFIELD Cloud usage

## 4.1.2 Standard Docker user

The following table shows the steps that you must perform in order to commission your netFIELD Compact device if you want to run your application containers in the *Standard Docker* of the device (i.e. if you do not want to connect your device to the Internet for managing it from the netFIELD Cloud).

#	Step	For details see
1	Install the device.	Section <i>Installation</i> [▶ page 25]
2	Establish LAN connection and login to Local Device Manager.	Section <i>Establish LAN connection and login to Local Device Manager</i> [▶ page 28]
3	If applicable (if your LAN uses HTTP/HTTPS/FTP proxy servers): Configure sensorEDGE for using proxy server.	Section <i>Network Proxy settings</i> [▶ page 66]
4	If applicable (if the default Docker IP addresses are not compatible with your LAN): Customize Docker Network Settings.	Section <i>Docker Network Settings</i> [▶ page 81]
5	Set local system time.	Section <i>Set system time</i> [▶ page 35]
6	Deploy and run containers in the Standard Docker.	-

Table 10: Tasks for commissioning the netFIELD Compact for Standard Docker usage



## 4.2 Installation

The netFIELD Compact device is shipped with an attached metal bracket for easy DIN top hat rail or wall mounting.

### DIN top rail mounting

- Press down the spring-loaded lower rail clamps at the back of the mounting bracket.

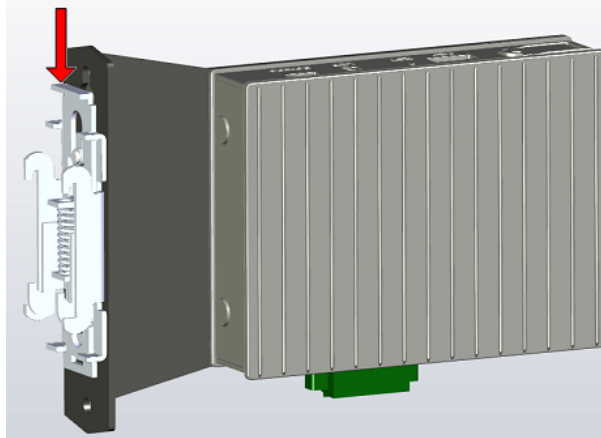


Figure 8: Spring-loaded rail clamps

- Hook the upper clamps onto the DIN top hat rail, then release the lower clamps to lock the bracket onto the rail.



Figure 9: Device mounted on DIN top hat rail

## Wall mounting

If your mounting site does not provide a top hat rail, you can screw the device directly to a flat solid base, like e.g. a wall.

- For this, you first have to remove the rail clamps from the rear end of the mounting bracket, which are attached to the bracket with two hexagon head screws. Use an M4 Allen key to remove the screws:

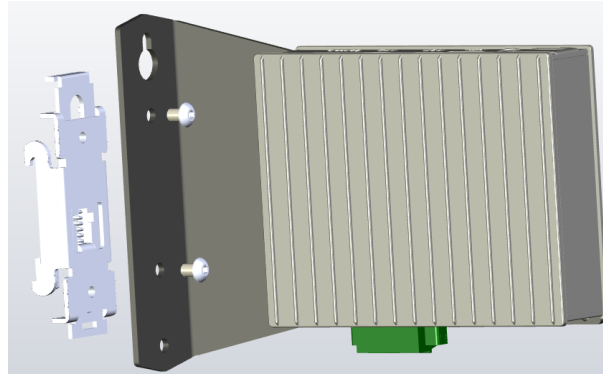


Figure 10: Remove rail clamp screws

- After having removed the rail clamps, hold the rear end of the mounting bracket at the desired position and mark the two places where the threads for the screws are to be cut into the wall.
- Use the M4 tap to cut an M4 thread at each of the two marked points, if necessary pre-drill with a drilling machine first.
- Use the Allen wrench/key respectively screwdriver and two M4 cylinder head screws of suitable length to screw the mounting bracket to the wall.

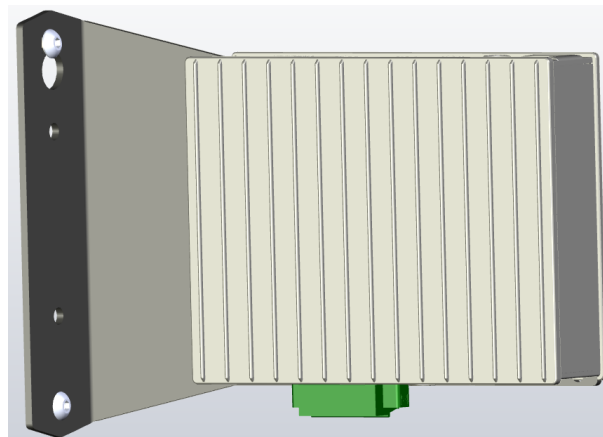


Figure 11: Wall mounted

### Connecting supply voltage

- After mounting, connect a supply voltage in the range of +8 V ... +36 V DC (typical: 24 V) to the power input jack (see position (3) in section *Device overview* [▶ page 15]) of the device).  
When inserting the plug into the jack, turn it clockwise to engage its “bajonet” locking mechanism, thus attaching the plug securely to the device.

---

### NOTICE

#### Device Destruction by Exceeding the Allowed Supply Voltage!

The supply voltage must not exceed +40 V DC, otherwise the device will be damaged.

---

### Switching-on the device

After connecting the supply voltage, switch on the device by pressing the power button (see position (11) in section *Device overview* [▶ page 15]) for approx. one second.

## 4.3 Establish LAN connection and login to Local Device Manager

### 4.3.1 Overview

You have two possibilities to establish an initial LAN connection with the **Local Device Manager** (which is the web-based management GUI of the device):

- Via DHCP at ETH 1 port (`eth0` in netFIELD OS)  
(“Fallback” of ETH 1 is *IPv4 link local*)
- Via direct (one-to-one) connection at ETH 2 port (`eth1` in netFIELD OS)

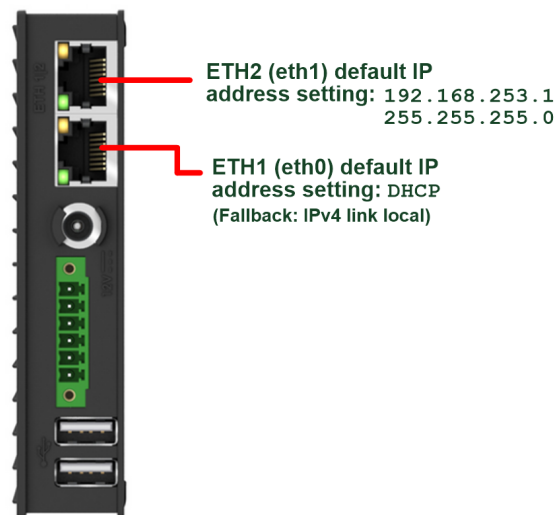


Figure 12: Factory IP address settings of LAN interfaces



#### Note:

The device contains a certificate issued by Hilscher. Therefore, your browser will probably issue an "unsecure connection" warning message when connecting to the device for the first time.

You can ignore the warning and – depending on your browser model – select the option to continue to the device's website anyway (respectively add an "exception rule" for this website).

On the **Certificate** page of the device's **Local Device Manager**, you can upload your own certificate to the device.

Note that the automatically created certificate is valid for one year.

On the **Certificate** page of the **Local Device Manager**, you can upload your own certificate to the netFIELD OS. The corresponding root certificate should be rolled out on each of your PC/devices that you use for connecting to the netFIELD OS.

## 4.3.2 Using DHCP server

In the device's state of delivery, port ETH 1 (`eth0` in netFIELD OS) is set to DHCP mode.

If a DHCP server is available in your local IT network, you can thus use the DHCP service to assign an IP address to the ETH1 interface.



---

**Note:**

If the device realizes that no DHCP service is available, it resets the ETH 1 interface address to *IPv4 link local* mode ("fallback" setting). *IPv4 link local* uses the address range from `169.254.0.0` to `169.254.255.255`.

The device outputs the *IPv4 link local* address at its **Console** interface, thus allowing you to connect an engineering PC/notebook and use a terminal program like e.g. PuTTY to find out the exact address. For more information on the Console interface, see section *Console interface* [▶ page 20]

---

- Make sure that a DHCP service is available in your local network.
- Plug an Ethernet cable into the ETH 1 port on the front panel of the device (see position (2) in section *Device overview* [▶ page 15]), to connect it to your local network and to the DHCP server.
- ↻ Your device should now automatically obtain an IP address from the DHCP server. This may take a few minutes.  
If you know the IP address that the DHCP server has assigned to your device, you can now access the **Local Device Manager** directly by entering the assigned IP address into the address bar of your web browser. If you do not know the IP address, you can use the Windows network environment (see "Alternative A" below) or the "host name" of the device (see "Alternative B" below) to connect with it.



---

**Note:**

The device outputs its hostname and the IP address (which it has received from the DHCP server) at its **Console** interface. Thus, connecting to the console allows you to check the assigned IP address.

In case no DHCP service is available, the "fallback" IPv4 link local address of the ETH 1 interface (`eth0` in netFIELD OS) will also be output at the console.

---

- Enter into the address bar of your browser the IP address that the DHCP server has assigned to the device.
- ↻ Your browser connects to the **Local Device Manager**, which is the graphical user interface of the device.



---

**Note:**

Your browser will issue an "unsecure connection" warning message.

You can ignore the warning and – depending on your browser – select the option to continue to the device's website anyway (respectively add an "exception rule" for this website).

On the **Certificate** page of the Local Device Manager, you can later upload your own trusted certificate to the device.

---

### Alternative A: Connecting via Windows network environment

Because the device supports the UPnP technology (Universal Plug and Play), it will be displayed in the **Windows** network environment panel after having received its IP address from the DHCP server. This allows you to connect to it by simple mouse-click.



#### Note:

Make sure the network discovery feature on your Windows PC is enabled for your LAN security zone. Note also that your engineering PC and the device must be located in the same subnet.

- To display all devices in the network, open your **Windows Explorer** and select **Network**.
- You will find the device listed under **Other Devices**. You can recognize it by its model name followed by its host name in brackets:



netfield-compact-x8m-rev1  
(nt0001c02e1f11)

- Double-click this entry to connect to the Local Device Manager of the device.

### Alternative B: Connecting via host name

- As a second alternative, you can also connect to the **Local Device Manager** by entering the device's host name into the address bar of your browser. You will find the host name printed on the device label next to **DHCP**, as shown in this example:

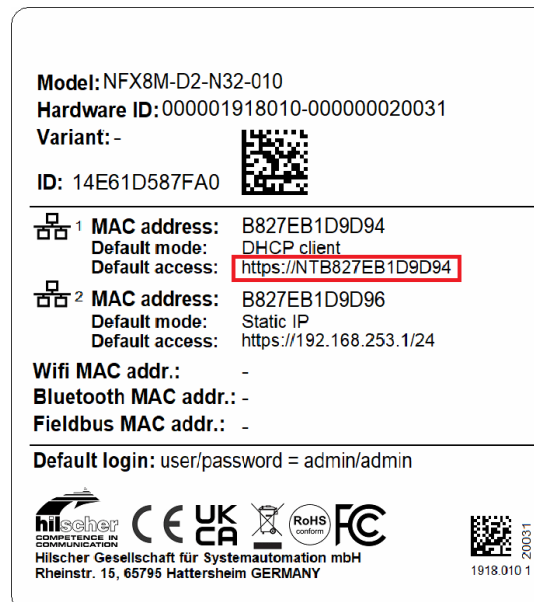


Figure 13: Host name on device label (example)



#### Note:

Your PC and your device must be located in the same subnet.

### 4.3.3 Establishing one-to-one connection to device (without DHCP server)

If no DHCP server is available in your network, you can connect your engineering PC or notebook via Ethernet cable directly to port ETH 2 (`eth1` in netFIELD OS) of the device (see position (1) in section *Device overview* [▶ page 15]).

For this, you must set an IP address on your PC or notebook that is compatible with the preset IP address and subnet mask of port ETH 2. In the device's state of delivery, the IP address of port ETH 2 is preset to `192.168.253.1`, the subnet mask is `255.255.255.0`.

1. Connect Ethernet cable.
  - Use an Ethernet cable to connect the ETH 2 interface directly to your PC or notebook:
2. Set IP address on your PC or notebook (under Microsoft Windows).
  - Open the Windows **Control Panel**. (**Start** menu > **Windows System** > **Control Panel**)
  - In the **Control panel**, select **Network and Internet**, then **Network and Sharing Center**.
  - In the **Network and Sharing Center**, select **Change adapter settings**.
  - In the **Network Connections** window, double-click the name of your direct network connection, e.g. **Local Area Connection** or **Ethernet**. (The name of the network connection may be different on your PC.)
  - In the **General** dialog window, click **Properties**.
  - In the **Networking** tab of the **Properties** dialog window, double-click **Internet Protocol Version 4 (TCP/IPv4)**
  - In the **General** tab, set IP address `192.168.253.2` and subnet mask `255.255.255.0`.

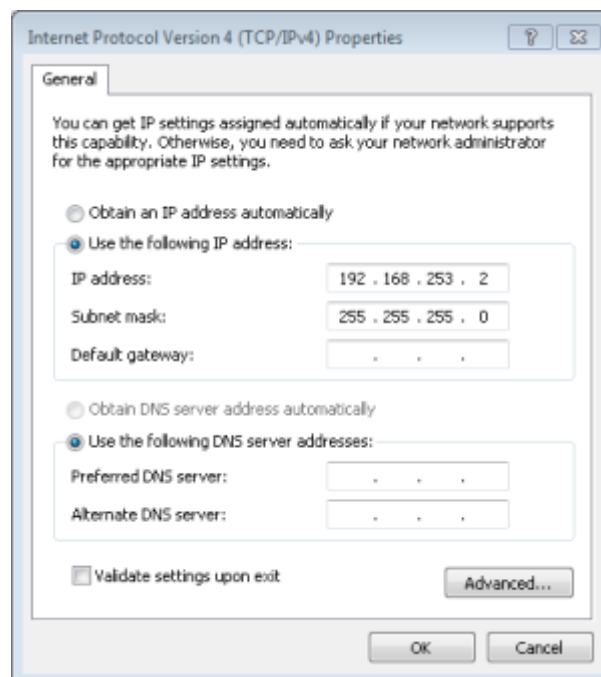


Figure 14: Setting IP address under Windows for direct LAN connection

- Click **OK** and then **Close**.

3. Open browser and connect to device.
  - You can now access the device from your PC or notebook via web browser by entering the following address into the address bar of your browser:  
`https://192.168.253.1`
  - A connection is established and the Local Device Manager opens in your browser window.

#### 4.3.4 Login to Local Device Manager

**Note:**

When connecting to the device for the first time, your browser will issue a security warning before displaying the Login screen of the Local Device Manager.

You can ignore the warning and – depending on your browser model – select the option to continue to the device’s website anyway (respectively add an “exception rule” for this website).

On the **Certificate** page of the **Local Device Manager**, you can upload your own trusted certificate to the netFIELD Compact device. The corresponding root certificate should be rolled out on each of your PC/devices that you use for connecting to the netFIELD Compact device.

After having established a LAN connection to the device, the **Sign In** dialog of the **Local Device Manager** appears:

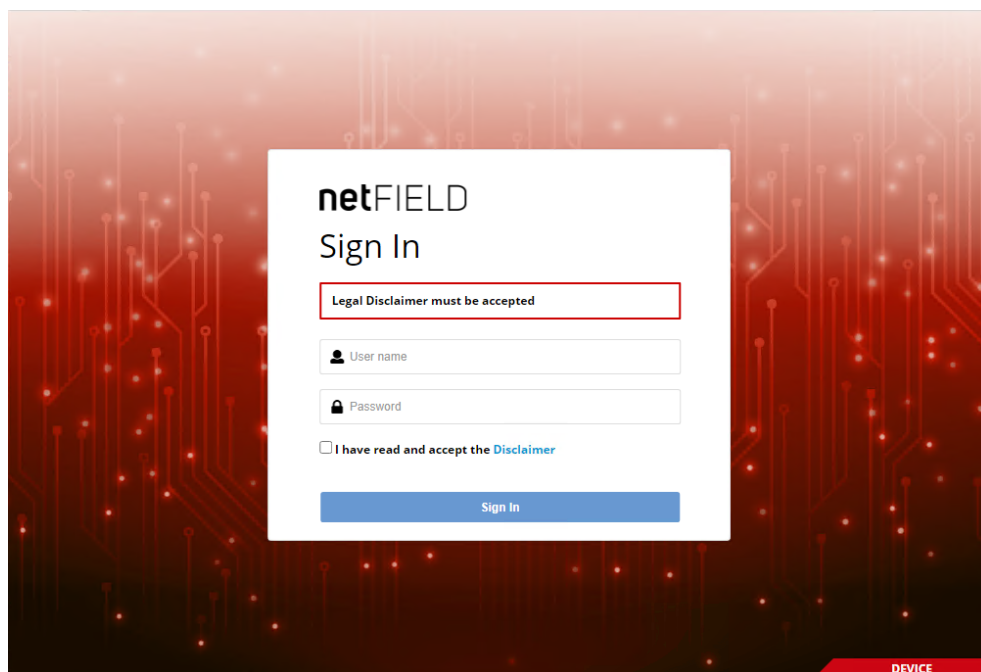


Figure 15: Login Device Manager

- In the **Sign In** dialog, enter the following default credentials:  
**User name:** admin  
**Password:** admin
- Read the **Disclaimer** then check the **I have read and accept the Disclaimer** box.



- Click **Sign In** button.
- For security reasons, you are now forced to change the default `admin` password immediately.
- In the **Current password** field, enter `admin` once again, then click **Sign In** button:

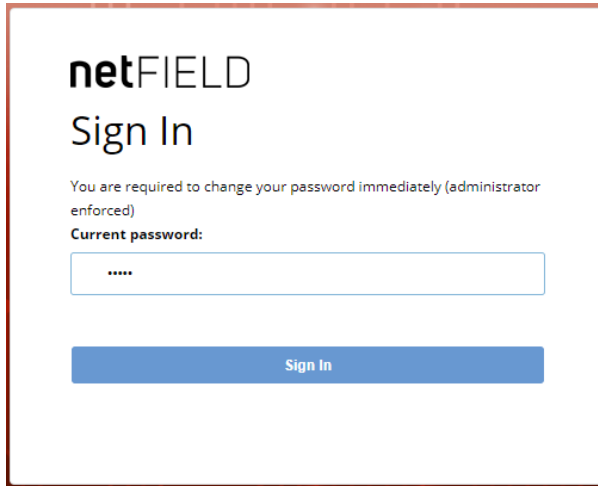
The screenshot shows the netFIELD Sign In interface. At the top, the text 'netFIELD Sign In' is displayed. Below this, a message states: 'You are required to change your password immediately (administrator enforced)'. Underneath, the label 'Current password:' is followed by a text input field containing five asterisks. A blue 'Sign In' button is positioned at the bottom of the form.

Figure 16: Enter current password dialog

- The **New password** dialog opens:



The screenshot shows the netFIELD Sign In interface. At the top, the text 'netFIELD Sign In' is displayed. Below this, the label 'New password:' is followed by a text input field containing seven asterisks. A blue 'Sign In' button is positioned at the bottom of the form.

Figure 17: Enter new password dialog

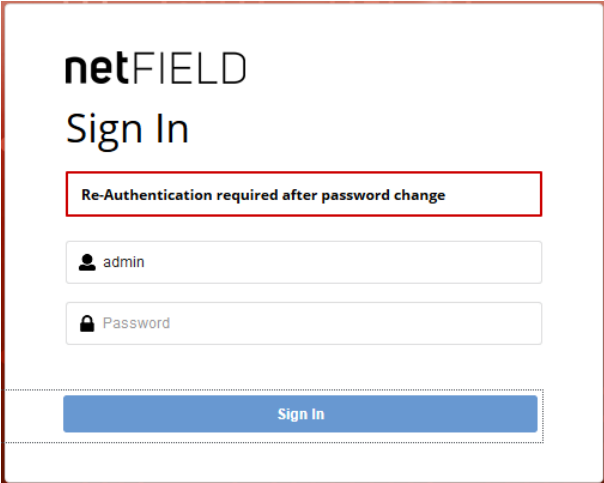
- In the **New password** field, enter a new and safe password, then click **Sign In** button.  
Enter your new password again in the **Retype new password** field, then click **Sign In** button again.



**Note:**

You can change the password again later in the **Local Device Manager** under **Accounts > System Administrator > Set Password** or under  (user menu) > **Account Settings**.

- The **Re-Authentication required after password change** dialog opens:



The screenshot shows a web interface for netFIELD. At the top left, the text "netFIELD" is displayed in a bold, sans-serif font. Below it, the text "Sign In" is displayed in a larger, bold, sans-serif font. A red rectangular box highlights a message that reads "Re-Authentication required after password change". Below this message, there are two input fields: the first is labeled "admin" and the second is labeled "Password". At the bottom of the form, there is a blue button with the text "Sign In".

Figure 18: Re-Authentication dialog

- Enter your new password once again, then click **Sign In** button
- The **Local Device Manager** opens.

## 4.4 Set system time

In the state of delivery of the device, the **Time Zone** of the system is set to **UTC** and the synchronization method (**Set Time**) to **Automatically using NTP (Network Time Protocol service)**.

- To configure your local system time, open the **System** page of the **Local Device Manager**, then click the red date/time value next to **System Time**:

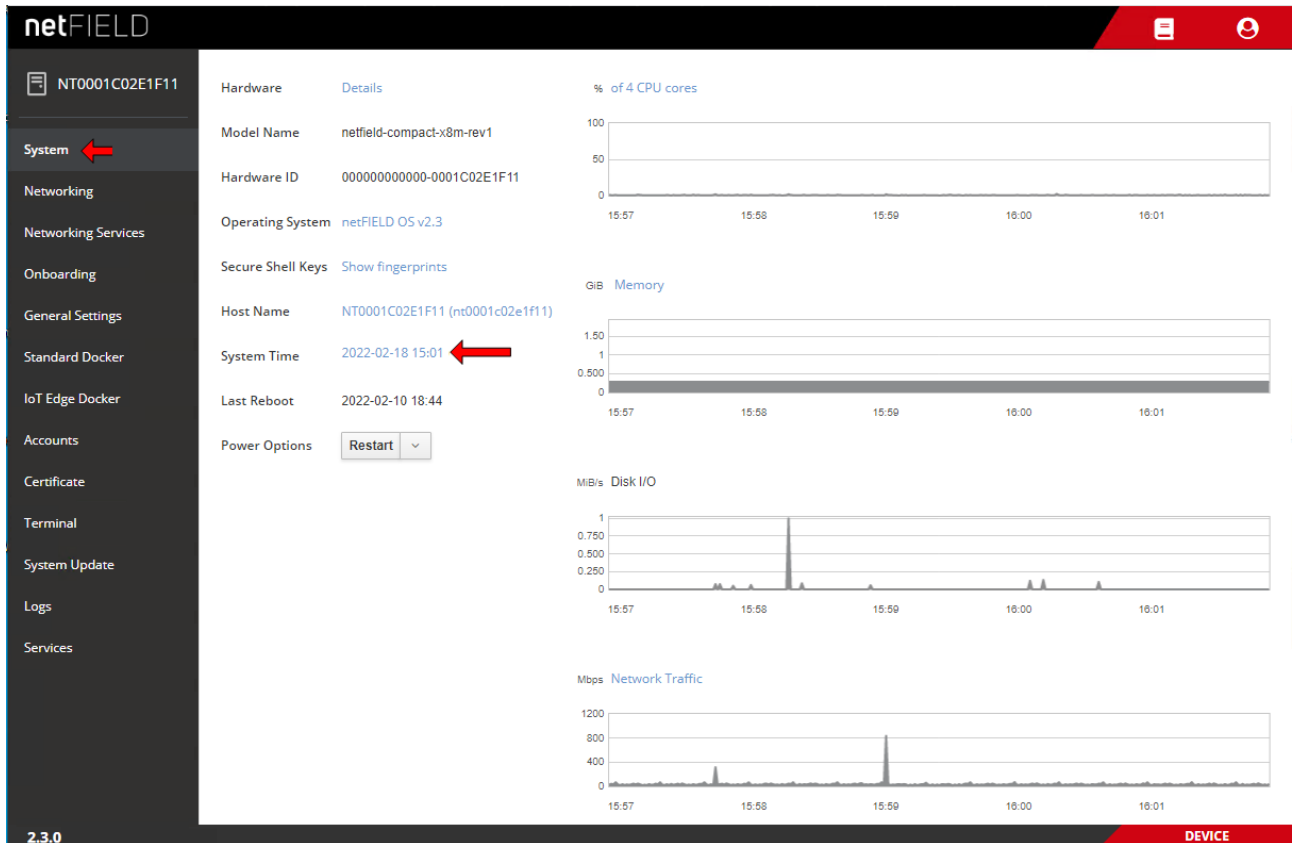


Figure 19: System time value

- The **Change System Time** dialog opens:

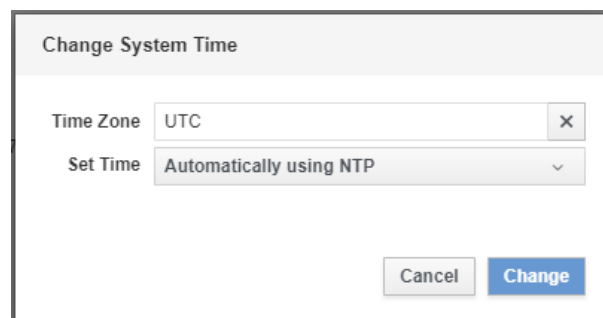


Figure 20: Change System Time dialog

- Click **x** button next to **Time Zone** field to delete the preset **UTC** value, then open the drop-down list and select the appropriate time zone region for your location (note that the list is searchable).

- To choose the synchronization method, choose one of the following options from the **Set Time** drop-down list:
  - **Manually:** Opens further fields for manually entering current date (yyyy-mm-dd) and time (hh:mm). Synchronization via NTP service will not be used.
  - **Automatically using NTP:** The system uses any available NTP server to obtain the correct time. (pool.ntp.org will be used by default).
  - **Automatically using specific NTP servers:** Opens further fields for entering the addresses of certain NTP servers that you want to use, e.g. ptbtime1.ptb.de.  
You can create a list of several servers; the system will use the first server in the list that delivers a valid response. Click the **+** button to add a server. Click the **x** button to remove a server.

The screenshot shows a dialog box titled "Change System Time". It features a "Time Zone" dropdown menu currently set to "Europe/Berlin". Below this is a "Set Time" dropdown menu set to "Automatically using specific NTP servers". Underneath, there are two input fields for NTP server addresses: "ptbtime1.ptb.de" and "ntp.uni-regensburg.de". Each input field has a small "x" button to its right for removal and a small "+" button to its right for adding more servers. At the bottom of the dialog are two buttons: "Cancel" and "Change".

- Click **Change** button to save the new settings and close the dialog window.
- To update the display of the system time (to adapt it to the changed time zone), refresh the web page by pressing the **F5** key on your keyboard.

## 4.5 "Onboard" (register) device in netFIELD Cloud

### 4.5.1 Overview

If you connect your device via Internet to the netFIELD Cloud (<https://www.netfield.io>), you can install containers and manage your device from the netFIELD Portal, which is the web-based user interface of the netFIELD Cloud. You can also stream MQTT messages from your device to 3rd party applications via the RESTful API of the netFIELD Platform, which is the backend of the netFIELD Cloud.

This section describes how to register your device in the netFIELD Portal.



---

**Note:**

Connecting your device to the netFIELD Cloud requires an account/subscription for the *netFIELD Cloud services*

<https://www.netfield.io>.

Contact your local Hilscher sales representative for information on terms and conditions.

---

Before your device can be managed from the portal, it must first complete a one-time registration process, called "onboarding".

This process is initialized by the device itself, not by the portal. There are three different onboarding methods: **Zero-Touch**, **Basic** and **Advanced**.

With the **Zero-Touch** method, the device registers itself automatically in the portal after it has been put into operation. Note that this method is implemented only in certain customer-specific Edge Device models.

With the **Basic** and **Advanced** methods, you start the registration process by locally entering authentication data in the **Onboarding** page of the **Local Device Manager**:

With the **Basic** method, you simply need to enter your netFIELD Portal's login credentials (if your user "role" in the portal entails permissions to "onboard" and "create" devices).

With the **Advanced** method (which allows onboarding in a certain separate instance of the netFIELD Portal), you must enter an `Activation Code`, an `API Key` and an `API End-Point URL`. You must research (respectively create) these parameters in the portal beforehand, then insert them in the **Onboarding** page of the Local Device Manager via clipboard ("copy and paste"). For the **Advanced** method, you therefore ideally need simultaneous access to the portal and the device in order to be able to copy the data from the portal conveniently into the corresponding fields of the **Onboarding** page of the Local Device Manager.

**Note:**

Before onboarding, make sure that your company's firewall does not block the TCP port (outgoing) of the upstream protocol (device-to-cloud communication) that you intend to use. The upstream protocol can be selected on the **Onboarding** page.

MQTT uses TCP port 8883

MQTT over WebSocket uses TCP port 443

AMQP (default protocol) uses TCP port 5671

AMQP over WebSocket uses TCP port 443

The following sections contain step-by-step instructions for the **Basic** and **Advanced** onboarding methods.

## 4.5.2 Onboarding using the “Basic” method

- In the navigation panel of the **Local Device Manager**, choose **Onboarding**.
- The **Onboarding** page opens:

The screenshot shows the netFIELD Local Device Manager interface. On the left is a dark sidebar with a menu. The 'Onboarding' option is highlighted with a red arrow. The main area shows the 'Basic' onboarding screen for device NT0001C02E1F11. It includes a 'Manual' onboarding method, a 'Hardware Id' of 000000000000-0001C02E1F11, and an empty 'Environment' field. Below are input fields for 'Device Name', 'E-Mail', and 'Password', and a dropdown for 'Upstream Protocol'. A checkbox for 'Use Deployment Manifest' is present and unchecked. An 'Onboard' button is at the bottom left of the form area. The bottom of the screen has a red bar with the word 'DEVICE' on the right and the version '2.3.0' on the left.

Figure 21: “Basic” onboarding screen in Local Device Manager

- Open the **Basic** tab.
- In the **Environment** drop-down list, select the portal's environment that you are using. Usually, this would be the `Production` environment.
- In the **Device Name** field, enter the name under which the device shall be displayed in the portal.
- In the **E-Mail** and **Password** fields, enter the credentials of a user of the **portal** who possesses `createDevices` and `onboardedDevices` permissions.

**Note:**

With these credentials (and the associated permissions), the device authenticates itself during onboarding in the portal and is automatically assigned to the organization or sub-organization of the user.

Ask your portal's system administrator for the necessary credentials.

- In the **Upstream Protocol** drop-down list, select the protocol that the netFIELD OS shall use for sending data to the netFIELD Cloud ("device-to-cloud" communication).

**Note:**

Note that messaging over WebSocket causes more "overhead" per telegram. This might limit the performance if you want to stream large quantities of data.

- **MQTT** – Uses TCP port 8883
- **AMQP** – Default protocol (most commonly used). Uses TCP port 5671
- **MQTTWS** – MQTT over WebSocket. Uses TCP port 443 (same as HTTPS)
- **AMQPWS** – AMQP over WebSocket. Uses TCP port 443 (same as HTTPS)

**Important:**

Make sure that your company's firewall does not block the TCP port (outgoing) of the selected upstream protocol.

**Note:**

If necessary, you can change the upstream protocol in the netFIELD Portal after onboarding. See section *Device Navigation: Edit device settings (Update mask)* in the operating instruction manual *netFIELD Portal*, DOC190701OixxEN.

- In case your organization has a "Deployment Manifest" that you want to use for your device, select the **Use Deployment Manifest** option.

**Note:**

The deployment manifest causes certain software containers defined in the manifest to be automatically installed on your device. (For further information on deployment manifests, see section *Deployment Manifest* in the *netFIELD Portal* manual, DOC190701OixxEN)

- Click **Onboard** button to start the onboarding process.
- ⇒ The device connects to the portal, is registered there and assigned to your organization or sub-organization.  
If the process has been successful, the following message appears:  
**Success – Device is now onboarded.**  
From now on, the device will be listed in the portal's **Device Manager** and can be managed from there.

**Note:**

If the message “Something went wrong – Device has already been created” appears, the device had already been created in the **Device Manager** of the portal for the “Advanced” onboarding method.

In this case you can either use the “Advanced” onboarding method, or you can delete the device in the portal, and then start the “Basic” onboarding procedure here locally for a second time.

### 4.5.3 Onboarding using the “Advanced” method

#### Requirements

- You are logged-in to the Local Device Manager.
- You are also logged-in to the netFIELD Portal.
- You possess the following rights as portal user: `createDevices`, `onboardedDevices` and `getKeys`.

#### Step-by-step instructions

1. Copy **Hardware ID**.
  - In the navigation panel of the **Local Device Manager**, choose **Onboarding**, then open **Advanced** tab:

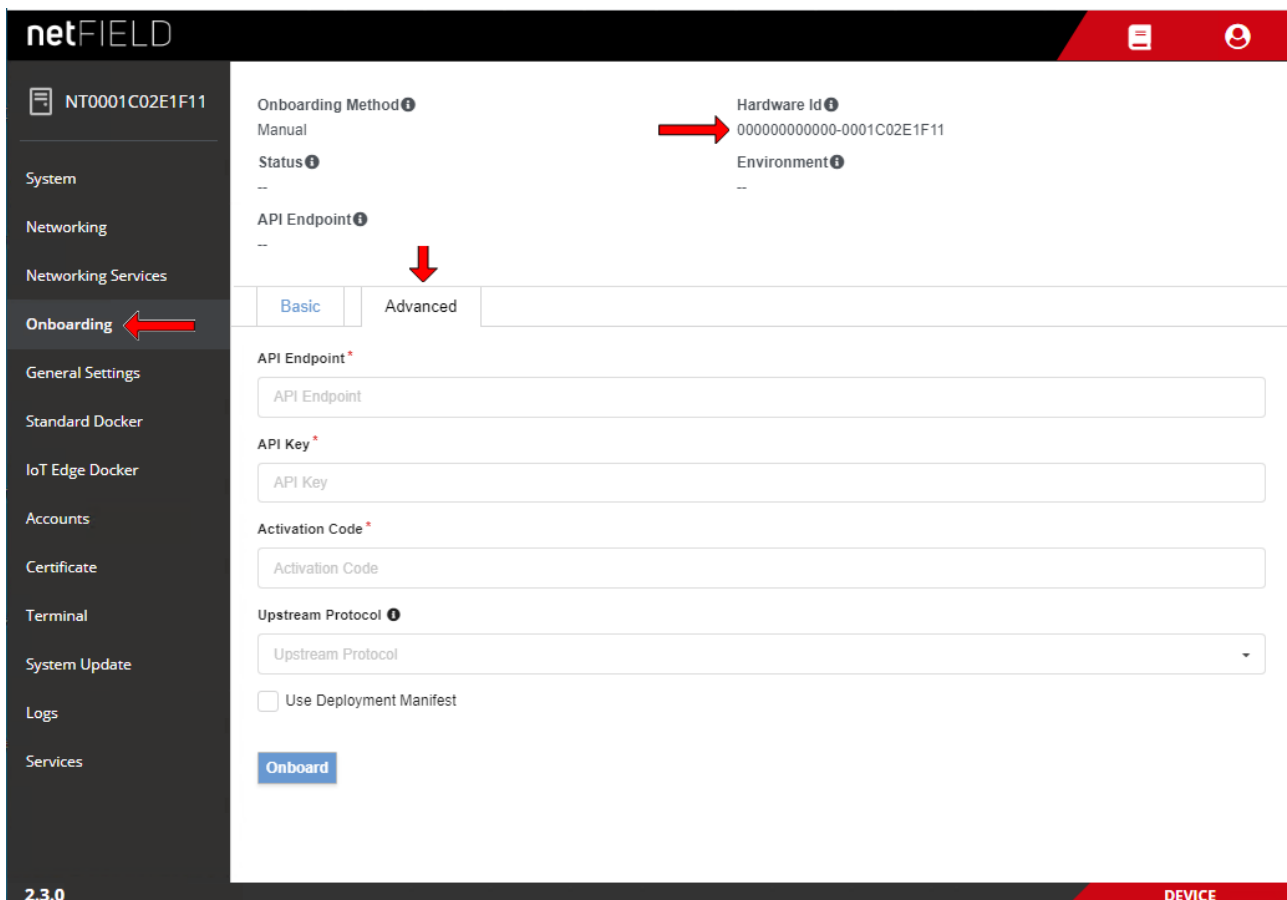


Figure 22: Research Hardware ID

- Select the **Hardware ID** and copy the string to your clipboard.



- Open a new tab in your browser and change to the portal, but do not close the connection to the **Local Device Manager** of your device in your first browser tab.
2. Add the device in the portal and create **Activation Code**.
- In the portal, open the **Device Manager**.
  - On the start page (**Manage your devices**) of the **Device Manager**, select **+ Add** button.
  - The **Add Device** mask opens:

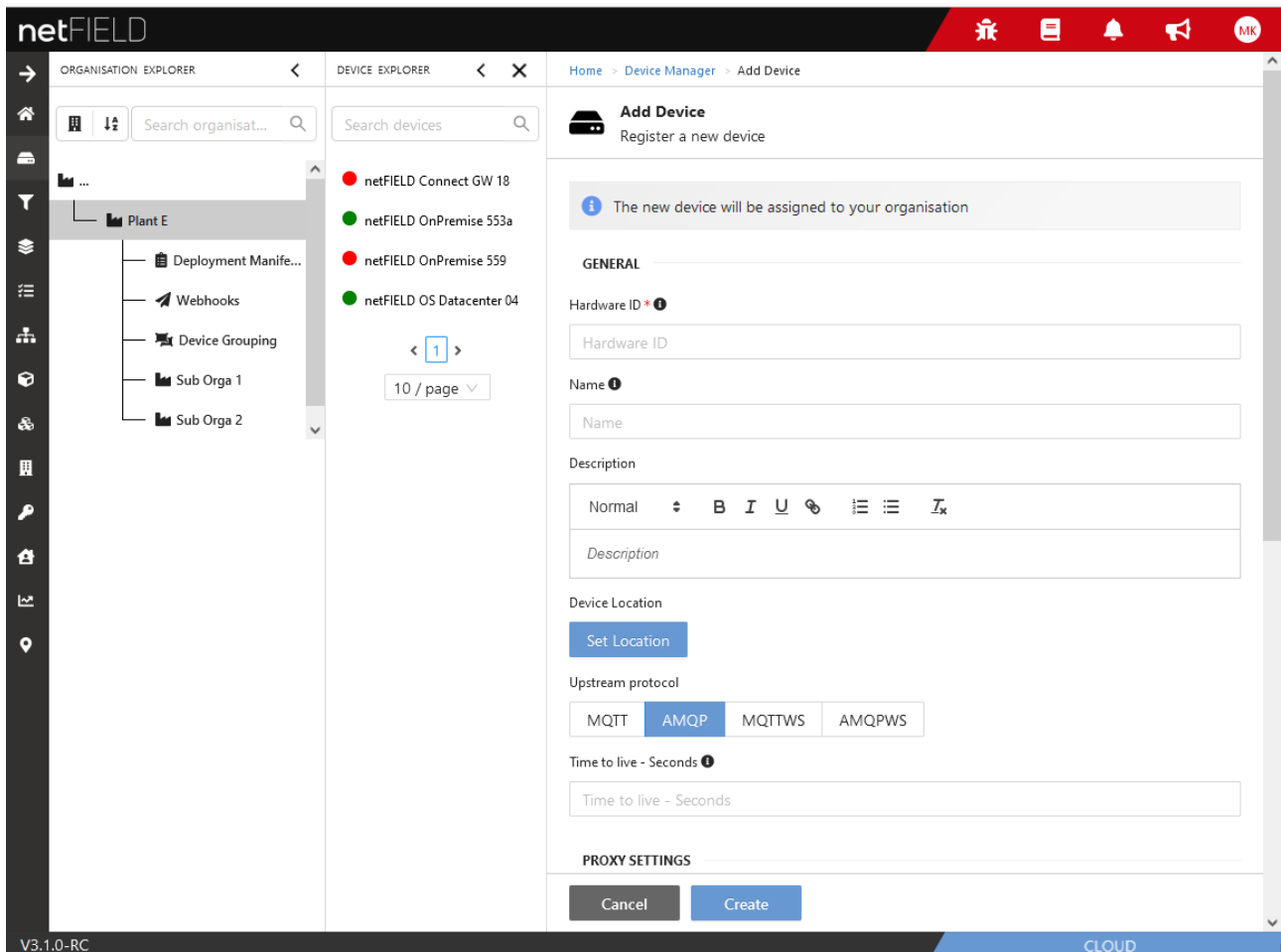


Figure 23: Add device mask in netFIELD Portal

- Copy the device's hardware ID from your clipboard into the **Hardware ID** field.
- In the **Name** field, enter a name for your device (optional but recommended).
- Keep all other parameters at their default settings. If necessary, you can reconfigure these parameters in the Portal later, after onboarding.



For information on how to configure these parameters, see section *Device Navigation: Edit device settings (Update mask)* in the *netFIELD Portal* manual (DOC1907010IxxEN).

- Click **Create** button.

- The mask closes, and the **Overview** page of the newly created device opens, showing the **Activation Code** that you will have to enter locally on your device:

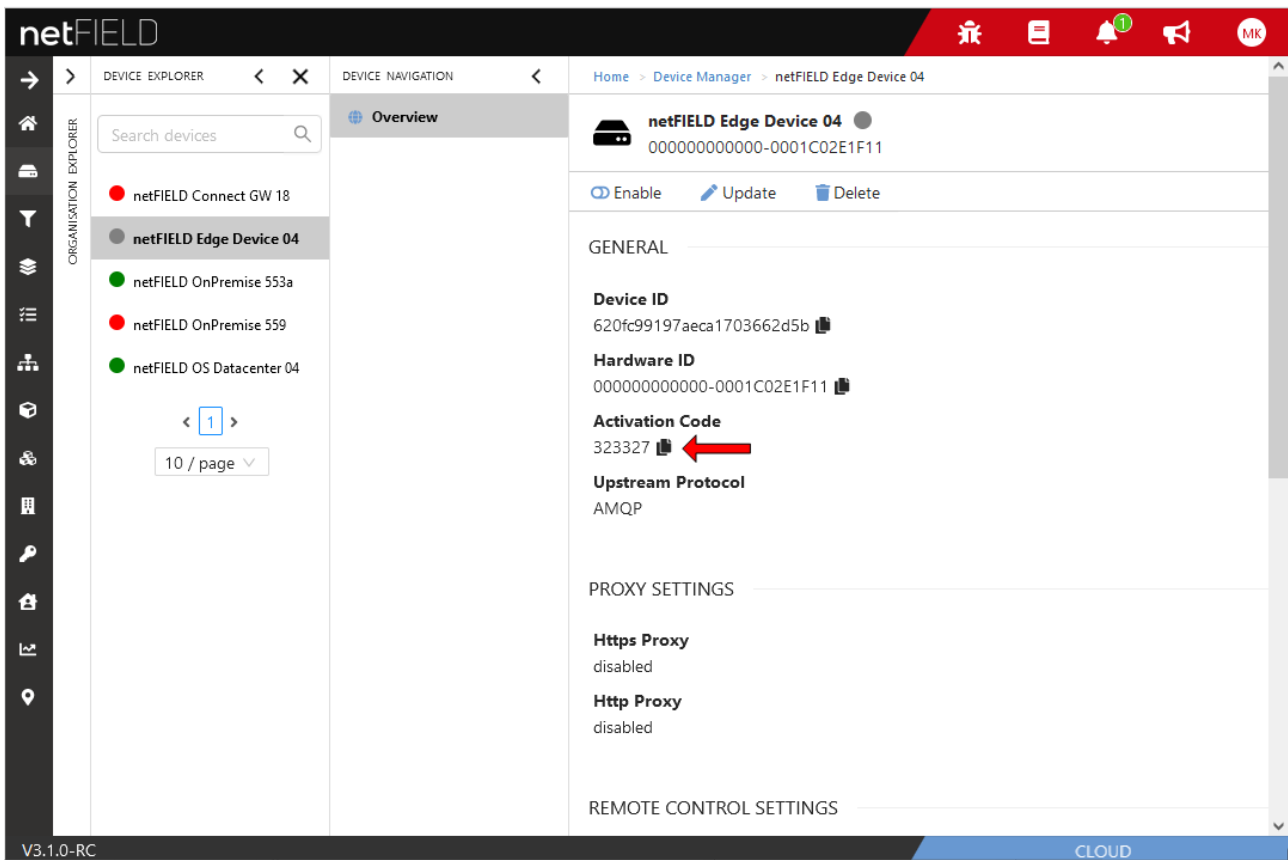


Figure 24: Activation Code in portal

- Copy the **Activation Code** to your clipboard.

3. Enter onboarding parameters in Local Device Manager.
  - Go back to the **Onboarding > Advanced** page in the **Local Device Manager** of your device.

The screenshot shows the netFIELD device's onboarding interface. The top bar displays the device ID 'NT0001C02E1F11'. The sidebar on the left lists various settings categories, with 'Onboarding' selected. The main content area is divided into 'Basic' and 'Advanced' tabs, with 'Advanced' currently active. Fields include:
 

- Onboarding Method:** Manual
- Hardware Id:** 000000000000-0001C02E1F11
- Status:** --
- Environment:** --
- API Endpoint:** api.netfield.io
- API Key:** [Redacted]
- Activation Code:** 323327
- Upstream Protocol:** AMQP
- Use Deployment Manifest

 An 'Onboard' button is visible at the bottom of the form.

Figure 25: Advanced Onboarding tab in device

- In the **API Endpoint** field, enter the URL of the REST-API interface of the portal.  
For the Hilscher *netFIELD Portal*, this is: `api.netfield.io`  
If you are using a different instance of the portal, ask your portal's system administrator for the URL.
- In the **API KEY** field, enter an API Key that possesses the right to onboard devices. (See *Side note: How to copy an API Key for onboarding* below).
- Copy the activation code (which you have created in step 2) into the **Activation Code** field.
- In the **Upstream Protocol** drop-down list, select the protocol that the netFIELD OS shall use for sending data to the netFIELD Cloud ("device-to-cloud" communication).



**Note:**

Note that messaging over WebSocket causes more "overhead" per telegram. This might limit the performance if you want to stream large quantities of data.

- **MQTT** – Uses TCP port 8883
- **AMQP** – Default protocol (most commonly used). Uses TCP port 5671
- **MQTTWS** – MQTT over WebSocket. Uses TCP port 443 (same as HTTPS)
- **AMQPWS** – AMQP over WebSocket. Uses TCP port 443 (same as HTTPS)

**Important:**

Make sure that your company's firewall does not block the TCP port (outgoing) of the selected upstream protocol.

**Note:**

If necessary, you can change the upstream protocol in the netFIELD Portal after onboarding. See section *Device Navigation: Edit device settings (Update mask)* in the operating instruction manual *netFIELD Portal*, DOC190701OIxxEN.

- In case your organization has a "Deployment Manifest" that you want to use with your device, select the **Use Deployment Manifest** option.

**Note:**

The deployment manifest causes certain software containers defined in the manifest to be automatically installed on your device. (For further information about deployment manifests, see section *Deployment Manifest* in the *netFIELD Portal* manual, DOC190701OIxxEN)

- Click **Onboard** button, to start the onboarding process.
- ⇒ The device connects to the portal and is registered there. If the process has been successful, the following message appears: **Success – Device is now onboarded.**

**Side note: How to copy an API Key for onboarding**

For onboarding by "Advanced" method, you need an API Key, which you can copy to your clipboard in the **API Key Manager** of the netFIELD Portal, and then paste into the Local Device Manager of your device during onboarding.

The key must have the permissions (i.e. Security Level **org+ch** or **org**) for the **onboardedDevices** and **createDevices** functions of the **devices** resource of your organization.

You can use an already existing API key (which, for example, was created by the system administrator) or create a new API key yourself.

For information on how to create a new API Key, see section *Create/edit API key* in the *netFIELD Portal* manual, DOC190701OIxxEN.

API Keys are administered in the **API Key Manager** of the portal.

For accessing existing keys in the **API Key Manager**, you must at least have the permission to use the **getKeys** function of the **keys** resource.

For creating a new key, you must have the permission to use the **createKeys** function of the **keys** resource.

- Open the **API Key Manager** in the portal.
- On the start page (**Manage your API Keys**), select from the list a key that allows the **onboardedDevices** function of the **devices** resource.

To find out the permissions of an API Key, click on the key in the list, then open its **Permissions** tab:

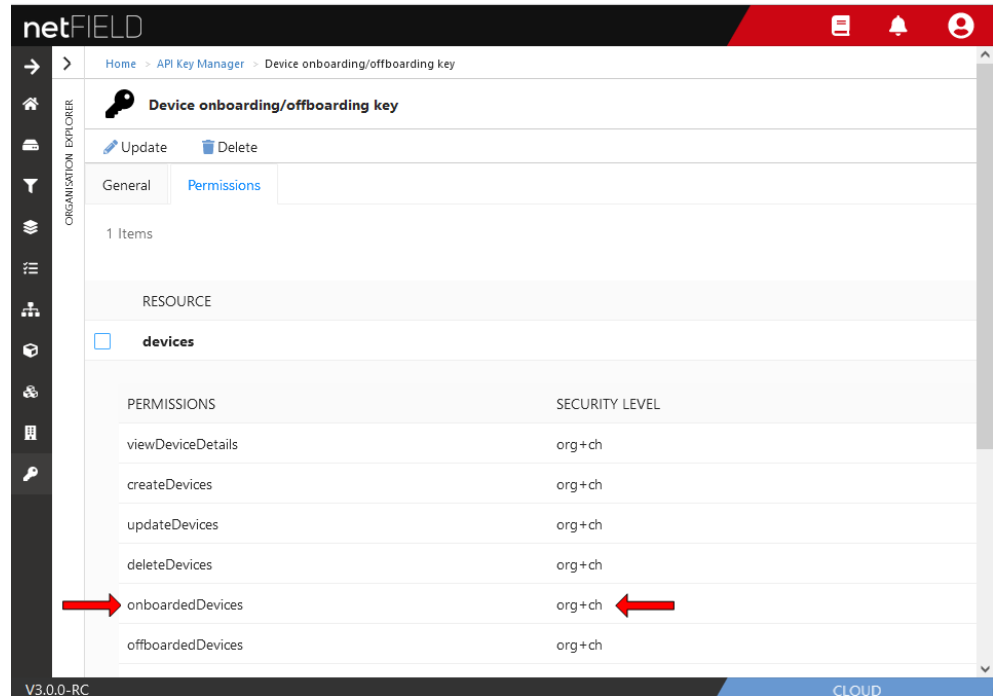



Figure 26: Example of an API Key permitting to onboard devices

- To copy the API Key in order to use it in the Local Device Manager of the device for the advanced onboarding process, change into the **General** tab.

- In the **General** tab, click  icon to copy the key to your clipboard:

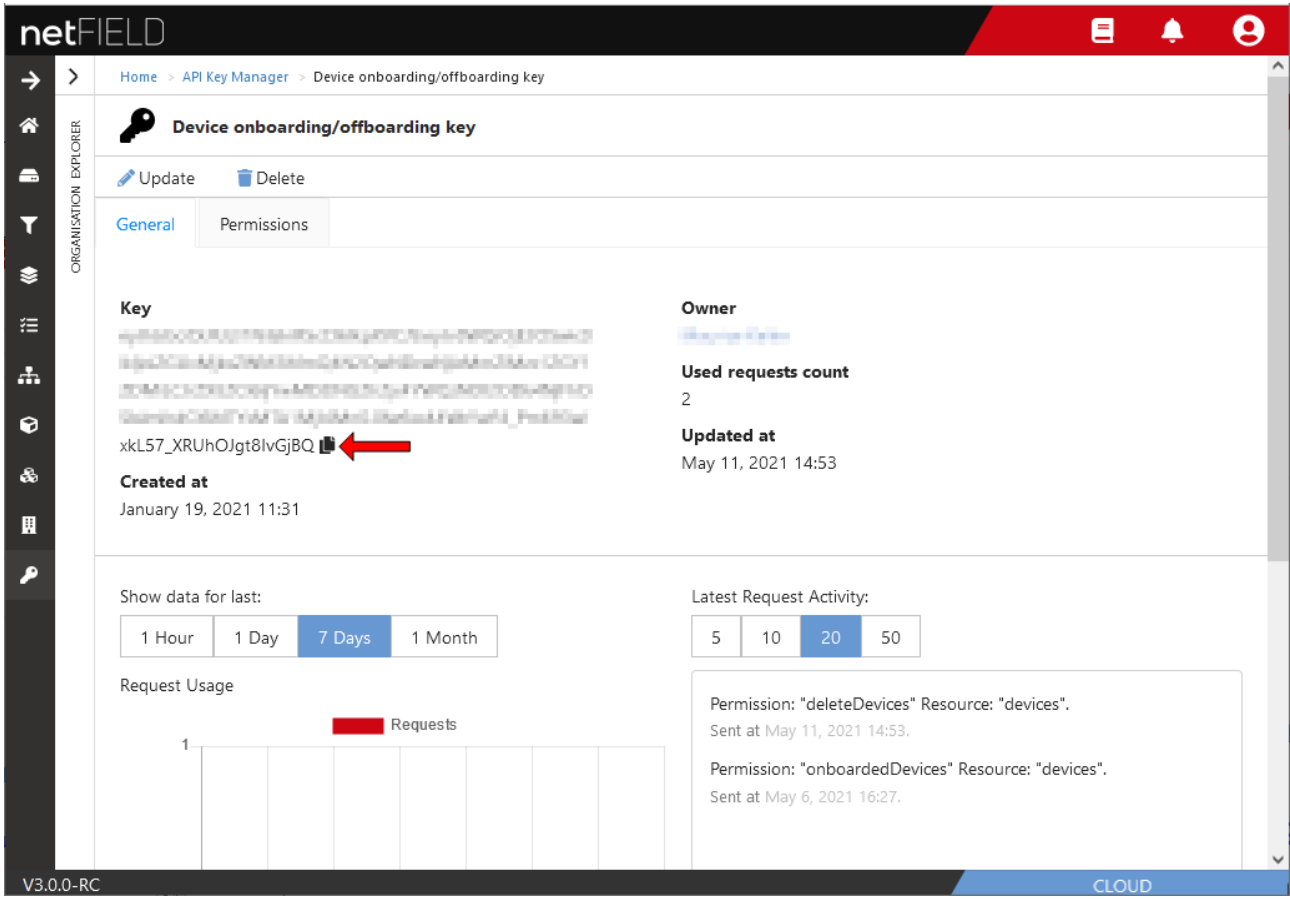


Figure 27: Copy key to clipboard

- Go to the **Onboarding > Advanced** page in the **Local Device Manager** of your local device and insert the key into the **API KEY** field.

## 5 Local Device Manager

### 5.1 Overview

The **Local Device Manager** is the web GUI for configuring and administering the netFIELD OS of the device. It is a customized version of the *Cockpit* web administration console for Linux server.

#### Description of the GUI

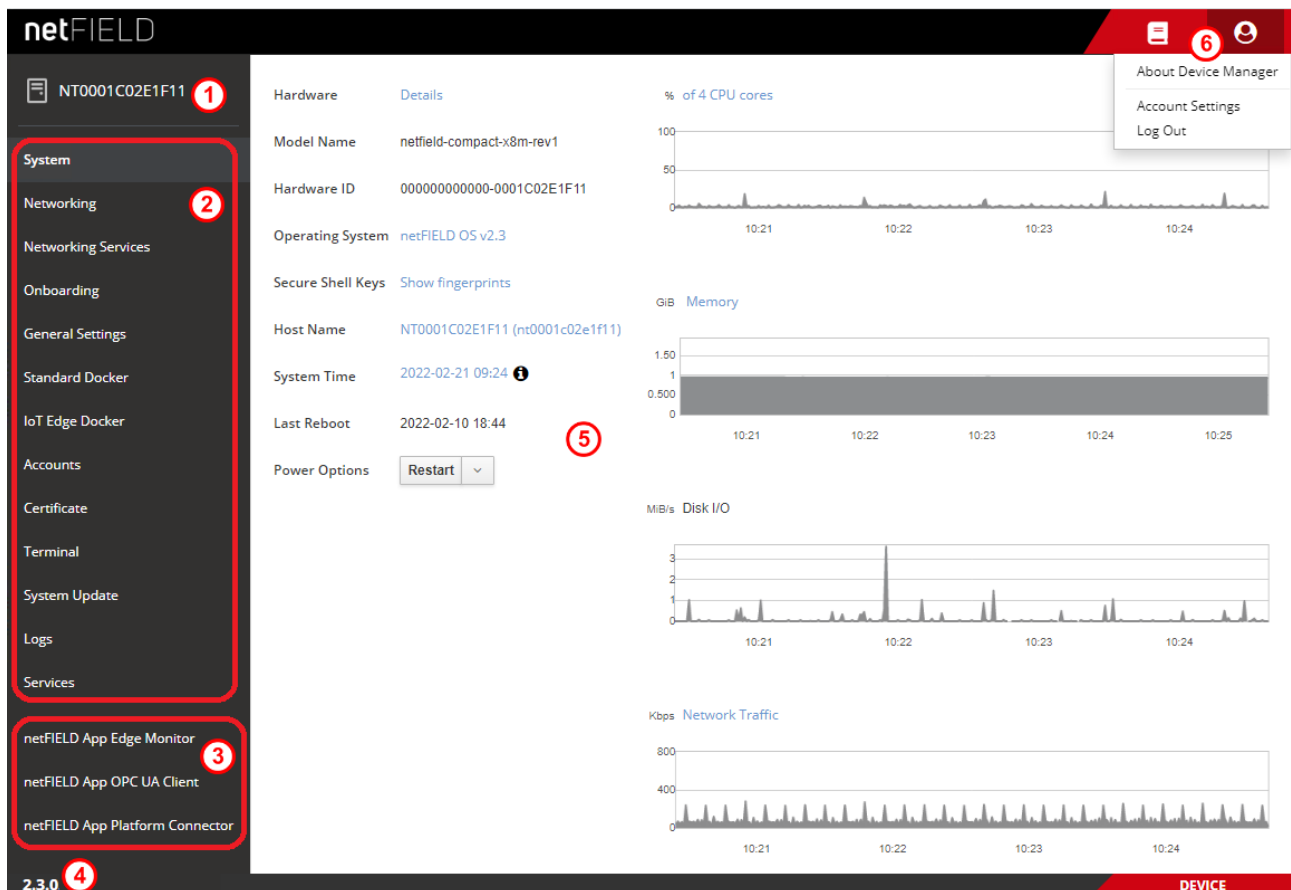




Figure 28: Overview Local Device Manager

- (1) “Pretty” host name of the device (can be adapted by the user, see subsection *Host Name* in section *System* [▶ page 49])
- (2) In the navigation panel on the left of the screen, you can select the available “standard” management pages.
- (3) Many Hilscher netFIELD application containers like e.g. *netFIELD App Platform Connector* or *netFIELD App OPC UA Client* provide their own configuration GUI, which can be selected here (if deployed on your device). Note that the functions and the GUI of individual containers are not described in this manual. Consult the documentation of the individual container for more information.
- (4) Shows the version of the netFIELD OS/Local Device Manager.

(5) Main screen displaying the management page that you have selected in the navigation panel.

Note that if a label, text or value is highlighted in blue, it contains a clickable link that opens a page or dialog box with further details or configuration options.

(6) Toolbar in the upper right corner of the screen:

- The  icon opens a page in the netFIELD Portal where you can find the currently available netFIELD documentation (including this user manual).
- The  icon opens the user menu:
  - **About Device Manager:** Shows information about the Local Device Manager.
  - **Account Settings:** Opens the configuration page of your currently used account (i.e. the account you are currently logged in with). See also *Accounts* [▶ page 101] section for further information.
  - **Log Out:** Logs you out of the Local Device Manager



## 5.2 System

The **System** page allows you to configure and monitor basic system parameters and resources.

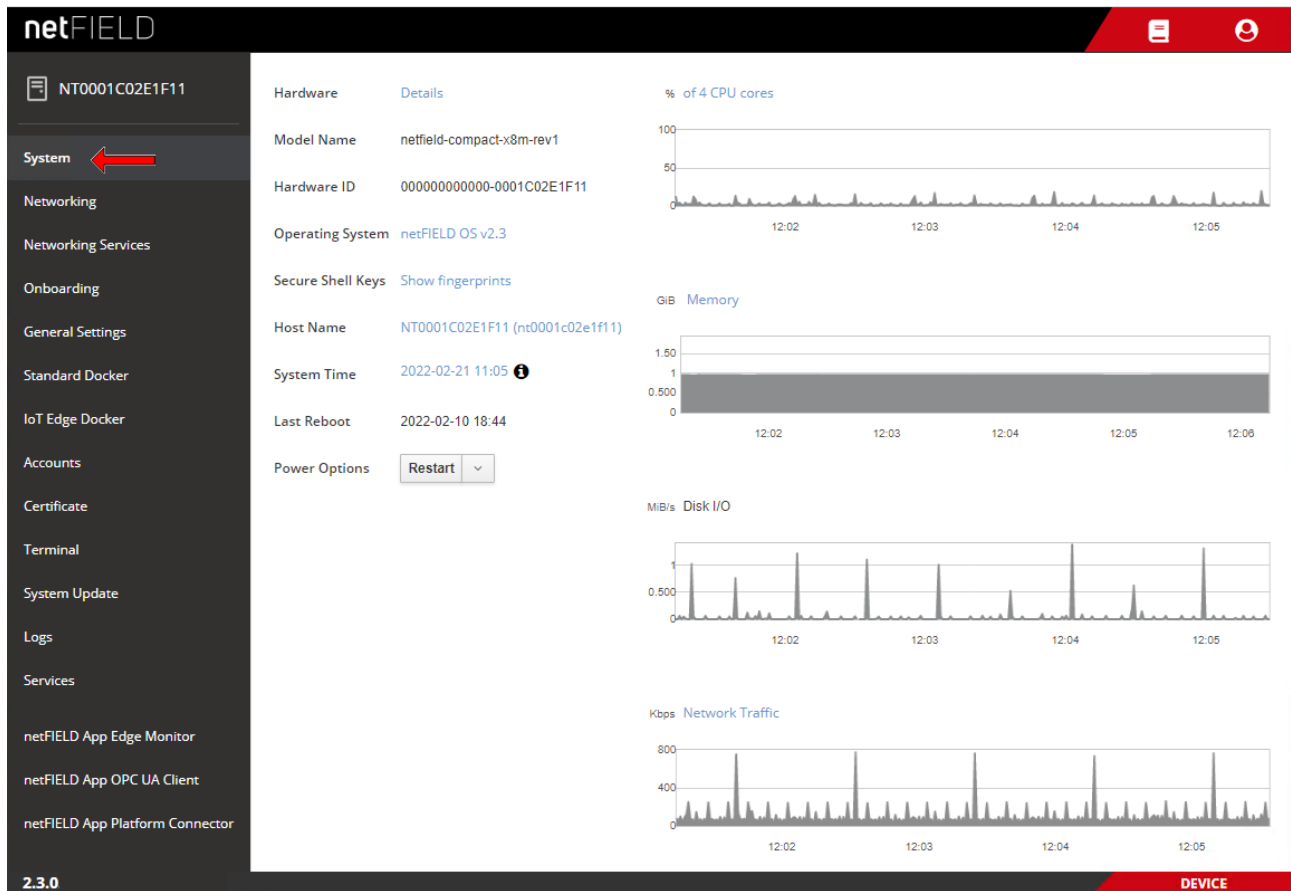


Figure 29: System page in Local Device Manager

### Hardware

Click on **Details** to open a page showing details about your device's hardware like processor cores, RAM, mass storage, CPU temperature etc.

### Model Name

Model name of the device

### Hardware ID

Unique identification number of the device. To match the required format, the ID may be "filled up" with zeros. This ID can also be used in the netFIELD Portal as unique identifier of your device.

### Operating System

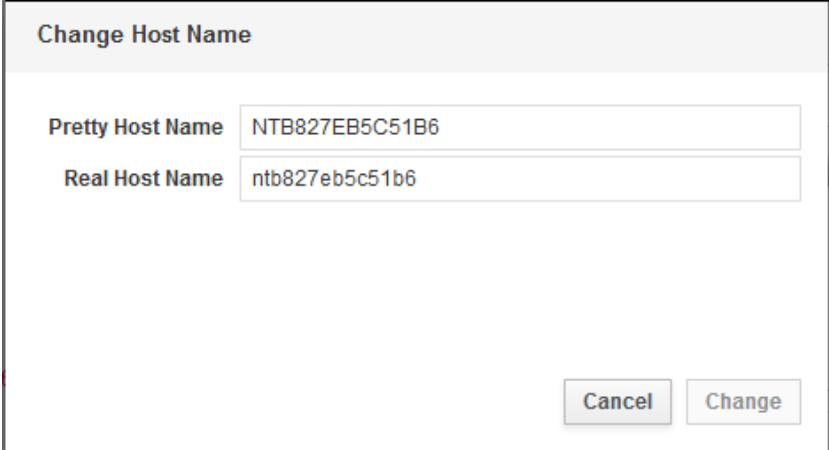
Name and version of the installed netFIELD OS. Click on the blue name to open a window showing further details (i.e. the exact firmware version).

## Secure Shell Keys

Click on **Show fingerprints** to open a window displaying the Machine SSH Key Fingerprints.

## Host Name

The host name identifies the device in a LAN or Wi-Fi network and can be used for connecting to the device. By default, the name consists of the letters `NT` followed by the MAC address of the LAN port of the device. If you want to change it, click on the blue name to open the **Change Host Name** dialog window.



The image shows a dialog window titled "Change Host Name". It contains two text input fields. The first field is labeled "Pretty Host Name" and contains the text "NTB827EB5C51B6". The second field is labeled "Real Host Name" and contains the text "ntb827eb5c51b6". At the bottom right of the dialog, there are two buttons: "Cancel" and "Change".


Figure 30: Change host name dialog

**Pretty Host Name:** Free-text (UTF8) name for presentation to the user. Will be displayed e.g. on top of the navigation panel in the Local Device Manager or as label in your browser tab.

**Real Host Name:** Equivalent to the transient host name which can be used to connect to the device and which can be changed by DHCP or mDNS at runtime. Can contain lower-case characters, digits, dashes and periods (with populated subdomains). Setting this value takes immediate effect and does not require a restart.

## System Time

Shows the system time of the device. By default, the time zone is set to UTC and the actual time is synchronized by an NTP (Network Time

Protocol) service. Hovering over the  icon opens a tooltip displaying details about the current settings, like e.g. the NTP service that was used for the synchronization.

For instructions on how to change the time settings, see section *Set system time* [▶ page 35].

## Last Reboot

Shows date and time of the last reboot (restart) of the netFIELD OS.

## Power Options

Use the drop-down button to restart or shutdown the netFIELD OS and the device.

To restart the device after shutdown, press the power button of the device (see position (11) in section *Device overview* [▶ page 15]).

## CPU cores

The graph shows the combined load of the CPUs of the device during the last five minutes. Click on the blue % **of 4 CPU cores** link to open a page showing the share of certain process categories:

- Nice (`ni`): User space processes that have been “niced” (i.e. “prioritized”).
- User (`us`): User space processes (i.e. applications and processes that do not belong to the kernel processes)
- Kernel (`sy`): Linux kernel processes
- I/O Wait (`wa`): Idle while waiting for an I/O operation to complete

## Memory

The graph shows the usage of the RAM memory of the netFIELD OS during the last five minutes. Click on the blue **Memory** link to open a page showing actually used memory and cached memory.

## Disk I/O

The graph shows the data access rate to the mass storage drive/disk/device during the last five minutes.

## Network Traffic

The graph shows the network traffic rate during the last five minutes. Click on the blue **Network Traffic** link to open the **Networking** page providing further details about the physical and virtual network interfaces of the device.

## 5.3 Networking

### 5.3.1 Overview

The **Networking** page allows you to configure IP parameters and to monitor the amount of traffic of the physical and virtual/logical (i.e. of containers) network interfaces that are managed by the netFIELD OS. You can also configure your firewall and HTTPS/HTTP/FTP Proxy server settings here.

The screenshot shows the netFIELD Networking page with the following sections:

- Traffic Monitoring:** Two line graphs showing network traffic. The left graph is labeled 'Mbps Sending' and the right is 'Kbps Receiving'. Both graphs show data from 14:19 to 14:23.
- Firewall:** A section titled 'Firewall' with a toggle switch set to 'On' and '0 Active Zones' listed below.
- Interfaces:** A table with columns 'Name', 'IP Address', 'Sending', and 'Receiving'. It includes buttons for 'Add Bond', 'Add Bridge', and 'Add VLAN'.
 

Name	IP Address	Sending	Receiving
br-f2387be54c95	10.253.0.1/24	336 bps	224 bps
eth0	10.11.5.28/16	38.0 Kbps	22.6 Kbps
eth1		Not available	
- Unmanaged Interfaces:** A table with columns 'Name', 'IP Address', 'Sending', and 'Receiving'.
 

Name	IP Address	Sending	Receiving
docker0	10.252.254.1/24		
iotedge0	10.252.253.1/24		
sit0			
veth0095a55			
- Network Proxy:** A table with columns 'Proxy', 'Host', and 'Port', and an 'Edit Proxy' button.
- Networking Logs:** A section titled 'Networking Logs' for 'FEBRUARY 21, 2022'.
 

10:18	<info> [1645435128.0204] device (vethecb221f): carrier: link connected	NetworkManager
10:18	<info> [1645435127.2871] manager: (vethecb221f): new Veth device (/org/freedesktop/Network...)	NetworkManager

Figure 31: Networking page


The **Networking** page features the following sections:

#### Sending/Receiving

The graphs in the section on top (1) show the amount of network traffic (sending and receiving) for the last five minutes.

## Firewall

The **Firewall** section (2) shows the number of active firewall zones.

With the  toggle switch, you can deactivate the firewall all together. Click on the blue **Firewall** link to open the firewall configuration page. (See section *Firewall* [▶ page 57] for more details.)

## Interfaces

The **Interfaces** section (3) lists the interfaces that can be managed by the netFIELD OS, and shows their basic parameters (IP address, current volumes of sending and receiving).

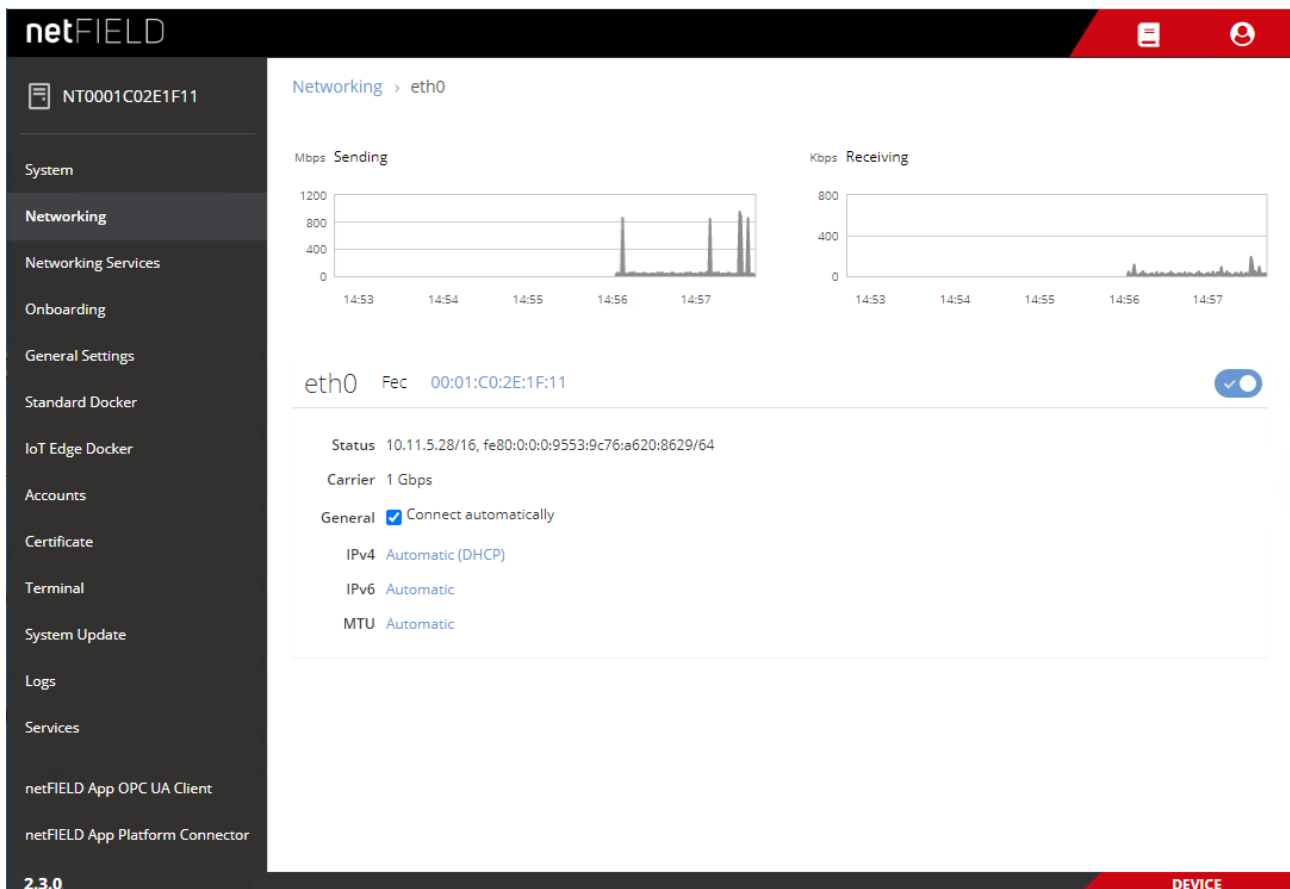
**br-xxxxxxxxxxxx** : This is a “bridge” that was automatically created by the IoT Edge Docker after “onboarding” the device.

**eth0**: This is the ETH 1 LAN interface of the device (for the location of the LAN connector on the device, see position (2) in section *Device overview* [▶ page 15]).

**eth1**: This is the ETH 2 LAN interface of the device (see position (1) in section *Device overview* [▶ page 15]).

### Open details page of Ethernet interface (e.g. for changing IP settings)

- ▶ You can click on an interface, e.g. **eth0**, in order to display further details or to configure its IP settings:




The screenshot displays the netFIELD web interface for the device NT0001C02E1F11. The left sidebar lists various system settings, with 'Networking' selected. The main content area shows the details for the 'eth0' interface. At the top, there are two line graphs: 'Mbps Sending' and 'Kbps Receiving', both showing data activity between 14:53 and 14:57. Below the graphs, the interface name 'eth0' and MAC address 'Fec 00:01:C0:2E:1F:11' are displayed, along with a toggle switch that is currently turned on. The configuration section includes the following details:

- Status: 10.11.5.28/16, fe80:0:0:0:9553:9c76:a620:8629/64
- Carrier: 1 Gbps
- General:  Connect automatically
- IPv4: Automatic (DHCP)
- IPv6: Automatic
- MTU: Automatic

The bottom of the interface shows the version '2.3.0' and the label 'DEVICE'.

Figure 32: Details of LAN interface (eth0)

**Important:**

Be careful not to deactivate the **eth0** and the **eth1** LAN interfaces by switching them off with the  toggle switch. Once you have deactivated an interface, the connection to your device via this interface will be lost. If you have deactivated both LAN interfaces, you will have to perform either a device recovery in order to be able to reconnect again (see section *Device recovery via USB* [▶ page 115]), or you can reactivate the LAN interfaces via the Console interface with a terminal program like e.g. PuTTY (see section *Console interface* [▶ page 20]).

To query the connectivity states of the LAN interfaces via console, use: `sudo nmcli dev status`

To reactivate an interface (e.g. eth0) via console, use:

```
sudo nmcli con up ifname eth0
```

- To change the IP settings, e.g. to set a fixed IP address, click on **Automatic (DHCP)** next to **IPv4**.
- The **IPv4 Settings** page opens.

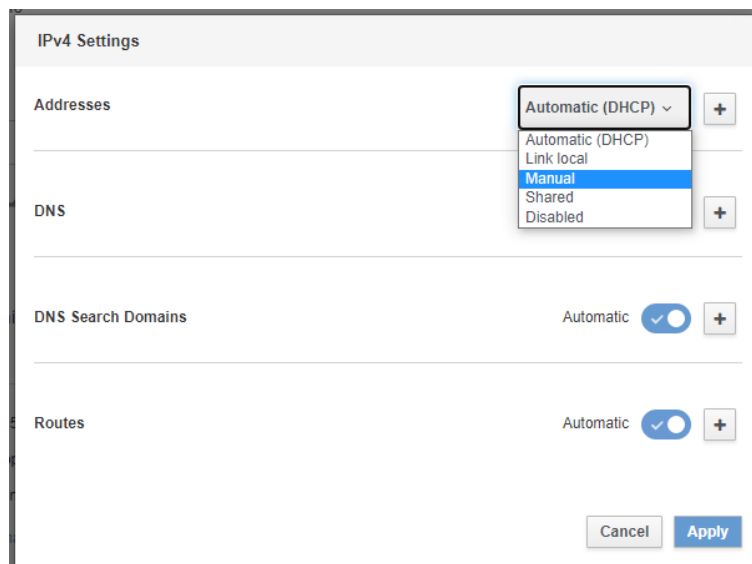


Figure 33: IPv4 Settings

- In the **Addresses** dropdown-list, select **Manual**.

The screenshot shows the 'IPv4 Settings' window. At the top, the 'Addresses' dropdown menu is open and set to 'Manual'. Below this, there are three input fields: 'Address', 'CIDR Suffix or Netmask', and 'Gateway'. Underneath these fields, there are three sections: 'DNS', 'DNS Search Domains', and 'Routes'. Each section has a toggle switch set to 'Automatic' and a '+' button. At the bottom right, there are 'Cancel' and 'Apply' buttons.

Figure 34: Manual IPv4 Settings

- Enter the address parameters, then click **Apply** button.

### Unmanaged Interfaces

The **Unmanaged Interfaces** section (4) lists virtual interfaces and their IP parameters (IP address, current send/receive volumes).

- **docker0**: Virtual interface (“bridge”) of the Standard Docker
- **lotedge0**: Virtual interface (“bridge”) of the IoT Edge Docker
- **vethxxxxxxx**: Virtual interface (“virtual Ethernet device”) of a container in a Docker
- **sit0**: Tunneling protocol (“Simple internet transition”) for using IPv6 over an existing IPv4 connection.



#### Note:

The IP addresses of the “unmanaged interfaces” cannot be changed here. If you want to change the pre-configured IP address of the virtual interface of the Standard Docker (**docker0**) or of the IoT Edge Docker (**lotedge0**), e.g. because it conflicts with other IP addresses in your company network, see section *Docker Network Settings* [▶ page 81] for further information.

## Network Proxy

The Network Proxy section (5) shows the HTTP/HTTPS/FTP proxy server settings of your netFIELD OS. Note that the **No Proxy** URIs `localhost` and `127.0.0.1` are “internal” destinations in the netFIELD OS and are therefore not to be addressed via Proxy server. They appear as **No Proxy** entries by default, even if you did not configure any Proxy server for your netFIELD OS. Do not edit or remove `localhost` and `127.0.0.1` from the **No Proxy** list.

To configure your network Proxy settings, click the **Edit Proxy** button to open the **Proxy Settings** dialog. (See section *Network Proxy settings* [▶ page 66] for more information.)

## NETWORKING LOGS

The **NETWORKING LOGS** section (6) lists messages issued by the Network Manager of the system.



## 5.3.2 Firewall

### Overview

netFIELD OS is equipped with a firewall.

You can add firewall zones and assign interfaces and/or subnets or IP address ranges for which the rules of a zone shall apply. You can also configure “port forwarding” and define allowed services and ports that shall remain “open” in a Drop zone, NAT-Drop zone or Block zone.



#### Important:

Note that in its “state of delivery”, there is no active firewall zone configured, which means that by default, all traffic is allowed and none blocked or dropped until you have configured one or more active zone(s).

---



#### Note:

Be aware that containers running in the Standard Docker or in the IoT Edge Docker may require certain ports on the host system to be “open” in order to function and communicate properly.

Therefore, make sure that you add these ports to the **Allowed Services** list when you define Drop, NAT-Drop or Block zones. The required ports of a container are defined in its *Container Create Options*.

For example, the *mosquitto* container (which is an MQTT Broker) requires the TCP port 1883 for its `mqtt` service to be open.

To find out the services/ports that your containers use, go to the **Standard Docker** page respectively **IoT Edge Docker** page of the Local Device Manager and check out the container’s port settings by clicking on the corresponding image or container instance.

---

- To open the Firewall configuration page, click the **FIREWALL** link on the **Networking** page.

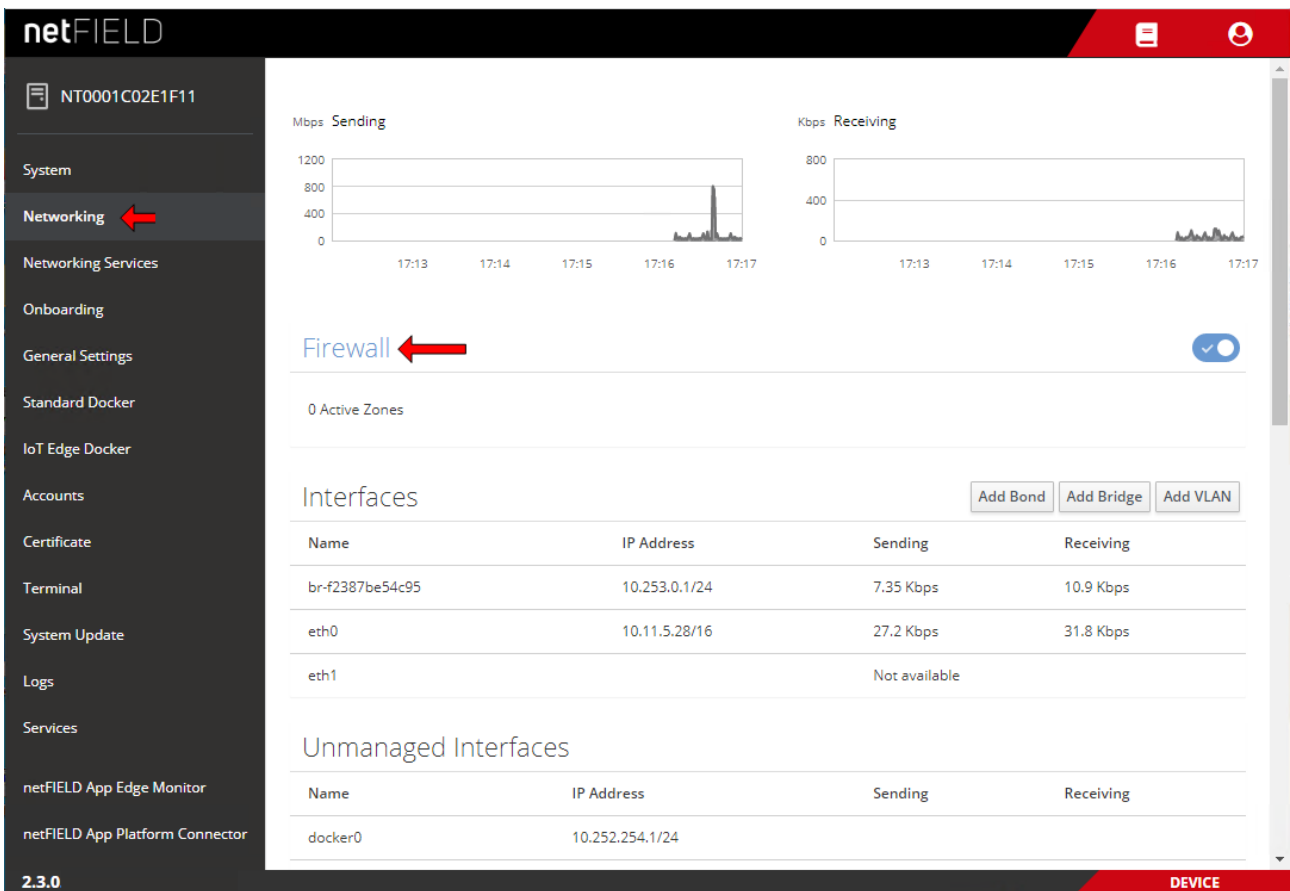


Figure 35: Open Firewall configuration page

➤ The Firewall configuration page opens:

netFIELD

NT0002A233E553

Networking > Firewall

7 [Save Permanent] + Add Zone

1 NAT-Drop Zone

2 Description: The source IP address of all outgoing IP packets is replaced by the assigned IP address of the interface. All incoming IP packets will be dropped.

3 Assigned Interfaces: eth1

4 Assigned Networks: 192.168.110.0/24 192.168.130.0/24

5 Allowed Services

Service	TCP	UDP	Action
Secure WWW (HTTPS)	443		
amqps	5671		

6 Forward Ports

Port	Protocol	To Port	To Address	Action
18666	TCP	18666	192.168.100.20	
18667	TCP	18667	192.168.0.101	

7

▼ NAT-Trusted Zone

Description: The source IP address of all outgoing IP packets is replaced by the assigned IP address of the interface. Incoming IP packets will be forwarded to the assigned IP address of the interface.

Assigned Interfaces: eth0

Assigned Networks: 192.168.120.0/24

Forward Ports

Port	Protocol	To Port	To Address	Action
No specified forward ports.				

2.3.0. DEVICE

Figure 36: Elements on Firewall configuration page

## Zones

(1) All zones that have been added to your firewall configuration are listed on the **Firewall** page.

Click the ▶ button (expand) in front of a zone's name to show the properties of the zone, like **Interfaces**, **Sources**, **Allowed Services**, **Forward ports** and a brief **Description**.

Click the ▼ button (collapse) to hide the properties of the zone.

Zones can be removed from the firewall by clicking the 🗑 button.

You can add the following zones to your firewall by clicking the **+ Add Zone** button:

Zone *	Description
Drop	All packets reaching the interface will be “silently” dropped.
NAT-Drop	NAT = Network Address Translation, a.k.a. “masquerading”. The source IP address of all outgoing IP packets is replaced by the assigned IP address of the interface. All incoming IP packets will be dropped.
Block	All packets reaching the interface will be dropped. The sender will be notified by an ICMP "unreachable" message.
NAT-Trusted	NAT = Network Address Translation, a.k.a. “masquerading”. The source IP address of all outgoing IP packets is replaced by the assigned IP address of the interface. Incoming IP packets will be forwarded to the assigned IP address of the interface.
Trusted	All IP packets are forwarded transparently. There is no need to add allowed Services/ports to this zone because all services/ports are open anyway. Thus, there is no “Allowed Services” table for this zone.

\* Sorted from “least trusted” to “most trusted”

Table 11: Available Firewall zones

- To add a new zone or to assign new interfaces or subnet(s)/IP address range(s) to an existing zone, click **+ Add Zone** button.
- The **Add Zone** dialog opens:

### Add Zone

**Trust Level**  
Sorted from least trusted to most trusted

**Zones**

Drop
  NAT-Drop
  Block
  NAT-Trusted
  Trusted

**Zone Description**  
The source IP address of all outgoing IP packets is replaced by the assigned IP address of the interface. All incoming IP packets will be dropped.

**Allowed Services**  
None  
The https service is automatically included

**Assign Interfaces**

br-636a5f62fa3d
  eth0
  eth1

**Assign Networks**

Entire subnet of interface  
 Networks ⓘ

Figure 37: Add Zone dialog

Element	Description	
Trust Level	Explains the sorting of the zones under <b>Zones</b>	
Zones	Select here the zone that you want to add to your firewall configuration. If you want to assign <b>Interfaces</b> or <b>Networks</b> to an already existing zone (i.e. to a zone that has already been added to your firewall configuration), select here the corresponding zone to which you want to add the new parameters.	
Zone Description	Displays a brief description of the selected zone.	
Allowed Services	Shows the allowed services/ports of the selected zone. Note that HTTPS is allowed by default in all zones. You can add or delete allowed services to/from an existing zone in the <b>Allowed Services</b> table of the corresponding zone.	
Assign Interfaces	Select here the physical or virtual interface(s) that you want to assign to the selected zone. Note that each interface can be assigned to one zone only. Interfaces that have already been assigned to a different zone are not displayed here and thus cannot be selected here. If you want to reassign an interface from one zone to another, you will first have to remove the interface from the zone to which it is currently belonging.	
Assign Networks	Here you can define subnets or IP address ranges for which the rules of the zone shall apply.	
	Entire subnet of interface	Select this option if the rules shall apply to the entire subnet(s) of the assigned interface(s).
	Networks	Select this option to enter address ranges or subnets for which the rules of the zone shall apply. Enter the subnet mask as CIDR Suffix. Multiple entries must be separated with commas, e.g.: 192.168.1.0/24, 10.14.0.0/16

Table 12: Elements in Add Zone dialog

## Description


(2) Brief description of the function of the zone.

## Assigned Interfaces

(3) Physical or virtual interfaces that are assigned to the zone (i.e. these are the interfaces to which the rules of the zone apply).

You can assign interfaces to a zone in the **Add Zone** dialog when you add a new zone to your firewall.

Note that each interface can be assigned to *one zone* only.

Interface(s) can be removed from a zone by clicking the  button.

If you later want to add another interface to an already existing zone, proceed as follows:


- Click **+ Add Zone** button to open the **Add Zone** dialog.
- In the **Add Zone** dialog, select the existing zone in the **Zones** area.
- Select the new interface in the **Assign Interfaces** area.
- Click the **Add Zone** button in the footer.
- The **Add Zone** dialog closes and the new interface is added to the zone.

## Assigned Networks

(4) These are the subnet(s) or IP address ranges that are assigned to the zone (i.e. these are the subnet(s) respectively IP address ranges to which the rules of the zone apply).

You can assign networks to a zone in the **Add Zone** dialog when you add a new zone to your firewall. If no networks are assigned, the rules of the zone will apply to the entire subnet of the interface by default.

Note that each network can be assigned to *one zone* only.

Networks can be removed from a zone by clicking the  button.

If you later want to add networks to an already existing zone, proceed as follows:

- Click **+ Add Zone** button to open the **Add Zone** dialog.
- In the **Add Zone** dialog, select the existing zone in the **Zones** area.
- Select the **Networks** option in the **Assign Networks** area.
- Enter new subnet(s) or IP address range(s) into the **Networks** field. (Enter the subnet mask as CIDR Suffix and separate multiple entries with commas.)
- Click the **Add Zone** button in the footer.
- ↻ The **Add Zone** dialog closes and the network(s) are added to the zone.

## Allowed Services

(5) The **Allowed Services** table shows the network services and ports that remain “open” in a Drop, NAT-Drop or Block zone.



### Note:

**Secure WWW (HTTPS)/TCP port 443** is by default allowed for all zones and interfaces because this service/port is the standard means of communication of the web server of the netFIELD OS with the netFIELD Cloud. When you add a new zone, HTTPS will therefore be automatically included in the **Allowed Services** list.



### Important:

Be aware that if you delete **HTTPS** from the **Allowed Services** list, you might shut yourself out from the netFIELD OS.



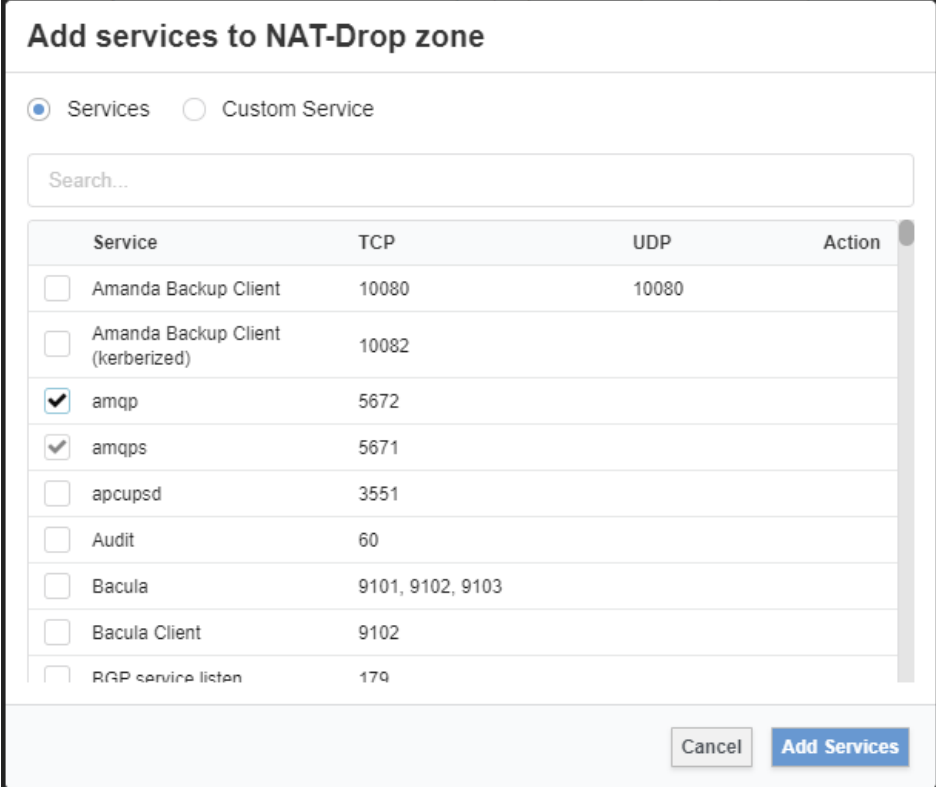
Element	Description	
Service	Name of the service or alias of the custom port that is allowed in the zone.	
TCP	Number of the TCP port that is allowed in the zone.	
UDP	Number of the UDP port that is allowed in the zone.	
Action		Opens a dialog for adding allowed services respectively custom services (ports) to the zone (see below).
		Deletes the allowed service respectively port. <b>Note:</b> Deleting an allowed service/port from a Drop Zone, NAT-Drop Zone or Block Zone can cause loss of connection to your device (if the interface via which you are connected belongs to such a zone).

Table 13: Columns/elements in Allowed Services table

To add a new service respectively port to the **Allowed Services** list of a zone, proceed as follows:

- Click the **+** button above the **Action** column.
- The **Add Services** dialog opens. The dialog features a list of commonly used services and their standard TCP or UDP port numbers:



Service	TCP	UDP	Action
<input type="checkbox"/> Amanda Backup Client	10080	10080	
<input type="checkbox"/> Amanda Backup Client (kerberized)	10082		
<input checked="" type="checkbox"/> amqp	5672		
<input checked="" type="checkbox"/> amqps	5671		
<input type="checkbox"/> apcupsd	3551		
<input type="checkbox"/> Audit	60		
<input type="checkbox"/> Bacula	9101, 9102, 9103		
<input type="checkbox"/> Bacula Client	9102		
<input type="checkbox"/> BGP service listen	179		

Figure 38: Add services

- To find the service/port you are looking for, you can scroll through the list by using the scroll bar or you can enter the name of the service or the port number into the **Search** field.
- Select the service(s)/port(s) in the check box, then click **Add Services** in the footer.
- The dialog closes and the allowed services/ports are added to the zone.

- If you want to add a port that is not bound to a specific service, you can select the **Custom Service** option and enter the port number in the **TCP** respectively **UDP** field. For reference, you should also enter a name for your custom service/port in the **Name** field. You can add several ports at once by separating the entries with a comma.

Figure 39: Add custom services dialog

- Click **Add Custom Service** in the footer.
- The dialog closes and the allowed custom service/port is added to the zone.

## Forward Ports

(6) The firewall supports “port forwarding”, which is commonly used together with NAT zones (NAT = Network Address Translation, a.k.a. “masquerading”). It allows traffic arriving at a certain port of an interface to be forwarded to a certain port of another interface, e.g. of an “internal” interface like a virtual container interface (“veth”), whose IP address is not “visible” to the “outside world”.

Port forwarding settings are displayed in the **Forward Ports** table of the zone.

Element	Description	
Port	Number of the port of the receiving interface from which the traffic is to be forwarded.	
Protocol	Protocol used by the service/port.	
To Port	Number of the port to which the traffic shall be forwarded.	
To Address	IP address of the interface to which the traffic shall be forwarded.	
Action	+	Opens a dialog for adding a new port forwarding definition.
	🗑️	Deletes the port forwarding definition.

Table 14: Columns/elements in Forward Ports table

To add a new port forwarding definition to a zone, proceed as follows:

- Click the **+** button above the **Action** column.



➤ The **Add Forward Port** dialog opens:

Figure 40: Add forward port dialog

- In the **Port** field, enter the number of the port of the receiving interface from which the traffic is to be forwarded.
  - In the **Protocol** drop-down list, select the corresponding protocol.
  - In the **To Port** field, enter the number of the port to which the traffic shall be forwarded.
  - In the **To Address** field, enter the IP address of the interface to which the traffic shall be forwarded.
  - Click the **Add Port** button in the footer.
- The **Add Forward Port** dialog closes and the new port forwarding definition is added to the existing zone.

### Control elements in main toolbar

(7) The main toolbar on top of the **Firewall** configuration page features the following control elements:

Element	Description
	Toggle switch to deactivate the firewall.
<b>Save Permanent</b>	Saves your new firewall configuration settings.
<b>+ Add Zone</b>	Opens the <b>Add Zone</b> dialog. In the <b>Add Zone</b> dialog, you can add a new active zone to your firewall configuration, or you can assign new interfaces or “networks” (subnets/IP address ranges) for an already existing active zone (i.e. for a zone that has already been added to your firewall).

Table 15: Control elements in main toolbar

### 5.3.3 Network Proxy settings

If your local IT network uses proxy server(s) for HTTP, HTTPS, or FTP communication, you must configure the **Network Proxy** settings of the netFIELD OS accordingly.



#### Note:

To ensure that the device will be able to communicate with the cloud, we strongly recommend you to configure the proxy settings *before onboarding* your device. The local proxy settings of the device will be transferred to the netFIELD Portal during onboarding and will be stored there.

The container images that you then deploy from the Portal can thus take over these proxy settings and use them for their own communication when they run on the device after their deployment. Note also that if you change the proxy settings locally on your device *after onboarding*, you must “synchronize” the settings with the netFIELD Portal in order to keep the settings there “up-to-date” (to synchronize, open the **Onboarding** page in the Local Device Manager, then click **Synchronize** button).

You can find the **Network Proxy** settings on the **Networking** page.

The screenshot shows the netFIELD web interface. The sidebar on the left contains navigation options: System, Networking (highlighted with a red arrow), Networking Services, Onboarding, General Settings, Standard Docker, IoT Edge Docker, Accounts, Certificate, Terminal, System Update, Logs, and Services. The main content area displays the Network Proxy configuration page. At the top, there are three input fields for device identifiers: sit0, vethd1de80b, and vethf740d97. Below these is the Network Proxy table, which is highlighted with a red box. The table has three columns: Proxy, Host, and Port. It lists three entries: HTTP (Host: HTTP://10.11.5.98, Port: 3128), HTTPS (Host: HTTPS://10.11.5.99, Port: 3128), and No Proxy (Host: localhost,127.0.0.1). An 'Edit Proxy' button is located in the top right corner of the table. Below the table is the Networking Logs section, dated FEBRUARY 22, 2022, with several log entries showing device status changes.

Proxy	Host	Port
HTTP	HTTP://10.11.5.98	3128
HTTPS	HTTPS://10.11.5.99	3128
No Proxy	localhost,127.0.0.1	

Figure 41: Network Proxy configuration

The **Network Proxy** table shows the current Proxy server settings of your netFIELD OS. The protocols for which a Proxy server is being used are listed in the **Proxy** column, the **Host** column shows the IP address or host name of the corresponding proxy server and the **Port** column shows the port number that the proxy server uses for the protocol.

The **No Proxy** entries designate destinations that shall not be addressed via Proxy server.

By default these are `localhost` and `127.0.0.1`, which are “internal” addresses of the netFIELD OS and are therefore not to be handled by a proxy server. The `localhost` and `127.0.0.1` entries appear in the **No Proxy** list even if you did not configure any Proxy Server for your netFIELD OS.

Do not edit or remove `localhost` and `127.0.0.1` from the **No Proxy** list.

To configure your network proxy settings, proceed as follows:



**Note:**

Ask your local network administrator for the parameters (IP address, ports, passwords etc.) of your local proxy server(s).

- Click the **Edit Proxy** button.
- The **Proxy Settings** dialog opens:

Proxy Settings

HTTP / HTTPS / FTP

Host  Port

Authentication required

Use this proxy server for all protocols

No Proxy

Host

(e.g. intranet.consor.de, .consor.de, 10.15.22.0/24, 10.15.22.12)

Cancel Apply

Figure 42: Proxy Settings dialog window

**Use case a: Using one proxy server for multiple protocols.**

- If the HTTP, HTTPS and/or FTP communication in your local network is handled by a single proxy server, select the **Use this proxy server for all protocols** option.

Proxy Settings

HTTP / HTTPS / FTP

Host  Port

Authentication required

Username

Password

Use this proxy server for all protocols

No Proxy

Host

(e.g. intranet.consor.de, .consor.de, 10.15.22.0/24, 10.15.22.12)

Cancel Apply

Figure 43: Using one Proxy server for all protocols

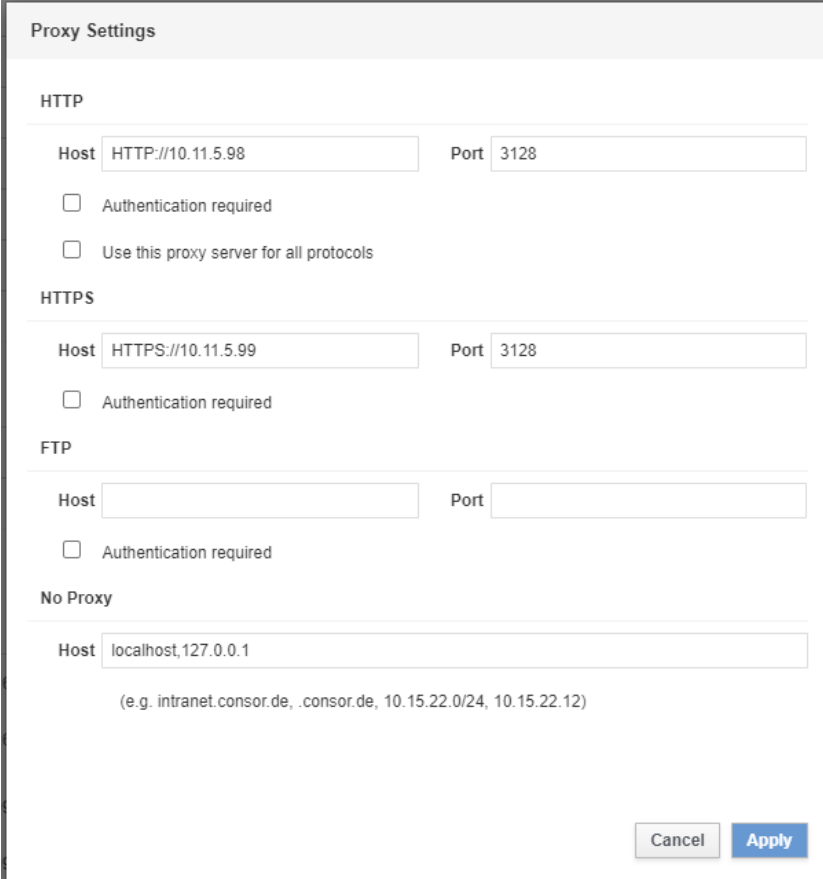
- In the **Host** field, enter the appropriate prefix of the protocol that the proxy server is using, followed by its IP address or host name, e.g.: `http://192.168.20.122`
- In the **Port** field, enter the number of the port that the proxy server is using.
- If your proxy server requires authentication, select the **Authentication required** option and enter **Username** and **Password** of the server.
- In the **No Proxy** section, you can specify destinations that shall not be handled by the proxy server(s). Multiple entries in the **Host** field must be separated by comma.

**Important:**

Do not change or remove the `localhost` and `127.0.0.1` entries in the **No Proxy** section. These are “internal” addresses of the netFIELD OS that cannot be handled by a proxy server because they are required for internal communication. You can, however, add further exceptions in the **Host** field.

**Use case b: Using separate proxy servers for different protocols.**

- If the HTTP, HTTPS and/or FTP communication in your local network is handled by separate proxy servers, uncheck the **Use this proxy server for all protocols** option.
- This enables separate configuration fields for the **HTTP**, **HTTPS** and **FTP** protocols:



The screenshot shows a 'Proxy Settings' dialog box with the following sections:

- HTTP**: Host field contains 'HTTP://10.11.5.98', Port field contains '3128'. There are two unchecked checkboxes: 'Authentication required' and 'Use this proxy server for all protocols'.
- HTTPS**: Host field contains 'HTTPS://10.11.5.99', Port field contains '3128'. There is one unchecked checkbox: 'Authentication required'.
- FTP**: Host and Port fields are empty. There is one unchecked checkbox: 'Authentication required'.
- No Proxy**: Host field contains 'localhost,127.0.0.1'. Below it is a note: '(e.g. intranet.consor.de, .consor.de, 10.15.22.0/24, 10.15.22.12)'. At the bottom right are 'Cancel' and 'Apply' buttons.

Figure 44: Separate HTTP/HTTPS/FTP configuration

- Enter the parameters of the individual proxy servers.

## Saving and restarting

- To save your new proxy server configuration, click **Apply** button.
- The following dialog appears:

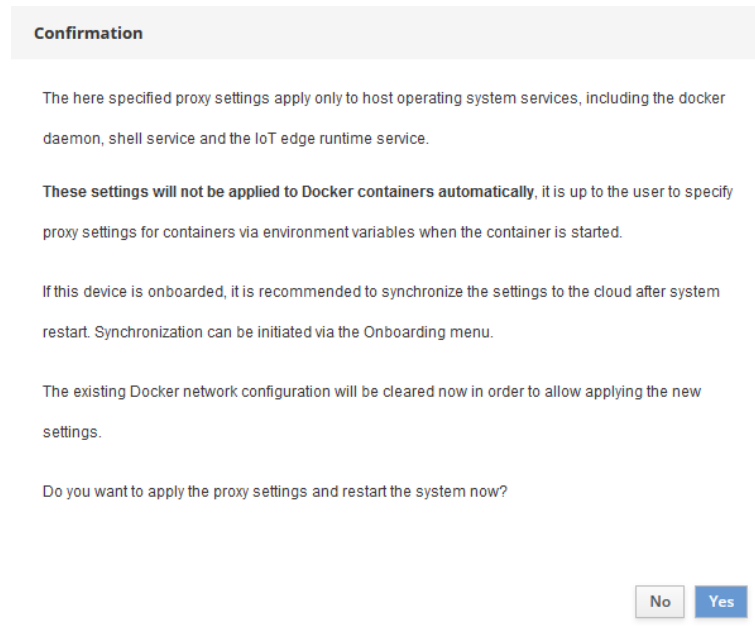


Figure 45: Restart dialog after changing proxy server configuration

- Read the note carefully.
- To apply the new settings, you must allow the netFIELD OS to perform an immediate restart.  
Click **Yes** to apply the new settings and restart the netFIELD OS.
- Click **No** to close the dialog without applying the new settings.

## Synchronizing new settings with the cloud

- If your device was already onboarded in the netFIELD Portal before changing the settings, you must “synchronize” the new proxy server settings with the corresponding data set of the “device twin” in the cloud.  
To do so, open the **Onboarding** page of the netFIELD OS.

- After having changed the proxy settings of an onboarded device, the **Onboarding** page should now display a **Proxy settings changed** note and the **Synchronize** button (if not, refresh the page by pressing **F5** on your keyboard).

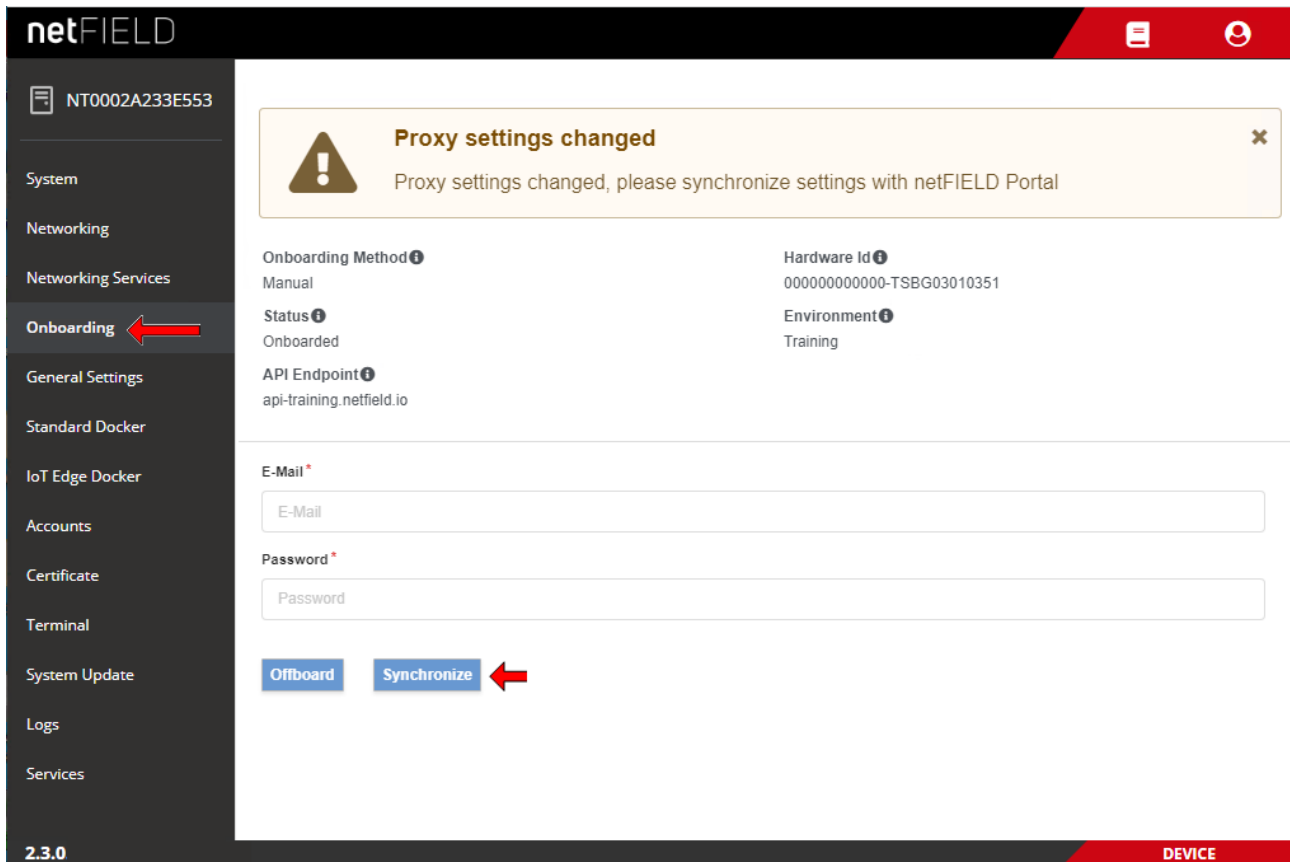


Figure 46: Synchronize proxy settings with netFIELD Portal

- In the **E-Mail** and **Password** fields, enter the credentials of a user of the **portal** who possesses the `updateDevices` permission.
- Click **Synchronize** button.
- If the credentials have been correct, the “**Device proxy settings were updated**” message appears. The proxy server settings of your device in the cloud are now identical with your local settings. You can check the new settings in the Device Manager of the netFIELD Portal under **Device Manager** > **[your device]** > **Overview**. The new settings should be displayed there.

### Removing or editing existing Proxy server settings

If you are not using proxy server(s) in your local IT network any more, you can simply open the **Proxy Settings** dialog window and delete (or edit) the entries in the corresponding fields. After clicking the **Apply** button, the proxy server will be removed from the configuration and the new settings will become effective after restarting the netFIELD OS.

If your device is onboarded in the netFIELD Portal, do not forget to synchronize the new settings.

## 5.4 Networking Services

### 5.4.1 Wi-Fi

On the **Wi-Fi** tab of the **Networking Services** page, you can configure the Wi-Fi settings of netFIELD Edge Devices that are equipped with a wireless interface. Since the **netFIELD Compact X8M** device (NFX8M-D2-N32-010) has no Wi-Fi interface, this tab simply says “Wi-Fi hardware is not available or disabled”.

### 5.4.2 DHCP Server

On the **DHCP Server** tab of the **Networking Services** page, you can configure the DHCP service of netFIELD Edge Devices that are equipped with a Wi-Fi interface providing an Access Point. Since the **netFIELD Compact X8M** device (NFX8M-D2-N32-010) has no Wi-Fi interface, this tab simply says “Wi-Fi hardware is not available or disabled”.



### 5.4.3 Connectivity Check

On the **Connectivity Check** tab of the **Networking Services** page, you can test the connectivity of the cloud communication channels that are used by the Linux host system and the *Azure IoT Edge runtime* in the IoT Edge Docker. Some other configuration settings that are relevant for proper connectivity (like the local host time and the Docker's DNS settings) are also checked here.

The cloud connectivity checking functions are provided by the *iotedge check* tool (version 1.2.5) of the IoT Edge runtime, which uses the *azureiotedge-diagnostics* container for this.

Therefore, your device must be onboarded in the netFIELD Cloud (which enables the IoT Edge Docker and the IoT Edge runtime) for using this function. However, using only the **Ping** test works without the device being onboarded.



For more information on the *iotedge check* tool, see <https://docs.microsoft.com/en-us/azure/iot-edge/troubleshoot?view=iotedge-2020-11> and <https://github.com/Azure/iotedge/blob/main/doc/troubleshoot-checks.md>

The screenshot shows the netFIELD interface with the 'Connectivity Check' tab selected. The interface includes a sidebar with navigation options and a main content area with a 'Ping' input field and a 'Check' button. Below this is a table of connectivity checks.

Check	Result
Host can connect to and perform TLS handshake with iotHub AMQP port	Green
Container on the default network can connect to upstream AMQP port	Green
Container on the IoT Edge module network can connect to upstream AMQP port	Green
Host can connect to and perform TLS handshake with iotHub HTTPS / WebSockets port	Green
Container on the default network can connect to upstream HTTPS / WebSockets port	Green
Container on the IoT Edge module network can connect to upstream HTTPS / WebSockets port	Green
Host can connect to and perform TLS handshake with iotHub MQTT port Could not connect to eplOthub-Training.azure-devices.net : could not complete TLS handshake	Red
Container on the default network can connect to upstream MQTT port	Green
Container on the IoT Edge module network can connect to upstream MQTT port	Green
Host time is close to reference time	Green
<b>DNS server</b> Container engine is not configured with DNS server setting, which may impact connectivity to IoT Hub. Please see <a href="https://aka.ms/iotedge-prod-checklist-dns">https://aka.ms/iotedge-prod-checklist-dns</a> for best practices. You can ignore this warning if you are setting DNS server per module in the Edge deployment.	Yellow

Figure 47: Connectivity Check tab

### LAN/Internet ping

To test the LAN respectively Internet connection, enter the IP address or the hostname of an endpoint in the **Ping** field, then click **Ping** button.

### Cloud Connectivity

To test the connectivity of the components that are involved in connecting the IoT Edge runtime to the netFIELD Cloud, click **Check** button.

The result is indicated with a dot:

- OK (available)
- Warning
- Error (not available)



For more information on the checks that are being performed, see <https://github.com/Azure/iotedge/blob/main/doc/troubleshoot-checks.md>

Note that the current netFIELD OS uses the *iotedge check* tool version 1.2.5.

---

## 5.5 Onboarding (and offboarding)

The **Onboarding** page allows you to “register” your device in the netFIELD Portal. For a detailed description of the onboarding process and the parameters on this page, see section *“Onboard” (register) device in netFIELD Cloud* [▶ page 37]. You can also “offboard” your device here.

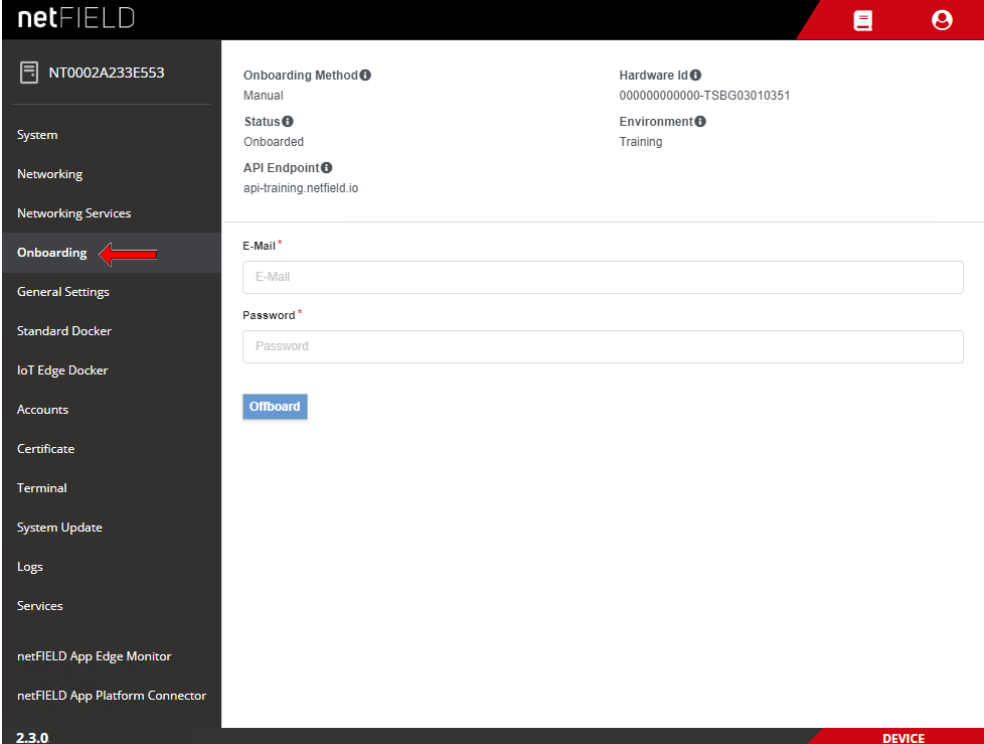
If you have changed the HTTP/HTTPS/FTP proxy server settings of your device *after onboarding*, you can also “synchronize” these new settings here with the netFIELD Portal by clicking the **Synchronize** button. (The **Synchronize** button will only be visible if you have actually changed the proxy server settings. See also section *Network Proxy settings* [▶ page 66] for further information.)

The screenshot shows the netFIELD Onboarding page for device NT0001C02E1F11. The page is divided into a sidebar and a main content area. The sidebar contains navigation links for System, Networking, Networking Services, Onboarding (highlighted with a red arrow), General Settings, Standard Docker, IoT Edge Docker, Accounts, Certificate, Terminal, System Update, Logs, and Services. The main content area displays device information: Onboarding Method (Manual), Status, API Endpoint, Hardware Id (000000000000-0001C02E1F11), and Environment. Below this, there are tabs for Basic and Advanced settings. The Basic tab is active, showing fields for Environment, Device Name, E-Mail, Password, and Upstream Protocol. A checkbox for 'Use Deployment Manifest' is also present. An 'Onboard' button is at the bottom. The footer shows version 2.3.0 and a 'DEVICE' label.

Figure 48: Basic Onboarding page

Once your device has been onboarded, the page changes and shows the parameters for “offboarding” the device. By offboarding it, the device will be “deleted” in the portal and removed from the device list of the portal’s **Device Manager**:

## Offboarding after having used the Basic Onboarding method



The screenshot shows the netFIELD web interface. The left sidebar contains a menu with the following items: System, Networking, Networking Services, **Onboarding** (highlighted with a red arrow), General Settings, Standard Docker, IoT Edge Docker, Accounts, Certificate, Terminal, System Update, Logs, Services, netFIELD App Edge Monitor, and netFIELD App Platform Connector. The main content area displays the following information:

Onboarding Method ⓘ	Hardware Id ⓘ
Manual	000000000000-TSBG03010351
Status ⓘ	Environment ⓘ
Onboarded	Training
API Endpoint ⓘ	
api-training.netfield.io	

Below the information, there are two input fields: "E-Mail \*" and "Password \*". A blue "Offboard" button is located below the "Password" field. The bottom of the interface shows the version "2.3.0" and the word "DEVICE" in a red bar.

Figure 49: Offboarding “Basic”

- In the **E-Mail** and **Password** fields, enter the credentials of a user of the **netFIELD Portal** who possesses `deleteDevices` and `offboardedDevices` permissions.
- Click **Offboard** button.
- After successful offboarding, the following message appears: **Success – Device is now deleted.**

## Offboarding after having used the Advanced Onboarding method

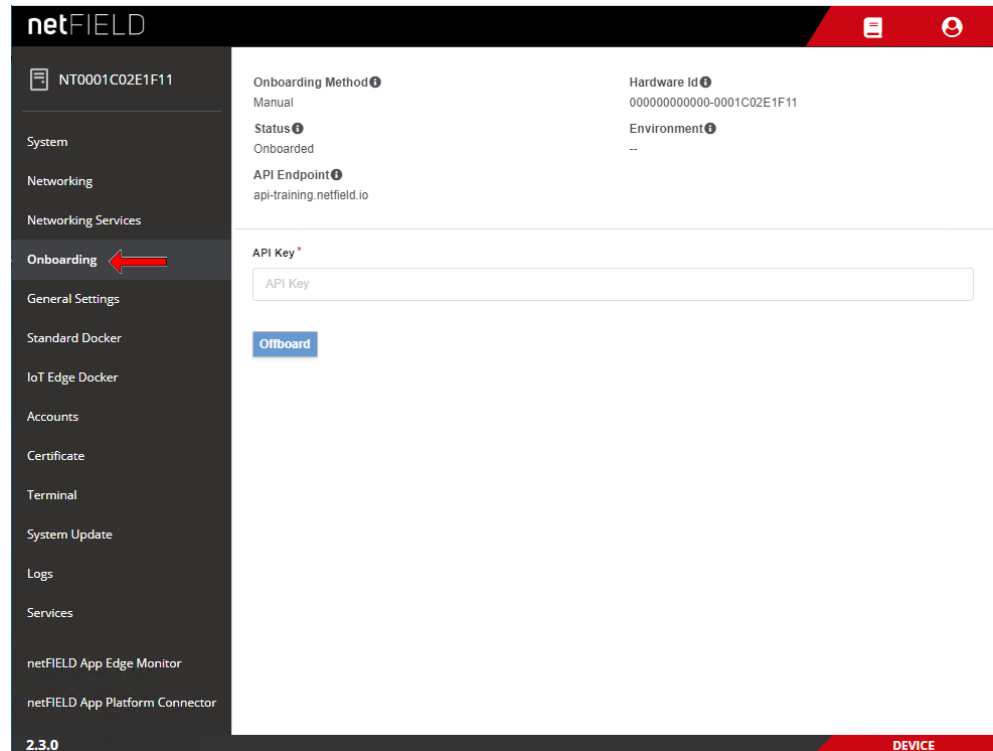


Figure 50: Offboarding “Advanced”

- In the **API KEY** field, enter an API Key that possesses the right to offboard devices. I.e. this key must have **Security Level** `org+ch` or `org` for the `deleteDevices` and `offboardedDevices` functions of the **devices** resource.
- Click **Offboard** button.
- ⇒ After successful offboarding, the following message appears: **Success – Device is now deleted.**



### Note:

After offboarding, all application containers managed by the netFIELD Portal are automatically deleted. However, the Docker images are still present on the device. They can be deleted manually on the **IoT Edge Docker** page of the Local Device Manager.

## 5.6 General Settings

### 5.6.1 Web Server (Port) Settings

On the **Web Server** tab of the **General Settings** page, you can change the TCP ports of the web server of the netFIELD OS.

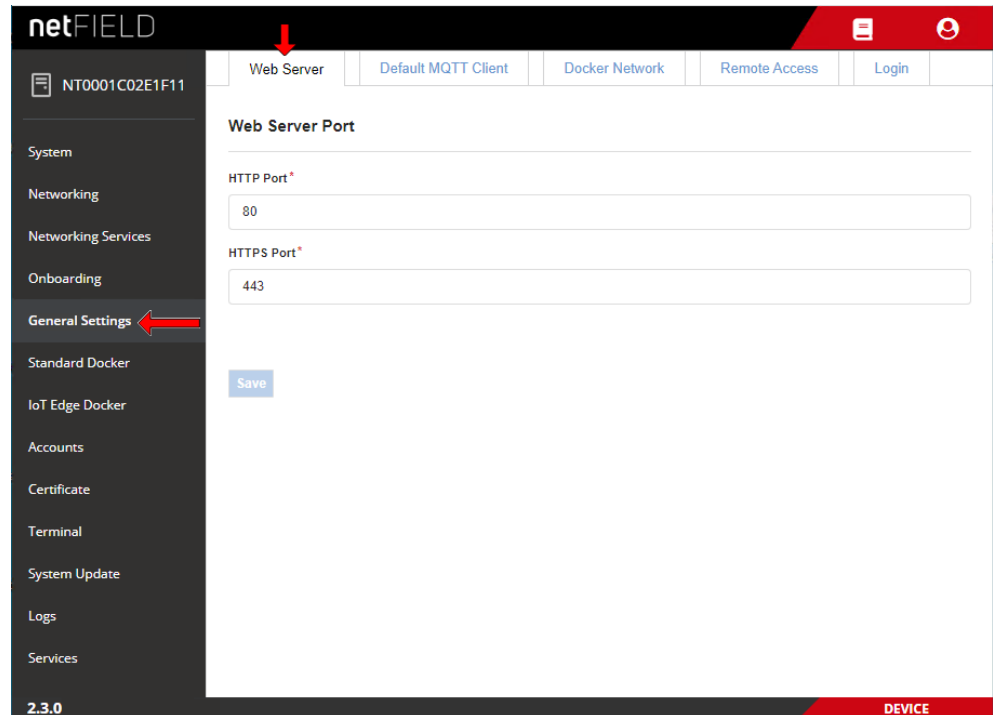


Figure 51: Web Server Settings tab

By default, the netFIELD OS uses port 80 for its HTTP communication and port 443 for its HTTPS communication.



#### Important:

The new settings become immediately effective after saving and confirming the changes, which means that your current HTTP/HTTPS connection to the netFIELD OS respectively Local Device Manager will be lost.

You will have to reconnect by specifying the new port number after the IP address in the address bar of your web browser.



#### Note:

Changing the web server port settings will have no effect on the **Remote Control** function that allows you to access the Local Device Manager from the netFIELD Portal via “web tunnel”. For more information about the Remote Control function, see *netFIELD Portal* operating instructions manual, DOC1907010IxxEN.

- Click **Save** button to save your new Web Server Settings.

## 5.6.2 Default MQTT Client Settings

On the **Default MQTT Client** tab of the **General Settings** page, you can change the MQTT Client configuration parameters that shall be used by the Docker containers that are running on your netFIELD OS. These settings are stored in a JSON configuration file in the netFIELD OS (`/etc/gateway/mqtt-config.json`).

By default, all Hilscher netFIELD Apps use this configuration file. Other containers (i.e. non-Hilscher application containers) that do not require their own customized MQTT client settings, can also use these settings here if the configuration file is referenced accordingly in the container image (e.g. in the *Container Create Options* of the netFIELD Portal, see *netFIELD Portal operating instructions manual, DOC1907010IxxEN*).

The screenshot shows the netFIELD web interface. The top navigation bar includes 'Web Server', 'Default MQTT Client' (selected), 'Docker Network', 'Remote Access', and 'Login'. The left sidebar menu includes 'System', 'Networking', 'Networking Services', 'Onboarding', 'General Settings' (selected), 'Standard Docker', 'IoT Edge Docker', 'Accounts', 'Certificate', 'Terminal', 'System Update', 'Logs', and 'Services'. The main content area is titled 'Gateway settings' and contains the following fields:

- Gateway prefix: 000000000000-0001C02E1F11
- MQTT Version: 3.1
- Keep Alive Interval (Seconds): 60
- Username: Username
- Password: Password
- Connect Timeout (Seconds): 300
- Clean Session:
- Server URIs: +

Figure 52: Default MQTT Settings

Element		Description	
Gateway settings	Gateway prefix	Identifies the device. By default, this is the Hardware ID of the device.	
Basic	MQTT Version	MQTT version to be used (depending on the MQTT broker).	
	Keep Alive Interval	Defines the maximum length of time in seconds that the broker and client may not communicate with each other.	
	Username	User name for authentication at the broker (if implemented and required by the broker). Note that the <i>mosquitto</i> broker from the netFIELD Portal does not require login authentication.	
	Password	Password for authentication at the broker (if implemented and required by the broker). Note that the <i>mosquitto</i> broker from the netFIELD Portal does not require login authentication.	
	Connect Timeout	Defines the maximum length of time in seconds that is allowed for completing the connection process.	
	Clean session	If <b>Clean session</b> is selected, the client does not want a persistent session (meaning that if the client disconnects for any reason, all information and messages that are queued from a previous persistent session are lost). If <b>Clean session</b> is unchecked, the broker creates a persistent session for the client.	
Server URIs		Server URI or FQDN of the MQTT broker <b>Note:</b> When multiple server URIs are specified, the client will try to connect to each server one after the other, starting with the first server in the list. If a server connection was established successfully, only this connection will be used. The client will not open multiple connections to multiple servers simultaneously.	
Last Will and Testament		Select this option if you want to use the “last will and testament” (LWT) feature of MQTT. (I.e. to notify other clients about an unexpected loss of connection to the broker)	
		Topic Name	Topic name of LWT message
		Retained	“Retained” flag of LWT message
		Quality of Service	QoS of LWT message
		Message	Message text, e.g. “unexpected loss of connection”
SSL / TLS		Select this option if you want to use SSL/TLS encryption for creating a secure connection to the MQTT broker. <b>Note:</b> This option is for expert users only! In the standard use case, in which the <i>mosquitto</i> broker and the Docker containers are running on the same device, a secure SSL/TLS connection is not necessary (the overhead of the secure connection can thus be avoided).	
		File name and path to private key in PEM format	Enter here the complete path to the private key on the device.
		File name and path to certificate chains in PEM format	Enter here the complete path to the certificate chains on the device.
		Override the trusted CA certificates in PEM format	Enter here the complete path to override the trusted CA certificates on the device.
		Enable verification of the server certificate	If this option is disabled, the Docker containers will also accept invalid certificates from the broker (not recommended).

Table 16: Default MQTT Client Settings

- Click **Save** button to save your new Default MQTT Client Settings.



### 5.6.3 Docker Network Settings

On the **Docker Network** tab of the **General Settings** page, you can change the network address settings of the Standard Docker and of the IoT Edge Docker.



**Important:**

These network address settings are predefined by Hilscher. Change these default addresses only if they are not compatible with your company’s LAN address configuration, i.e. to avoid an address conflict.

Note that after changing the address settings of the Standard and/or IoT Edge Docker all containers running on the corresponding Docker will be stopped and deleted and the netFIELD OS will be automatically restarted. After restart, you might have to re-deploy the deleted containers.

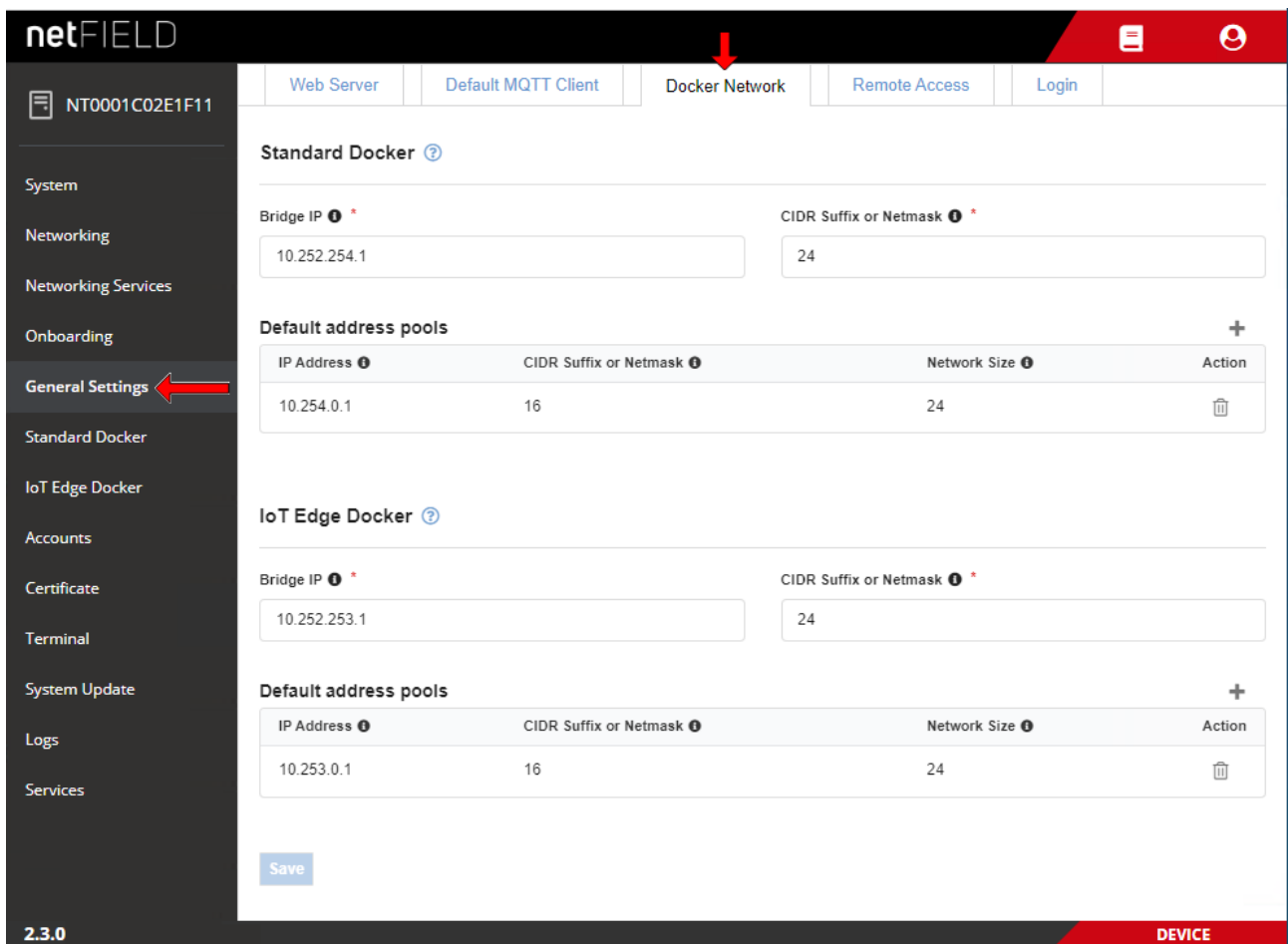


Figure 53: Docker Network Settings

## Standard Docker

The **docker0** bridge is a virtual interface created by the Standard Docker. By default, it uses the address 10.252.254.1/24 (“private range” as defined in RFC 1918) if the address is not already used on the host machine.

All containers running on the Standard Docker connect to this **docker0** bridge by default. The containers can use the iptables/NAT rules (NAT = Network Address Translation, a.k.a. “masquerading”) created by the Standard Docker to communicate with destinations outside the netFIELD OS.

Element	Description	
Bridge IP	IP address of the <b>docker0</b> bridge. Default: 10.252.254.1 <b>Note:</b> Do not change the default address unless necessary to avoid an address conflict with your LAN. Do not use the same Bridge IP address for both Standard and IoT Edge Docker.	
CIDR Suffix or Netmask	Subnet mask of the <b>docker0</b> bridge as CIDR Suffix or in “dotted decimal notation”. Default (CIDR Suffix): 24 Default (dotted decimal notation): 255.255.255.0	
Default address pools	Here you can define “reserve” address pools (subnets) for the internal Docker bridge networks. The default pool consisting of the IP address/CIDR Suffix 10.254.0.1/16 with network size 24 means that the first additional Docker network bridge interface will be created with the IP address/CIDR Suffix 10.254.0.1/24, the second will be 10.254.1.1/24, the third will be 10.254.2.1/24, and so on.	
	IP address	Reserved IP address of the internal Docker bridge network.
	CIDR Suffix or Netmask	Subnet mask of the internal Docker bridge network as CIDR Suffix or in “dotted decimal notation”.
	Network Size	Number of bits used as the netmask for further Docker bridge networks.
	Action	<div style="display: flex; justify-content: space-between;"> <div style="text-align: center;">+</div> <div>Opens a dialog for adding a new pool of reserved addresses.</div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div style="text-align: center;">🗑</div> <div>Deletes the address pool.</div> </div>

Table 17: Standard Docker Network Settings

## IoT Edge Docker

The **iotedge0** bridge is a virtual interface created by the IoT Edge Docker. By default, it uses the address `10.252.253.1/24` (“private range” as defined in RFC 1918) if the address is not already used on the host machine.

All containers running on the IoT Edge Docker connect to this **iotedge0** bridge by default. The containers can use the iptables/NAT rules (NAT = Network Address Translation, a.k.a. “masquerading”) created by the IoT Edge Docker to communicate with destinations outside the netFIELD OS.







Element	Description				
Bridge IP	IP address of the <b>iotedge0</b> bridge. Default: <code>10.252.253.1</code> <b>Note:</b> Do not change the default address unless necessary to avoid an address conflict with your LAN. Do not use the same Bridge IP address for both Standard and IoT Edge Docker.				
CIDR Suffix or Netmask	Subnet mask of the <b>iotedge0</b> bridge as CIDR Suffix or in “dotted decimal notation”. Default (CIDR Suffix): <code>24</code> Default (dotted decimal notation): <code>255.255.255.0</code>				
Default address pools	Here you can define “reserve” address pools (subnets) for the internal IoT Edge Docker bridge networks. The default pool consisting of the IP address/CIDR Suffix <code>10.253.0.1/16</code> with network size <code>24</code> means that the first additional IoT Edge Docker network bridge interface will be created with the IP address/CIDR Suffix <code>10.253.0.1/24</code> , the second will be <code>10.253.1.1/24</code> , the third will be <code>10.253.2.1/24</code> , and so on.				
	IP address	Reserved IP address of the internal IoT Edge Docker bridge network.			
	CIDR Suffix or Netmask	Subnet mask of the internal IoT Edge Docker bridge network as CIDR Suffix or in “dotted decimal notation”.			
	Network Size	Number of bits used as the netmask for further IoT Edge Docker bridge network.			
	Action	<table border="1"> <tr> <td></td> <td>Opens a dialog for adding a new pool of reserved addresses.</td> </tr> <tr> <td></td> <td>Deletes the address pool.</td> </tr> </table>		Opens a dialog for adding a new pool of reserved addresses.	
	Opens a dialog for adding a new pool of reserved addresses.				
	Deletes the address pool.				

Table 18: Standard Docker Network Settings

➤ Click **Save** button to save your new Docker Network Settings.

The following picture illustrates the default Docker network configuration:

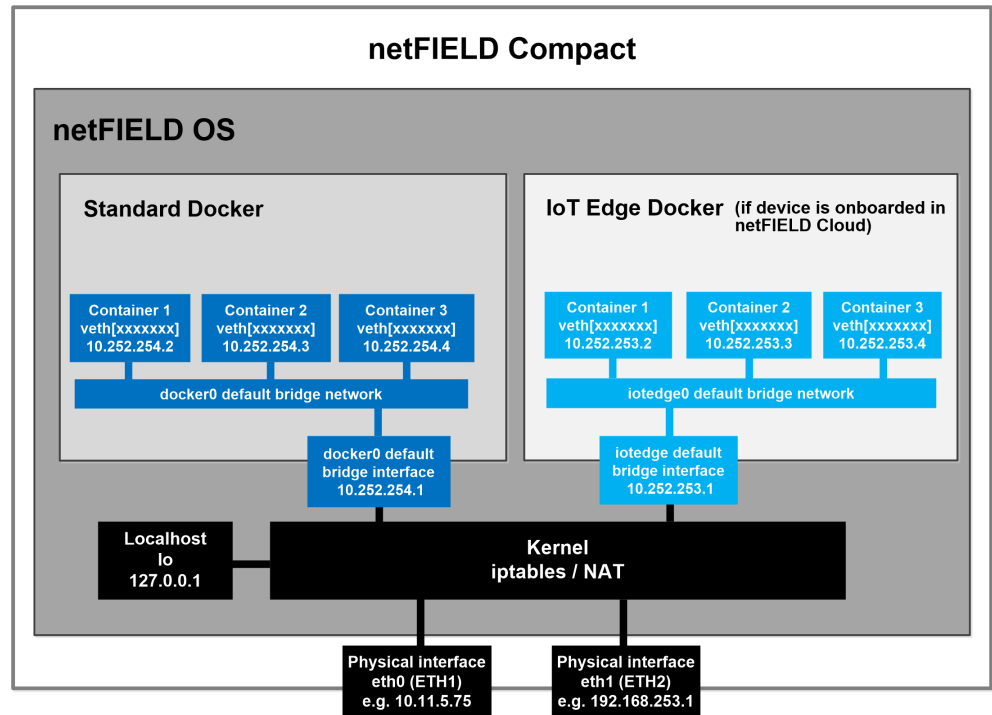


Figure 54: Default docker network configuration

## 5.6.4 Remote Access

On the **Remote Access** tab of the **General Settings** page, you can enable (on) or disable (off) *Remote Control* access from the netFIELD Portal to your device.

**Note:**

Note that your device must be onboarded in the netFIELD Cloud and connected to the Internet in order to use the remote control functions.

Contact your local Hilscher sales representative for information on the terms and conditions of an account/subscription for the *netFIELD Cloud services* (<https://www.netfield.io>).

---

For security reasons, remote control access is by default switched off. To allow remote control of your device, you must enable it here in the Local Device Manager *and* in the netFIELD Portal (“four-eyes-principle”).

**Note:**

The “Remote Control” functions of the Portal allow you to access IP services (like e.g. HTTP(S), SSH, VNC, RDP or other TCP-based services) running on your netFIELD Edge Device/netFIELD OS (or on other devices connected to a network that is accessible by the netFIELD Edge Device/netFIELD OS) from a remote PC via a HTTPS tunnel. The HTTPS tunnel is established by the remote agent container, which is automatically downloaded and started on your device/netFIELD OS when you click the **Enable Remote Control** button on the **Overview** page of your device in the Portal for the first time.

For a detailed description of the remote control functions, see section *Remote Control* in the *netFIELD Portal* manual, (DOC1907010IxxEN).

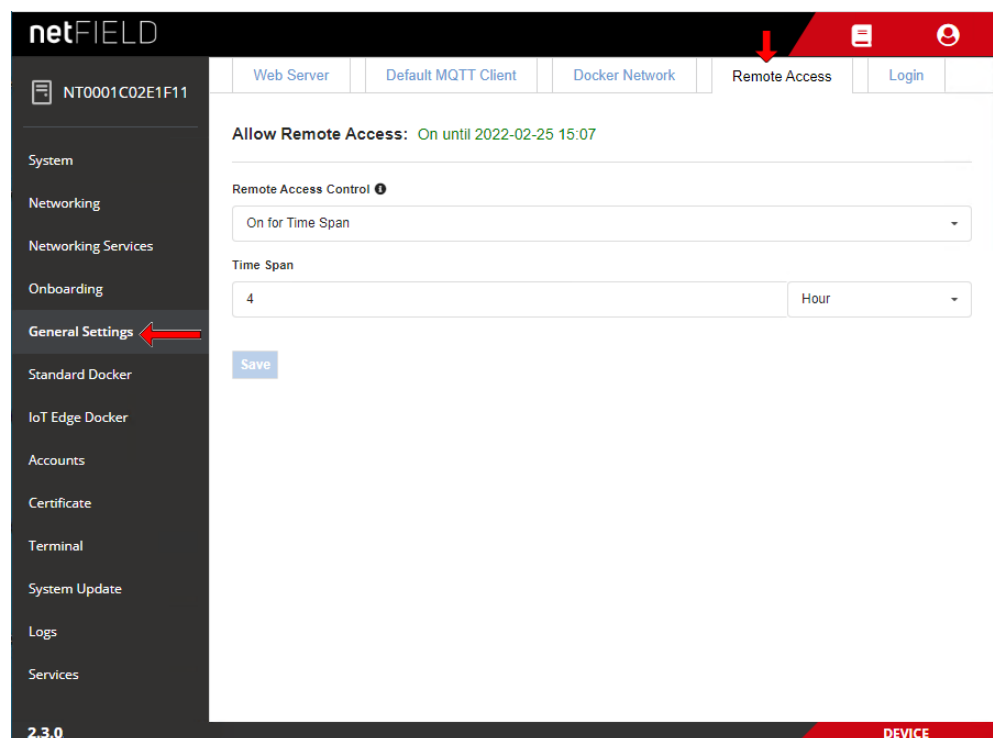


Figure 55: Remote Access tab

- In the **Remote Access Control** dropdown-list, enable (**on**) or disable (**off**) the remote access according to your use case. You can also define time limits (**On for Time Span**) for allowing remote access to the device.

**Important:**

Be aware that disabling the Remote Access and clicking the **Save** button will instantly cut off your remote connection from the netFIELD Portal to your device. Accessing the netFIELD OS will then be possible via local LAN, SSH or Console connection only.

- Click **Save**.

## 5.6.5 Login

On the **Login** tab of the **General Settings** page, you can define a message that will be displayed on the login screen of the Local Device Manager. This allows you e.g. to implement a “system use notification” in accordance with IEC 62443.

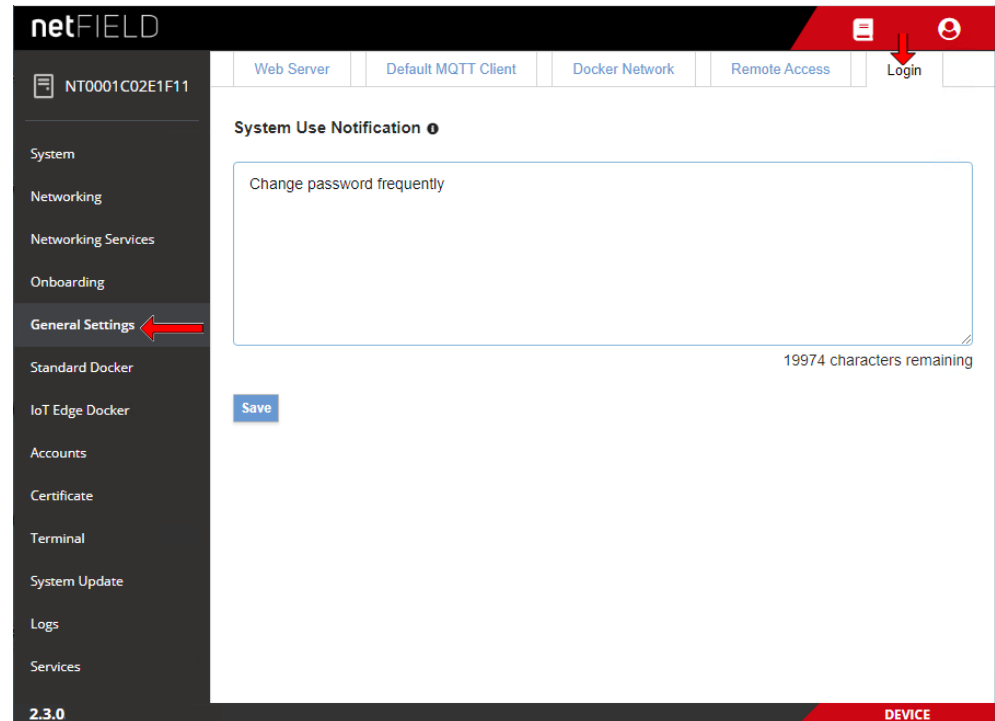


Figure 56: Login tab

- In the text field, enter the message that shall be displayed, then click **Save** button.
- The message will be displayed in the *Sign In* dialog of the Local Device Manager:

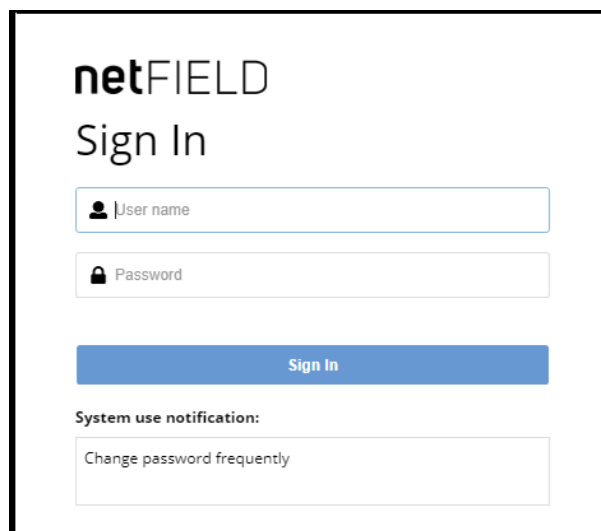


Figure 57: Notification on Sign In dialog

- To remove the message from the *Sign In* dialog again, go to **General Settings > Login** and delete the message from the text field, then click **Save** button.

## 5.7 Standard Docker

The **Standard Docker** page allows you to manage Docker images and containers from the “standard” Docker Hub or from a local repository. It lists all containers that were deployed on the device; except for those that were deployed from the netFIELD Cloud via netFIELD Portal (containers deployed from the netFIELD Cloud are listed on the **IoT Edge Docker** page – see section *IoT Edge Docker* [▶ page 95]).

Unlike the **IoT Edge Docker**, the Standard Docker can be used without having to “onboard” the device in the portal beforehand.

If your device is connected to the Internet, you can pull here images directly from the Docker Hub by clicking the **Get new image** link on this page.



### Note:

The network address settings of the Standard Docker can be managed under **General Settings > Docker Network Settings** (see section *Docker Network Settings* [▶ page 81]).

The screenshot displays the netFIELD Standard Docker interface. On the left is a navigation menu with options like System, Networking, and Standard Docker (highlighted with a red arrow). The main content area is divided into sections:

- Header (1):** A dropdown menu set to 'Images and running containers' and a search bar labeled 'Type to filter...'. A red circle with the number 1 is next to the dropdown.
- System Metrics (2):** Two line graphs showing '% Combined usage of 4 CPU cores' and 'MIB Combined memory usage' over time. A red circle with the number 2 is next to the first graph.
- Containers (3):** A table listing running containers. A red circle with the number 3 is next to the title.
 

Name	Image	Command	CPU	Memory	State
> postgres01	postgres:latest	docker-entrypoint.sh postgres	4%	2.97 MiB	running
> portainer	portainer/portainer-ce:latest	/portainer	0%	5.98 MiB	running
- Images (4):** A table listing Docker images. A red circle with the number 4 is next to the title.
 

Name	Created	Size	Action
> portainer/portainer-ce:latest	Last Sunday at 11:06 PM	152 MiB	▶
> postgres:latest	05/14/2021	253 MiB	▶

Figure 58: Standard Docker

### Filter options in header

The elements in the header (1) allow you to filter the display of containers and images.

You can choose in the drop-down list:

- **Images and running containers** – All downloaded Docker images and currently running containers are displayed (default).
- **Everything** - All Docker images and containers are displayed (including stopped containers).



Use the **Filter** field to display only certain containers.

## Graphs

The graphs (2) show you the load of the containers on the system resources.

**Combined usage of 4 CPU cores:** Load of the containers on the CPUs.

**Combined memory usage:** Load of the containers on the memory.

The graph in the upper right corner shows the amount of mass storage memory taken by the images and containers (blue bar) and the amount of mass storage left available.

## Containers

The **Containers** area (3) lists the container instances of the Docker images according to your Filter options settings in the header (1).

- To expand a box showing concise container details, or to display control buttons to restart, stop or delete it, click on the blue > arrow icon on the left of the container in the list:

The screenshot shows the netFIELD interface with a sidebar on the left containing navigation options like System, Networking, and Standard Docker. The main area displays system metrics (CPU and memory usage graphs) and a 'Containers' table. The table lists containers with columns for Name, Image, Command, CPU, Memory, and State. The 'postgres01' container is selected, and its details are expanded below the table, showing ID, creation time, image, command, and state. A red arrow points to the blue expand icon on the left of the 'postgres01' container name.

Name	Image	Command	CPU	Memory	State
> portainer_1	portainer/portainer-ce:latest	/portainer	0%	10.4 MiB	running
> postgres01	postgres:latest	docker-entrypoint.sh postgres			running

**Details for postgres01:**

- Id: 972d741362379dd52d6d45f9d5ed3de5b7a3393eabc525720b70bf90d1df83f5
- Created: Today at 12:39 PM
- Image: postgres:latest
- Command: docker-entrypoint.sh postgres
- State: Up since Today at 1:44 PM

Figure 59: Expand concise container details

- To manage a container, click on it in the list.

- A page featuring detailed container information opens. Depending on its configuration, the page also includes a terminal or a “console output” window for the running container. Here you can also start, stop, restart, delete or commit the container, or change its resource limits:

The screenshot displays the netFIELD interface for a container named 'portainer\_1'. The left sidebar shows navigation options: System, Networking, Networking Services, Onboarding, General Settings, Standard Docker (selected), IoT Edge Docker, Accounts, Certificate, Terminal, System Update, Logs, and Services. The main content area shows container details for 'portainer\_1' with the following information:

- Container: portainer\_1
- Id: 8a15677859e3ab29277a1186852d8a41a1537281dc72b4c4839e34c05033cf69
- Created: Today at 1:40 PM
- Image: portainer/portainer-ce:latest
- sha256:e3c34da15e54b9e35e33762299ab5dd7c39b4b2d61934ffc8eebeee0cef91802
- Command: /portainer
- State: Up since Today at 1:46 PM
- Restart Policy: Always
- IP Address: 10.252.254.2
- CIDR Suffix: 24
- Gateway: 10.252.254.1
- MAC Address: 02:42:0a:fc:fe:02
- Memory usage: 24.9 MIB
- CPU usage: 0%
- 1024 shares

At the bottom of the container details, there is a 'Change resource limits' button. On the right side, there is a terminal window showing the following log output:

```
2021/05/27 11:40:49 server: Reverse tunnelling enabled
2021/05/27 11:40:49 server: Fingerprint 35:29:8a:69:55:59:d3:26:4d:18:49:d5:10:c
a:105:4d
2021/05/27 11:40:49 server: Listening on 0.0.0.0:8000...
2021/05/27 11:40:49 Starting Portainer 2.5.0 on :9000
2021/05/27 11:40:49 [DEBUG] [chisel, monitoring] [check_interval_seconds: 10.000
000] [message: starting tunnel management process]
2021/05/27 11:40:49 [DEBUG] [internal,init] [message: start initialization monit
or ]
2021/05/27 11:45:50 [FATAL] [internal,init] No administrator account was created
in 5.000000 mins. Shutting down the Portainer instance for security reasons
2021/05/27 11:46:02 server: Reverse tunnelling enabled
2021/05/27 11:46:02 server: Fingerprint 35:29:8a:69:55:59:d3:26:4d:18:49:d5:10:c
a:105:4d
2021/05/27 11:46:02 server: Listening on 0.0.0.0:8000...
2021/05/27 11:46:02 Starting Portainer 2.5.0 on :9000
2021/05/27 11:46:02 [DEBUG] [chisel, monitoring] [check_interval_seconds: 10.000
000] [message: starting tunnel management process]
2021/05/27 11:46:02 [DEBUG] [internal,init] [message: start initialization monit
or ]
```

Figure 60: Container parameters with terminal window

- To go back to the **Standard Docker** overview page, click the blue **Show all containers** link in the page header.

## Images

The **Images** area (4) lists the Docker images that you have downloaded from the “standard” Docker Hub.

- You can download a Docker image by clicking the **Get new image** link.
- The **Image Search** dialog opens, allowing you to search the Docker Hub registry:

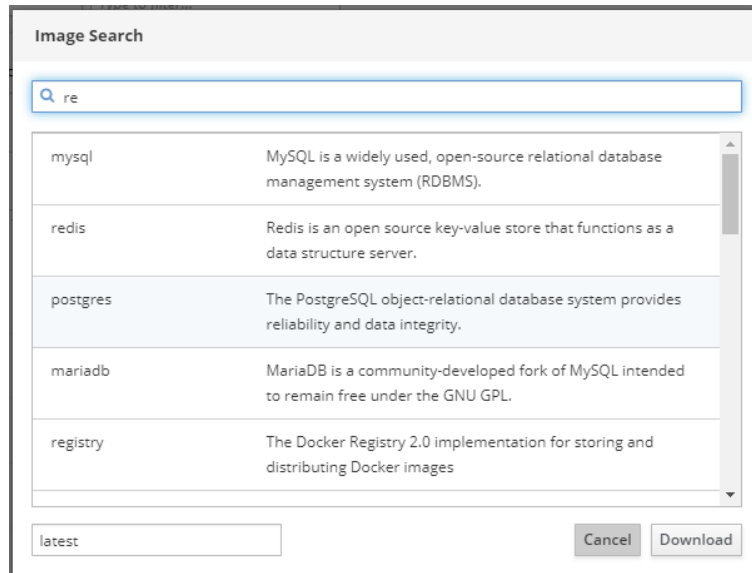

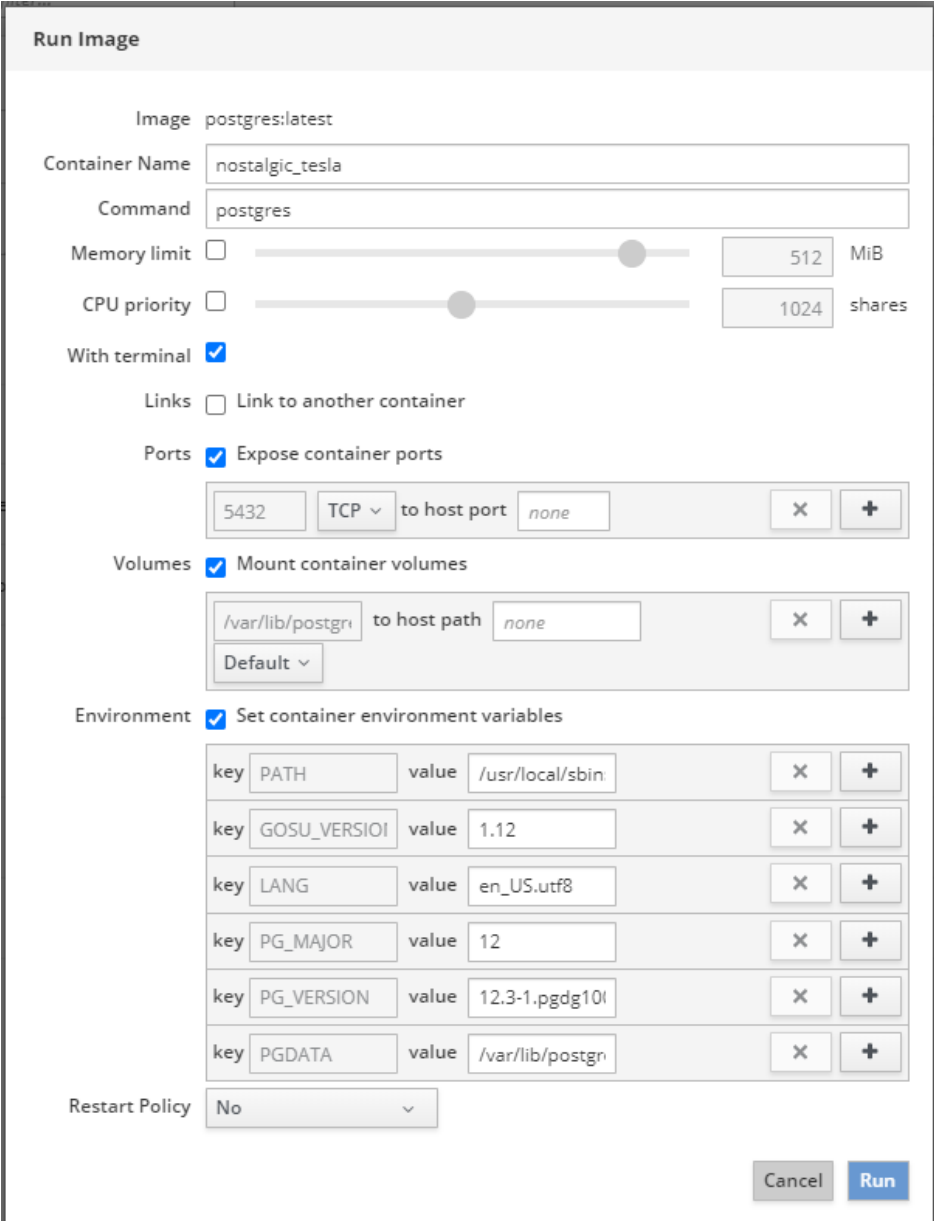


Figure 61: Image Search dialog of Standard Docker

- In the search field, type-in a name or search string, then press **Enter** on your keyboard.
- A list featuring the search results is displayed.
- Select an image in the list, then click **Download** button.
- The image is downloaded, extracted and displayed in the **Images** area.

### Starting a container

- You can start a container (i.e. run an instance of the program contained in the image), by clicking the  button on the right side of the image in the list.
- The **Run Image** dialog opens, in which you can configure the container before running it:



**Run Image**

Image postgres:latest

Container Name nostalgic\_tesla

Command postgres

Memory limit  512 MiB

CPU priority  1024 shares

With terminal

Links  Link to another container

Ports  Expose container ports

5432 TCP to host port none

Volumes  Mount container volumes

/var/lib/postgr to host path none

Default

Environment  Set container environment variables

key	value
PATH	/usr/local/sbin
GOSU_VERSION	1.12
LANG	en_US.utf8
PG_MAJOR	12
PG_VERSION	12.3-1.pgdg10l
PGDATA	/var/lib/postgr

Restart Policy No

Cancel Run

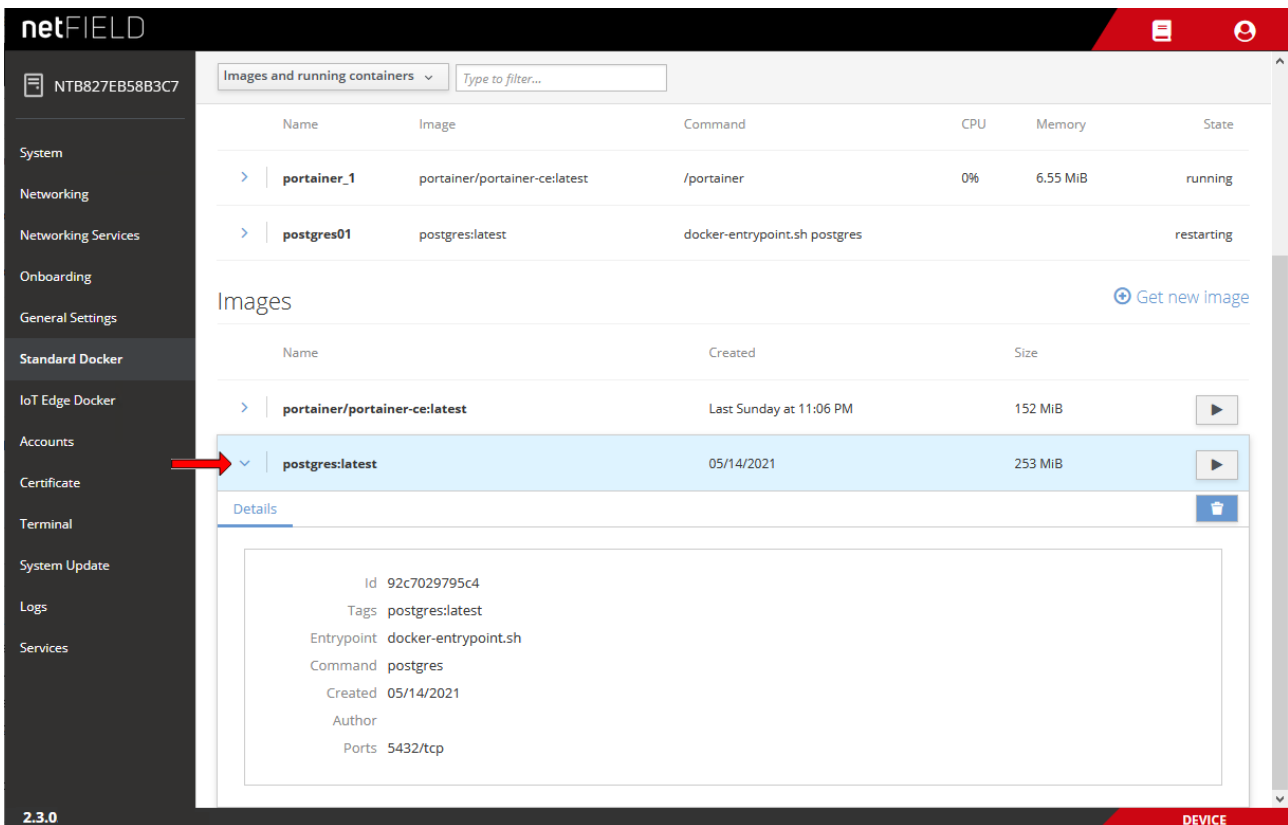
Figure 62: Run Image dialog



#### Note:

For information about the configuration parameters and environment variables that the container requires, consult the documentation or description of the image on Docker Hub.

- To expand a box showing concise image details, or to display a control button to delete it, click on the blue > arrow icon on the left of the image in the list:



The screenshot shows the netFIELD interface with a sidebar on the left containing various system settings. The main area displays 'Images and running containers' with a table of containers and a section for 'Images'. The 'Images' section is expanded to show details for the 'postgres:latest' image. A red arrow points to the blue expand/collapse arrow on the left of the image list item.

Name	Image	Command	CPU	Memory	State
> portainer_1	portainer/portainer-ce:latest	/portainer	0%	6.55 MiB	running
> postgres01	postgres:latest	docker-entrypoint.sh postgres			restarting

Name	Created	Size
> portainer/portainer-ce:latest	Last Sunday at 11:06 PM	152 MiB
▼ postgres:latest	05/14/2021	253 MiB

Details

```
Id 92c7029795c4
Tags postgres:latest
Entrypoint docker-entrypoint.sh
Command postgres
Created 05/14/2021
Author
Ports 5432/tcp
```

Figure 63: Expand image details

- To manage an image, click on it in the list.

➤ A page featuring detailed information opens:

The screenshot shows the netFIELD interface for the 'postgres:latest' image. The sidebar on the left lists various system settings, with 'Standard Docker' selected. The main area displays the image details and a table of containers using the image.


**Image Details:**



- Id: 92c7029795c4
- Tags: postgres:latest
- Entrypoint: docker-entrypoint.sh
- Command: postgres
- Created: 05/14/2021
- Author:
- Ports: 5432/tcp

**Used by Containers:**

Name	Image	Command	CPU	Memory	
drunk_babbage	postgres:latest	docker-entrypoint.sh ...			▶
postgres02	postgres:latest	docker-entrypoint.sh ...	0%	1.23 MiB	▶
postgres01	postgres:latest	docker-entrypoint.sh ...			▶

Figure 64: Image details

Here you can also start a new container for the image (by clicking the **Run** button in the header) or delete the image altogether (by clicking the  button in the header).

The **Used by Containers** area shows the containers that are running on the image (you can create more than one container of the same image), and the resources they consume. You can start or stop a container with the  and  buttons, or open the details page of the container by clicking on it in the list.

- To go back to the **Standard Docker** overview page, click the blue **Show all images** link in the page header.



#### Note:

The Standard Docker can also be managed by using Docker commands on the embedded **Terminal** page of the Local Device Manager (see section *Terminal* [▶ page 105]) or via SSH client connection (e.g. with PuTTY). For examples (e.g. “Docker Compose” support), see section *Useful CLI commands and parameters in Terminal* [▶ page 119].

You can also use the **Portainer.io** container as an additional tool for managing your Standard Docker images and containers. The Portainer.io provides a well-documented web-based management GUI that can be deployed here in the Standard Docker like any other container from the Docker Hub.

## 5.8 IoT Edge Docker

On the **IoT Edge Docker** page, you can monitor the Docker images and containers that were deployed from the netFIELD Cloud via the netFIELD Portal.

Note that you have to “onboard” your device (see section *“Onboard” (register) device in netFIELD Cloud* [▶ page 37]) before you can access this page.

Note also that you have only limited control over the images and containers here (i.e. you cannot download, configure, start or stop them here), because they are managed exclusively from the netFIELD Cloud, respectively netFIELD Portal (where you can e.g. define environment variables for a container before or after its deployment). This distinguishes the IoT Edge Docker from the Standard Docker, which allows the parameterization of containers before they are started (see section *Standard Docker* [▶ page 88]).

Here you can, however, change the limits of the resources (memory and CPU priority) that your application container is allowed to consume on the device.

You can also “remove” an obsolete container image here, but only if you have deleted it in the Device Manager of the portal beforehand. (If you delete an image only locally on the device without having deleted it in the portal beforehand, the image will be automatically deployed again).



**Note:**

The network address settings of the IoT Edge Docker can be managed under **General Settings > Docker Network Settings** (see section *Docker Network Settings* [▶ page 81]).

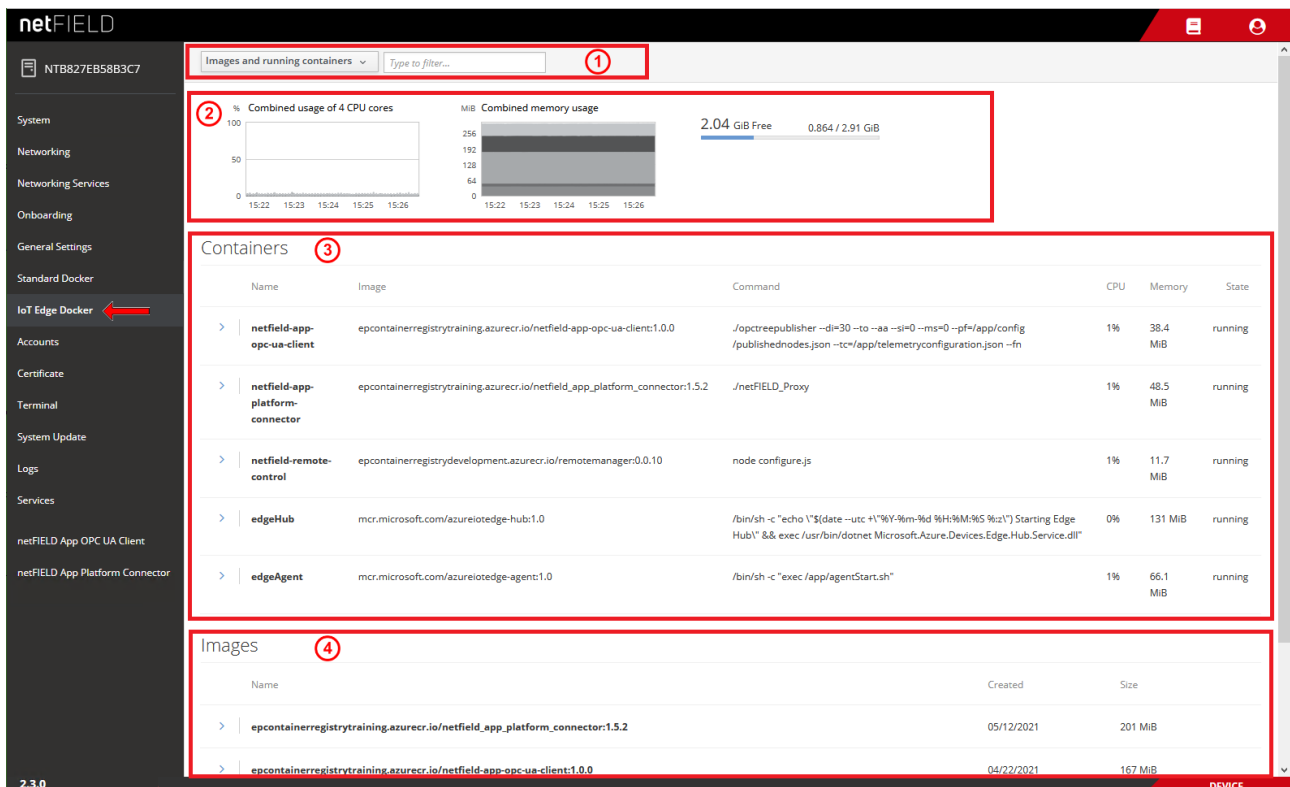


Figure 65: IOT Edge Docker

**Note:**

The *edgeHub* and *edgeAgent* are Microsoft images/containers (called “modules” in Microsoft terms) that make up the Azure IoT Edge runtime, which is necessary for connecting your device to the netFIELD Cloud (which uses the Azure cloud).

The *edgeAgent* is automatically downloaded and instantiated on the device after onboarding; the *edgeHub* is automatically downloaded and instantiated when you deploy a container from the portal for the first time.

**Filter options in header**

The elements in the header (1) allow you to filter the display of containers and images.

You can choose in the drop-down list:

- **Images and running containers** – All downloaded Docker images and currently running containers are displayed (default).
- **Everything** - All Docker images and containers are displayed (including stopped containers).

Use the **Filter** field to display only certain containers.

**Graphs**

The graphs (2) show you the load of the containers on the system resources.

**Combined usage of 4 CPU cores:** Load of the containers on the CPUs.

**Combined memory usage:** Load of the containers on the memory.

The graph in the upper right corner shows the amount of mass storage memory taken by the images and containers (blue bar) and the amount of mass storage left available.



## Containers

The **Containers** area (3) lists the container instances of the Docker images according to your Filter options settings in the header (1).

- To expand a box showing concise container details, or to display a control button to restart it, click on the blue > arrow icon on the left:

The screenshot shows the netFIELD interface with a sidebar on the left and a main content area. The sidebar includes sections like System, Networking, and IoT Edge Docker. The main content area displays system metrics (CPU and memory usage) and a list of containers. The 'netfield-app-opc-ua-client' container is selected, and its details are expanded, showing fields like ID, Created time, Image, Command, and State. A 'Restart' button is visible next to the container name.

Name	Image	Command	CPU	Memory	State
> mosquito	eclipse-mosquitto:1.6	/docker-entrypoint.sh /usr/sbin/mosquitto -c /mosquitto/config/mosquitto.conf	0%	2.31 MiB	running
> netfield-app-opc-ua-client	epcontainerregistrytraining.azurecr.io/netfield-app-opc-ua-client:1.0.0	.\optreepublisher --di=30 --to --aa --si=0 --ms=0 --pf=/app/config/publishednodes.json --tc=/app/telemetryconfiguration.json --fn	1%	63.1 MiB	running
> netfield-app-platform-connector	epcontainerregistrytraining.azurecr.io/netfield_app_platform_connector:1.5.2	.\netFIELD_Proxy	1%	42.8 MiB	running
> netfield-remote-control	epcontainerregistrydevelopment.azurecr.io/remotemanager:0.0.10	node configure.js	2%	11.5 MiB	running
> edgeHub	mcr.microsoft.com/azureiotedge-hub:1.0	/bin/sh -c "echo \"\$(date --utc +%Y-%m-%d %H:%M:%S %z)\""	0%	124 MiB	running

Figure 66: Container details expanded

- To display more details of the container, click on it in the list.

- A page featuring detailed information including a “console output” opens. Here you can also restart the container or change its resource limits:

The screenshot displays the netFIELD IoT Edge Docker interface. The left sidebar shows navigation options: System, Networking, Networking Services, Onboarding, General Settings, Standard Docker, IoT Edge Docker (selected), Accounts, Certificate, Terminal, System Update, Logs, Services, netFIELD App OPC UA Client, and netFIELD App Platform Connector. The main content area shows details for the container 'netfield-app-opc-ua-client'. It includes fields for ID, Created time, Image, Command, State, Restart Policy, IP Address, CIDR Suffix, Gateway, MAC Address, Volumes, Memory usage (61.9 MiB), and CPU usage (1% / 1024 shares). A 'Restart' button is visible in the top right of the container details. A console output window is open, showing error logs related to connecting to a broker.

Figure 67: Container parameters


- To go back to the **IoT Edge Docker** overview page, click the blue **Show all containers** link in the page header.

## Images

The **Images** area (4) lists the Docker images that were deployed from the netFIELD Portal.



### Note:

To remove an image and its container from the device, you must first delete the container in the **Device Manager** of the portal. If you delete it only locally (i.e. here on the IoT Edge Docker page by clicking the  button) while the container is still “deployed” from the portal, the image will be automatically downloaded to the device again.

- To expand a box showing concise image details, or to display a control button to delete it, click on the blue > arrow icon on the left:

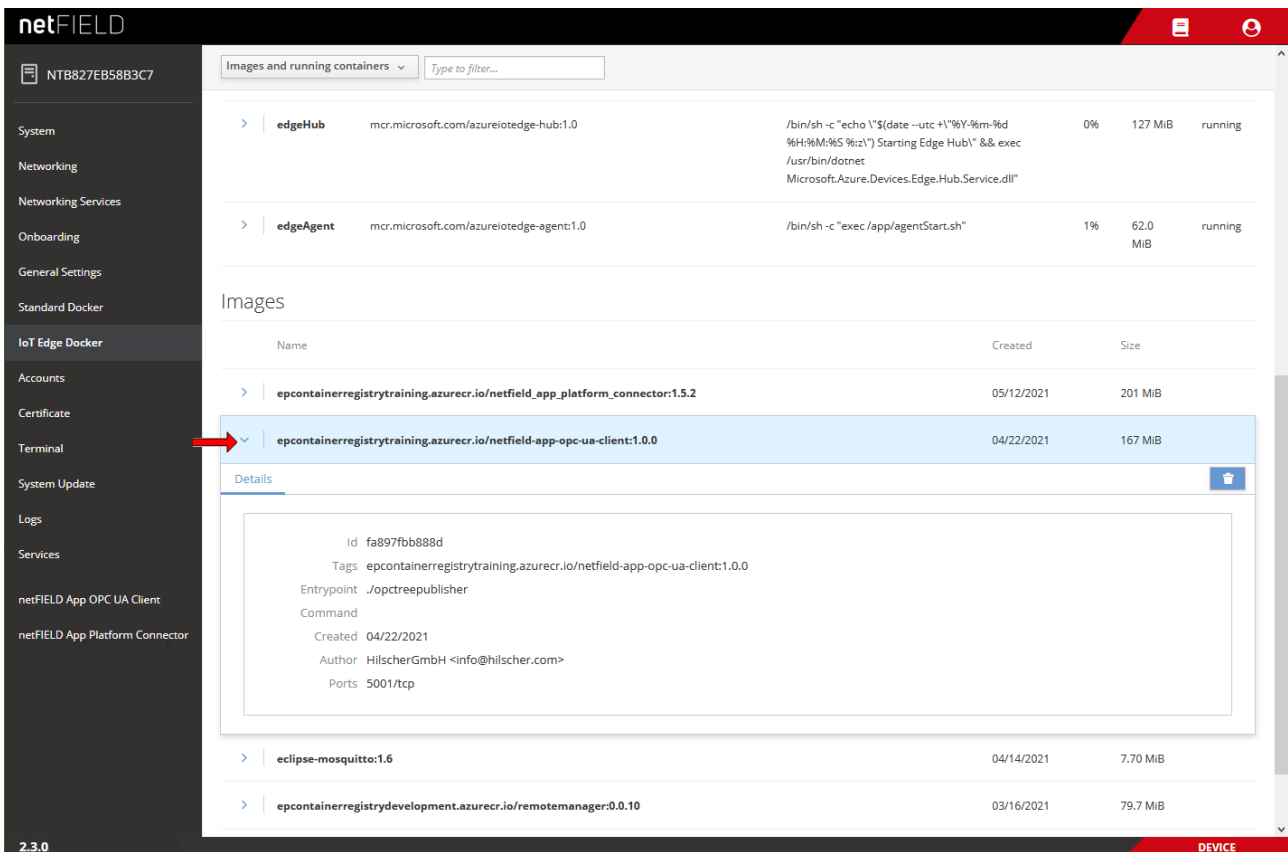


Figure 68: IoT image expanded

- To show more details of an image, click on it in the list.

➤ A page featuring detailed information opens:

The screenshot shows the netFIELD IoT Edge Docker interface. The sidebar on the left contains navigation options: System, Networking, Networking Services, Onboarding, General Settings, Standard Docker, IoT Edge Docker (highlighted), Accounts, Certificate, Terminal, System Update, Logs, Services, netFIELD App OPC UA Client, and netFIELD App Platform Connector. The main content area displays the details of a Docker image: epcontainerregistrytraining.azurecr.io/netfield-app-opc-ua-client:1.0.0. The details include: Id: fa897fbb888d, Tags: epcontainerregistrytraining.azurecr.io/netfield-app-opc-ua-client:1.0.0, Entrypoint: ./opctreepublisher, Command: (empty), Created: 04/22/2021, Author: HilscherGmbH <info@hilscher.com>, and Ports: 5001/tcp. Below the details is a 'Used by Containers' table with the following data:

Name	Image	Command	CPU	Memory
netfield-app-opc-ua-client	epcontainerregistrytrainin...	./opctreepublisher...	1%	63.2 MiB

Figure 69: Details of netFIELD Proxy image

Here you can delete the image by clicking the  button.

The **Used by Containers** area shows the containers that are running on the image, and the resources they consume. You can open the details page of the container by clicking on it in the list.

- To go back to the **IoT Edge Docker** overview page, click the blue **Show all images** link in the page header.



#### Note:

The IoT Edge Docker can also be managed (with the same limitations as in the UI) by using docker commands with the CLI in the Terminal.

See section *Useful CLI commands and parameters in Terminal* [▶ page 119] for examples.

## 5.9 Accounts

On the **Accounts** page, you can manage the user accounts of the netFIELD OS.

You can create new users and define passwords and access right (“roles”).

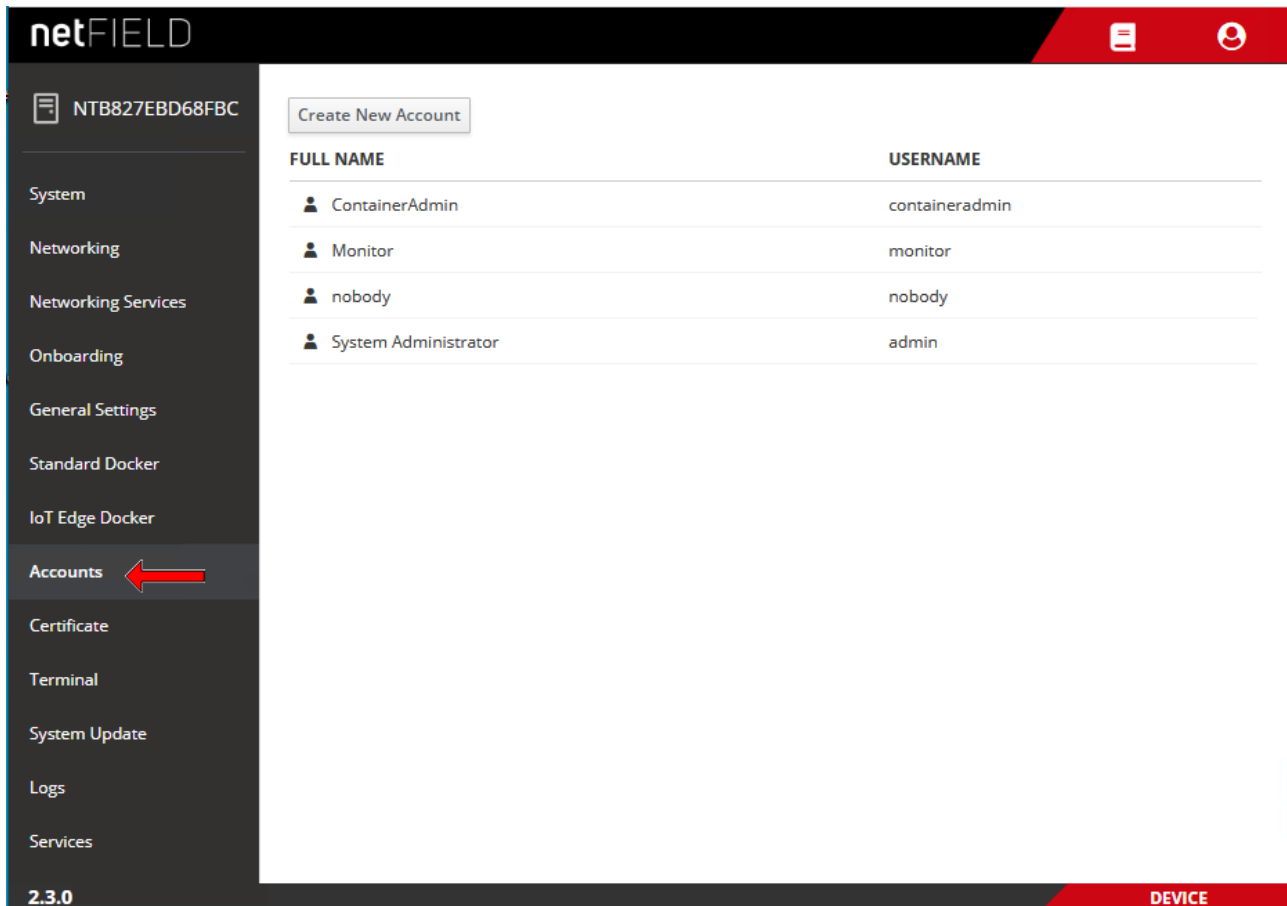


Figure 70: Accounts

- To create a new user account, click on the **Create New Account** button.
- The **Create New Account** dialog opens:

Figure 71: Create new account

- Fill in the form, then click **Create** button.

- To configure an account (e.g. assign roles, change password or lock account), click on the name in the list.
- The configuration dialog for the account opens:

The screenshot shows the 'Accounts > ContainerAdmin' configuration page. At the top, there are 'Terminate Session' and 'Delete' buttons. The main form contains the following fields and options:


- Full Name:** ContainerAdmin
- User Name:** containeradmin
- Roles:**
  - Server Administrator
  - Container Administrator
- Last Login:** Never
- Access:**
  - Lock Account (with a link 'Never lock account')
- Password:**
  - Buttons: 'Set Password', 'Force Change'
  - Link: 'Never expire password'

Below the main form is a section for 'Authorized Public SSH Keys' with a '+' button. The text below this section reads: 'There are no authorized public keys for this account.'

Figure 72: Edit account



#### Note:

You can open the configuration dialog for your currently used account (i.e. the account you are currently logged in with) also by selecting  > **Account Settings** in the toolbar.

#### Roles

The **Server Administrator** has full access rights to all functions of the netFIELD OS (including Standard Docker and IoT Edge Docker).

The **Container Administrator** has access to the **Standard Docker** and **IoT Edge Docker**, but is otherwise not allowed to make any changes to the netFIELD OS settings.

The **Container Administrator** can download container images in the **Standard Docker**, and can also start and stop the containers. Note that the containers running in the **IoT Edge Docker** are deployed and managed exclusively from the netFIELD Cloud, respectively netFIELD Portal. As **Container Administrator** you can, however, “clean” a netFIELD container image from the device after it has been deleted in the *Device Manager of the Portal*. (If you delete an image only locally on the device without having deleted it in the Portal beforehand, the image will be automatically deployed again).

If you assign **neither role** to an account, the user has only “read” access to the netFIELD OS functions respectively to the host configuration. Furthermore, a user without a role will have no access to the Standard Docker or to IoT Edge Docker (not even “read” access). Note, however, that this user will have access to the plug-in dashboards of the netFIELD application containers in the Local Device Manager.

### Authorized Public SSH Keys

This area lists the SSH keys assigned to this account.

Click on the  button to add an SSH key.



**Note:**

With a SSH key pair (private and public key), you can login (e.g. with a terminal program like PuTTY) to your account via netFIELD OS SSH shell by using your private key. The password is replaced by the private key, and you only have to specify a valid netFIELD OS account name (e.g. “*admin*”) for authentication when you login.

---

## 5.10 Certificate

On the **Certificate** page, you can manage the web server certificate of the device's web UI and turn it into a trusted one. You can display details of your currently installed certificate and upload a new certificate and the corresponding private key file in \*.pem format to the netFIELD OS.

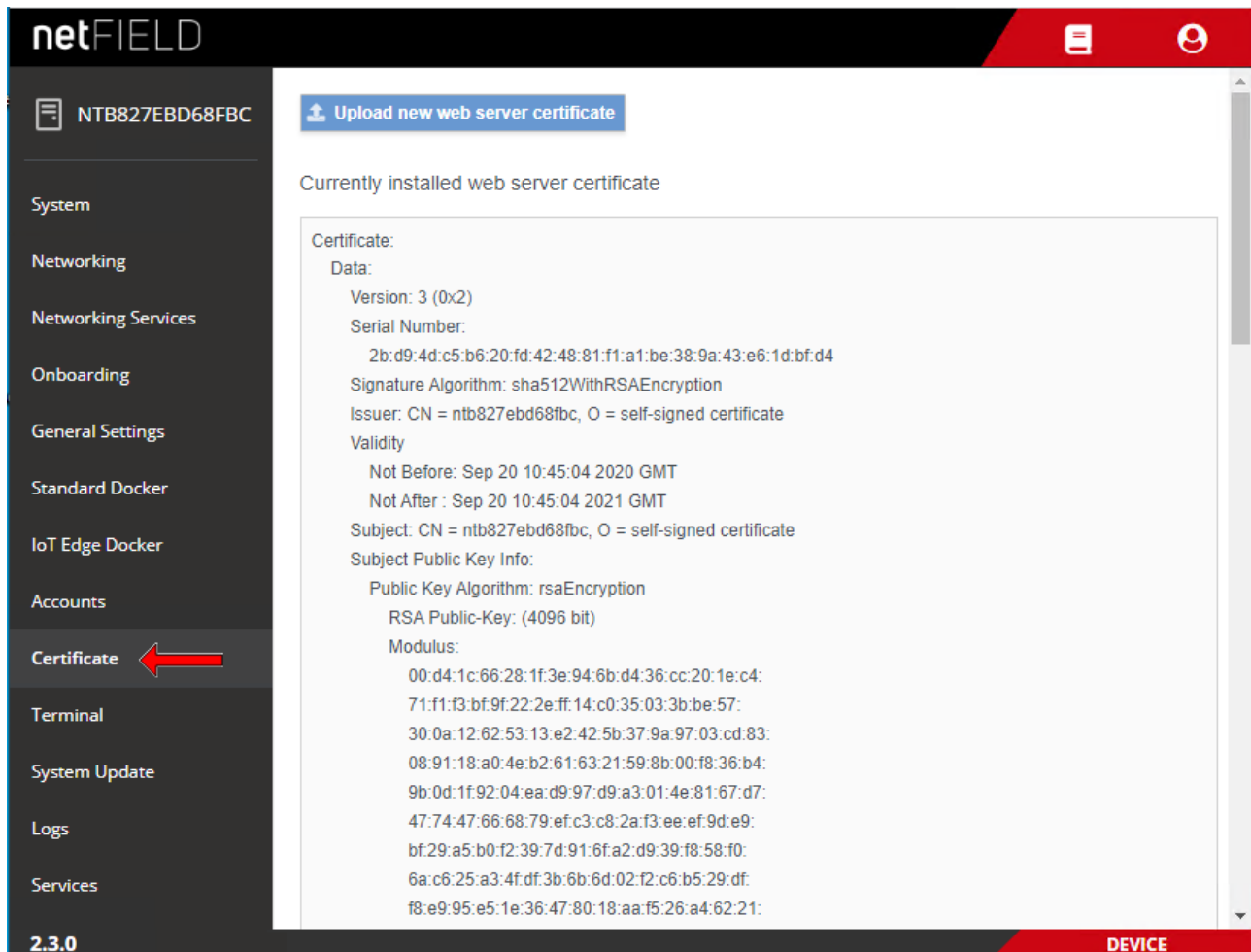


Figure 73: Web Server Certificate page



### Note:

The netFIELD OS contains a certificate issued by Hilscher. Note that the automatically created certificate is valid for one year. You can upload your own certificate to the netFIELD OS here. The corresponding root certificate should be rolled out on each of your PC/devices that you use for connecting to the netFIELD OS.



## 5.11 Terminal

The “in-browser” **Terminal** page allows command line-based administration of the netFIELD OS. Note that this is for Linux experts only.

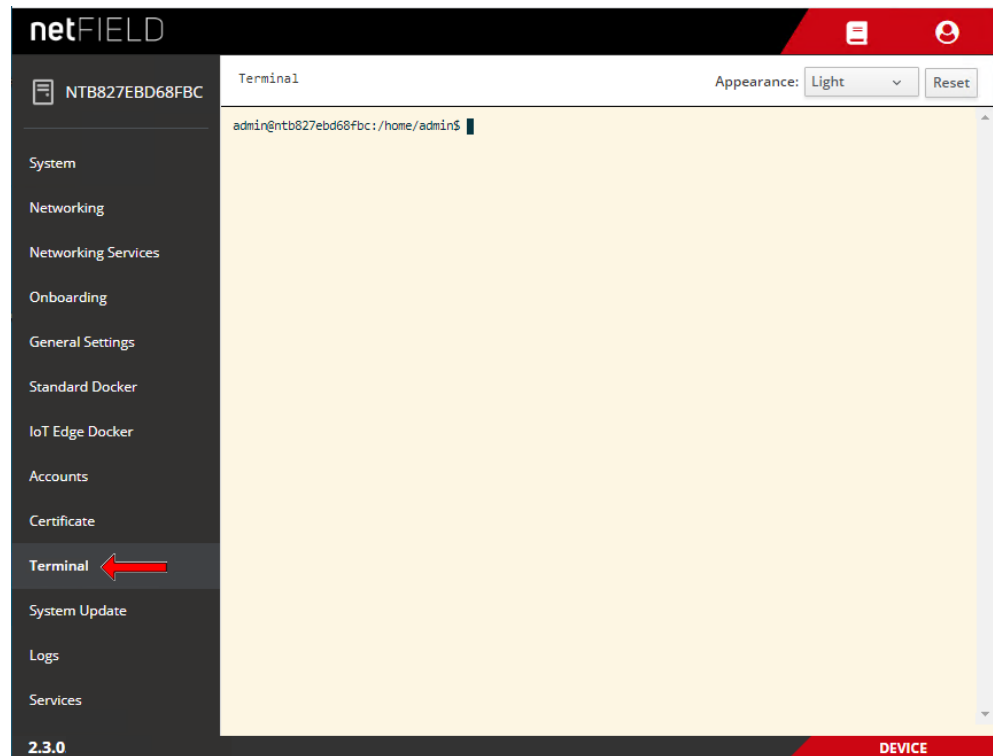


Figure 74: Terminal



### Note:

As an alternative, you can also access the netFIELD OS command line interface by using an external Terminal client program (like e.g. PuTTY) via Ethernet and SSH (standard port 22) or via serial **Console** interface (UART-to-USB, see section *Console interface* [▶ page 20]). Note that file transfer via SCP protocol is also supported.

For working with root privileges in the CLI, use “`sudo`”.

Examples of commands and parameters are provided in section *Useful CLI commands and parameters in Terminal* [▶ page 119].

## 5.12 System Update

You can update the netFIELD operating system (netFIELD OS) by simply uploading an `swu` update file on the **System Update** page of the Local Device Manager.

You can also perform an OS “Recovery” here by uploading a recovery image (also in `swu` format) instead of an update file.



### Important:

Be aware of the difference between an OS *update* and a *recovery*: In an *update*, bug fixes and/or new functions will be added to the existing netFIELD OS. Your device’s configuration settings, containers, user accounts, passwords and its cloud registration (“onboarding”) will thereby be preserved.

In a *recovery*, the currently installed OS will be fully replaced by the new recovery image. This restores the “factory settings” of the device, which means that individual configurations settings, containers, user accounts and passwords will be lost. Therefore, you will have to reconfigure and “onboard” your device again after a *recovery*.

---

Note that it is not possible to “downgrade” your OS; i.e. the installation of an OS version that is “older” than the currently installed OS version will be denied.

**Note:**

The netFIELD OS update process requires a certain amount of free RAM on your device. If you are running application containers with extensive memory usage, we recommend you to stop these containers before you start the update process, in order to “free” the required RAM for the process. You can restart the containers after having finished the OS Update.

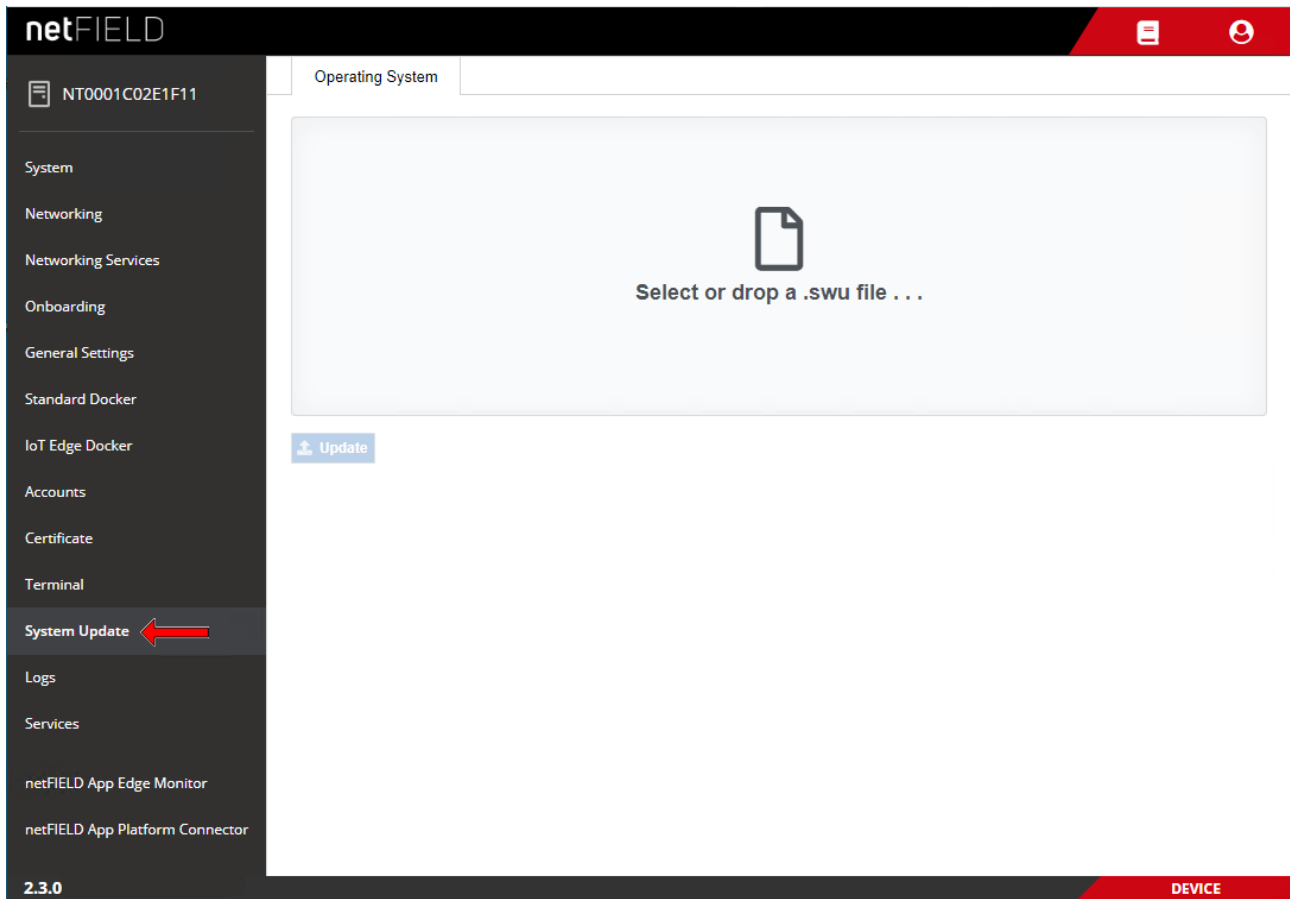


Figure 75: OS update page

**Note:**

As an alternative to using the Local Device Manager for your OS update, it is also possible to update your device’s OS from the netFIELD Portal in the cloud. However, this requires access to the portal (i.e. an account) and the deployment of the *netFIELD App Platform Connector* on your device.



- The image is uploaded to the device. This might take a few minutes. After uploading has been finished, the following screen appears:

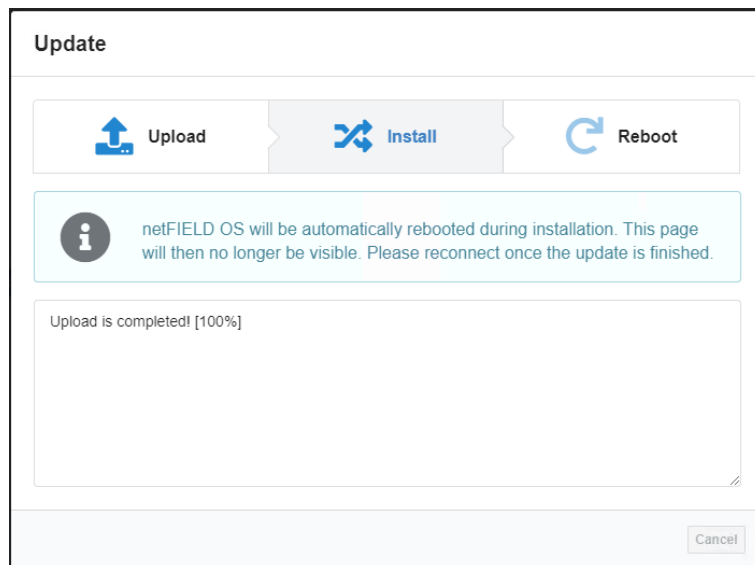


Figure 77: Upload finished message



**Note:**

If you receive an error message, this may be because of a lack of sufficient free storage space on the flash memory. To remedy this, restart the netFIELD OS, then try again. The restart will clear remanent data from the flash memory and provide sufficient space for buffering the update file.

The installation process (i.e. the actual update of the OS) is automatically started. The device reboots and closes the LAN connection.

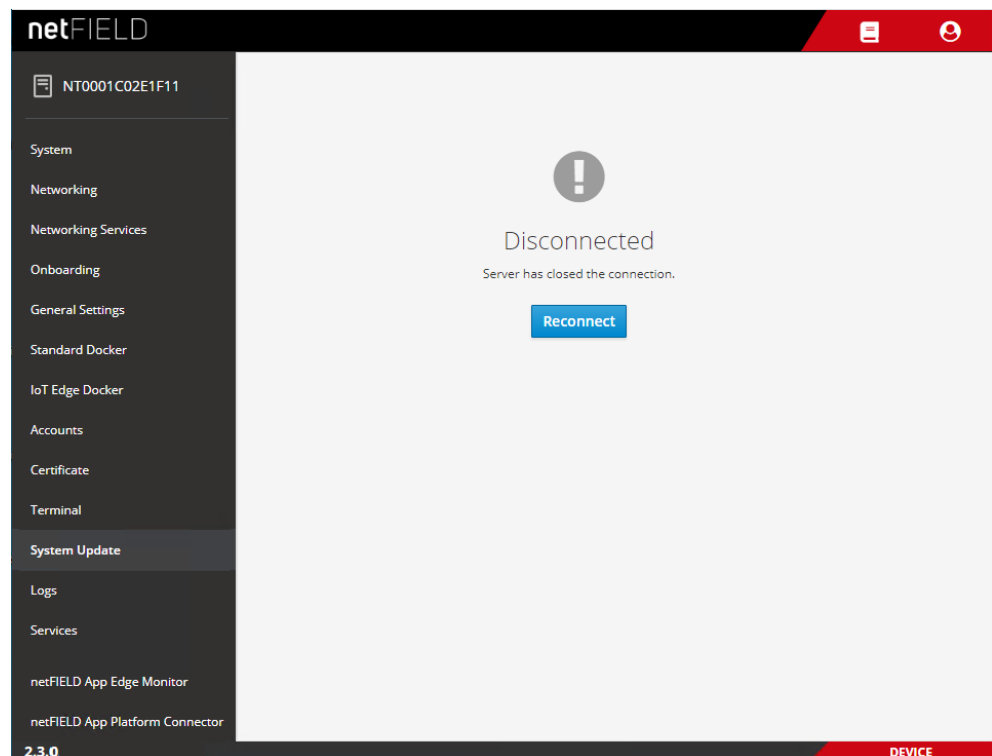


Figure 78: OS update "Disconnected" message

- Click **Reconnect** button.
- ⇒ You have updated the OS of your device. You can now sign-in again with your usual login credentials. The new firmware version is indicated in the bottom left corner of the **Local Device Manager** screen.

**Note:**

If you have performed a *recovery* (by uploading and installing a recovery image) instead of an *update*, all configuration settings have been deleted, and you now must commission the device again (see chapter *Commissioning and first steps* [▶ page 23]).

---

## 5.13 Logs

The **Logs** page allows you to monitor the messages produced by the `systemd journal`.

- In the drop-down lists in the header, you can filter the messages by time/date, **Severity** (type) and **Service** (i.e. the “service” that issued the message).
- Click on a message in the list to display the information in full detail.

The screenshot shows the netFIELD interface with the following components:

- Header:** netFIELD logo, user profile icon, and date filter (February 28, 2022).
- Filters:** Severity (Warning and above) and Service (All).
- Sidebar:** Navigation menu with 'Logs' highlighted by a red arrow.
- Log Table:**

FEBRUARY 28, 2022		
10:33	curl: (28) Connection timed out after 1000 milliseconds	cockpit-bridge
10:33	[237 bytes of binary data]	cockpit-bridge
10:33	Dload Upload Total Spent Left Speed	cockpit-bridge
10:33	% Total % Received % Xferd Average Speed Time Time Time Current	cockpit-bridge
▲ 10:33	1646040808: Socket error on client 93b6b88f9a784fa2a4959b2c9b9bd29f, discon...	972a6d73439a
▲ 10:33	+ echo ... done removing configui from host fs	33467e598fe0
▲ 10:33	+ rm -r /host/share/cockpit/netfield-app-opc-ua-client/	33467e598fe0
▲ 10:33	+ echo removing configui from host fs ...	33467e598fe0
10:33	curl: (28) Connection timed out after 1001 milliseconds	cockpit-bridge
10:33	[158 bytes of binary data]	cockpit-bridge
10:33	Dload Upload Total Spent Left Speed	cockpit-bridge
10:33	% Total % Received % Xferd Average Speed Time Time Time Current	cockpit-bridge
▲ 10:32	<6> 2022-02-28 09:32:37.623 +00:00 INFO Connected to mqtt broker	4306add4e4a4
▲ 10:32	1646040757: New client connected from 10.253.0.1 as auto-B7689436-A06D-5D9...	972a6d73439a
▲ 10:32	1646040756: New connection from 10.253.0.1 on port 1883.	972a6d73439a
▲ 10:32	<6> 2022-02-28 09:32:36.573 +00:00 INFO trying to connect to: tcp localhost 1883	4306add4e4a4
- Footer:** 2.3.0 and DEVICE status.

Figure 79: Logs

## 5.14 Services

### Overview

The **Services** page allows you to manage and monitor services of the netFIELD OS.



#### Important:

Note that this feature is for expert users only! Changing the state or the startup settings of a service here can lead to malfunctioning of the netFIELD OS respectively of your device!

NAME	DESCRIPTION	STATE	AUTOMATIC STARTUP
alsa-restore.service	Save/Restore Sound Card State	inactive (dead)	Static
alsa-state.service	Manage Sound Card State (restore and store)	inactive (dead)	Static
apparmor.service	AppArmor initialization	active (exited)	Enabled
auditd.service	Security Auditing Service	active (running)	Enabled
autovt@.service	autovt@.service Template		
avahi-daemon.service	Avahi mDNS/DNS-SD Stack	active (running)	Enabled
aziot-certd.service	Azure IoT Certificates Service	active (running)	Disabled
aziot-edged.service	Azure IoT Edge daemon	active (running)	Enabled
aziot-identityd.service	Azure IoT Identity Service	active (running)	Disabled
aziot-keyd.service	Azure IoT Keys Service	active (running)	Disabled
aziot-tpmd.service	Azure IoT TPM Service	inactive (dead)	Disabled
blk-availability.service	Availability of block devices	inactive (dead)	Disabled
bluetooth-start.service	Run hciattach when HCI UART device becomes available	inactive (dead)	Disabled
bluetooth.service	Bluetooth service	inactive (dead)	Enabled
busybox-klogd.service	Kernel Logging Service	inactive (dead)	Disabled
busybox-syslog.service	System Logging Service	inactive (dead)	Disabled
cifxeth.service	LSB: Raise and configure the netX based virtual ethernet interfaces	inactive (dead)	Static

Figure 80: Services page

- (1) Click the tabs in the header to select a service type.
- (2) In the filter field, you can perform a text search for name and description of a service.  
To remove the filter, delete the text in the field.
- (3) In the drop-down list, you can filter the services by their automatic startup setting; i.e. **Static**, **Enabled** and **Disabled**.
- (4) List of services showing their current states and automatic startup settings.



## Service details/settings page

- Click on a service in the list to display further information (including the service logs) and/or to change its running state or startup settings.
- The details/settings page of the service opens:

The screenshot shows the 'Service details and settings page' for the 'aziot-edged.service'. The page is titled 'Services > aziot-edged.service' and displays the following information:

- Azure IoT Edge daemon**
- Status:** active (loaded). Buttons: Stop, Restart (dropdown).
- Running Since:** February 25, 2022 1:59 PM
- Automatic Startup:** enabled. Button: Disable (dropdown).
- Path:** /lib/systemd/system/aziot-edged.service
- Requires:** sysinit.target, system.slice, aziot-edged.mgmt.socket, aziot-edged.workload.socket
- Wants:** iotedge-docker.service, network-online.target
- Wanted By:** multi-user.target
- Consists Of:** aziot-edged.workload.socket, aziot-edged.mgmt.socket
- Conflicts:** shutdown.target
- Before:** multi-user.target, shutdown.target
- After:** systemd-journald.socket, sysinit.target, system.slice, aziot-edged.mgmt.socket, iotedge-docker.socket, basic.target, aziot-edged.workload.socket, network-online.target, iotedge-docker.service
- Triggered By:** aziot-edged.workload.socket, aziot-edged.mgmt.socket

**SERVICE LOGS**

MARCH 1, 2022

10:01	2022-03-01T09:01:23Z [INFO] - [mgmt] - - - [2022-03-01 09:01:23.645152486 UTC] "GET /modules?api-version=2020-07-07 HTTP/..."	aziot-edged
10:01	2022-03-01T09:01:19Z [INFO] - Edge runtime is running.	aziot-edged

Figure 81: Service details and settings page

The buttons in the **Status** section allow you to **Stop/Start** or **Restart/Reload** the service.

The drop-down button in the **Automatic Startup** section allows you to configure the startup behavior of the service like e.g. "masking" it in order to prevent the service from running.

Other services that are related to the service (e.g. required services displayed under **Requires**) are displayed as clickable links.

The log messages of the service are displayed under **SERVICE LOGS** in the footer.

## Managing Timers

On the **Timers** tab, you can display existing timers and create new timer units. A timer allows you to execute a certain command at a certain time.

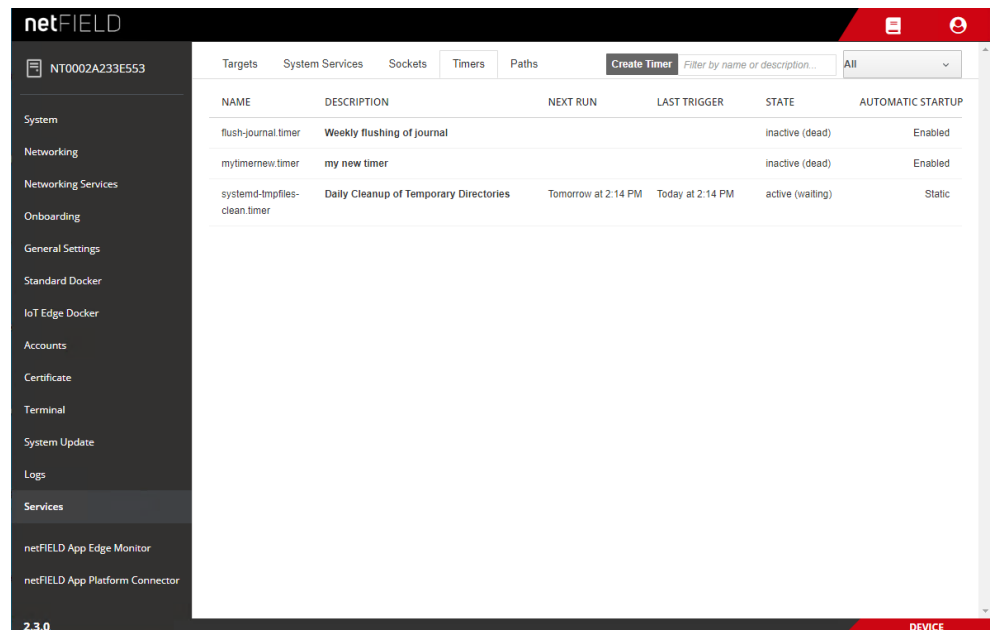


Figure 82: Service types: Timers

- Click on a timer in the list to display further information and/or to change its running state or startup settings.
- To configure a new timer, click **Create Timer** button in the header.
- The **Create Timers** dialog opens:

**Create Timers**

Service name: flush-journal

Description: Weekly flushing of journal

Command: systemd-journal-flush.service

Run: At specific time

Repeat Weekly: Monday

Time: 11:00

Buttons: Cancel, Save

Figure 83: Create timer dialog

- In the **Command** field, enter the name of the service that shall be triggered by the timer.
- Set all desired parameters, then click **Save** button.



### Note:

Note that you can create but cannot delete timers here. (You can however stop a timer here by opening its details/settings page, then clicking the **Stop** button in the **Status** section).

To remove a timer completely, you have to use the **Terminal** to delete it manually in the corresponding `systemd` configuration.

## 6 Good to know

### 6.1 Device recovery via USB

#### Overview

This section describes how to reset the netFIELD OS of your device by installing a “recovery” image firmware from a USB stick.

A device recovery via USB can be necessary if the netFIELD OS has become instable or corrupted, or if you have “locked yourself out” of the **Local Device Manager** because you have deactivated or misconfigured its Ethernet interfaces or if you have forgotten the administrator’s password.

Note that it is not possible to “downgrade” your OS; i.e. the installation of an OS version that is “older” than the currently installed OS version is not supported.



#### Important:

In a device recovery, all configuration settings, user accounts and deployed containers of the current netFIELD OS will be deleted. This means that you will have to commission and configure your device again after the recovery procedure.

Note also that the firmware of the netX communication controller will not be affected by the recovery.

---

#### Requirements

- FAT32-formatted USB stick with a minimum of 500 MByte storage capacity.



#### Note:

USB sticks with a storage capacity of more than 64 GByte cannot be easily formatted under Windows in FAT32. If you intend to use such a high-capacity stick, use a tool like e.g. HP USB STICK FORMAT to format the stick under Windows.

---

- You have downloaded the recovery image from Hilscher to your local PC (see below for instructions).
- You have physical access to the device (in order to plug-in the USB stick).

### Step-by-step instructions

1. Download the zip archive containing the recovery image from Hilscher to your local PC and unpack it.
  - Go to the *netFIELD Software Overview* page <https://kb.hilscher.com/x/sSAfBw> and navigate to the latest *netFIELD OS Releases* version. On the *netFIELD OS Version* page, scroll down to the *netFIELD Compact* section. In the *Software* table of the *netFIELD Compact* section, look for the *Recovery (factory reset) via USB memory stick* entry and download the linked `recovery.zip` file.
  - Unpack the downloaded zip archive on your local PC.
  - The unpacked `recovery` archive contains the following files, which you will later have to copy to the USB stick (after having formatted the stick):

- boot
- firmware
- Image
- VERSION

2. Format and rename USB stick.
  - Connect the USB stick to your Windows PC.
  - Open the Windows Explorer.
  - Select the USB stick and choose **Format...** from the context menu.

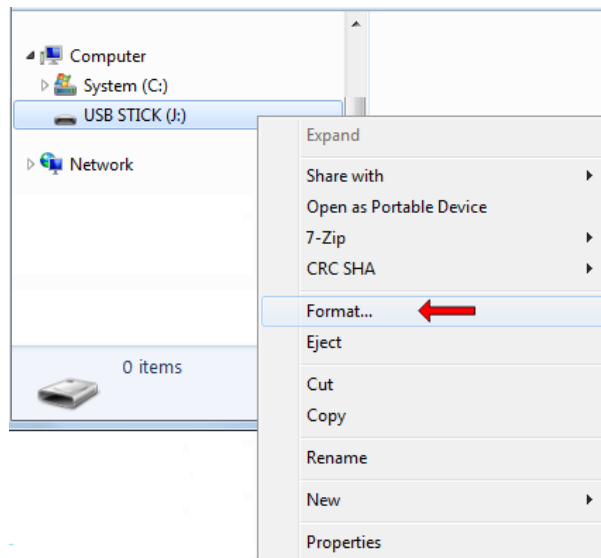


Figure 84: Formatting USB stick

➤ The **Format USB STICK** dialog window opens:

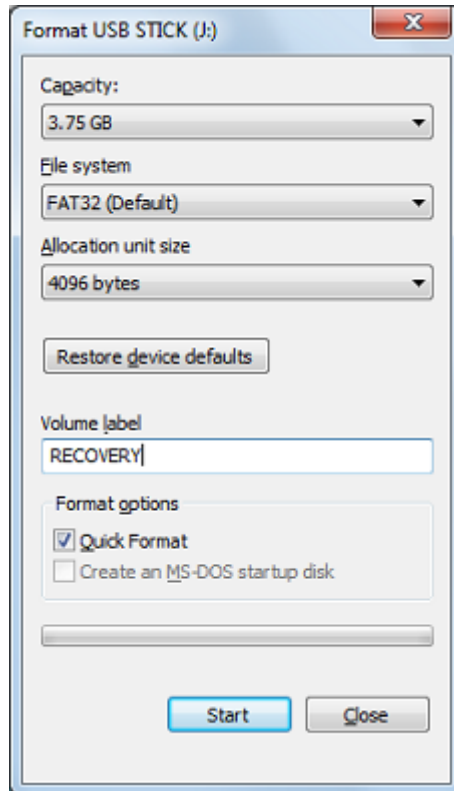


Figure 85: Format USB STICK dialog window

- In the **File system** drop-down list, select **FAT32 (Default)** option.
- In the **Volume label** field, enter the name `RECOVERY`.



#### Important:

The volume label name `RECOVERY` is mandatory. Do not use any other name, otherwise the procedure will fail.

- Under **Format options**, check **Quick Format** option.
- Click **Start** button.
- Acknowledge the warning message with **OK**.
- After formatting is finished, the USB stick is labelled in the Windows Explorer by its new name "RECOVERY".

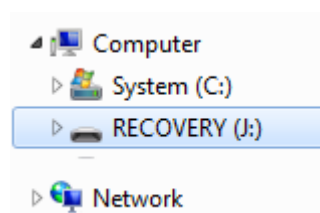


Figure 86: Formatted USB stick

### 3. Copy recovery files onto the USB stick.

- Open the unpacked `recovery` archive folder and copy the `boot`, `firmware`, `Image` and `VERSION` files onto the USB stick.
- ⇒ The USB stick with the copied recovery image files must now contain the following elements:

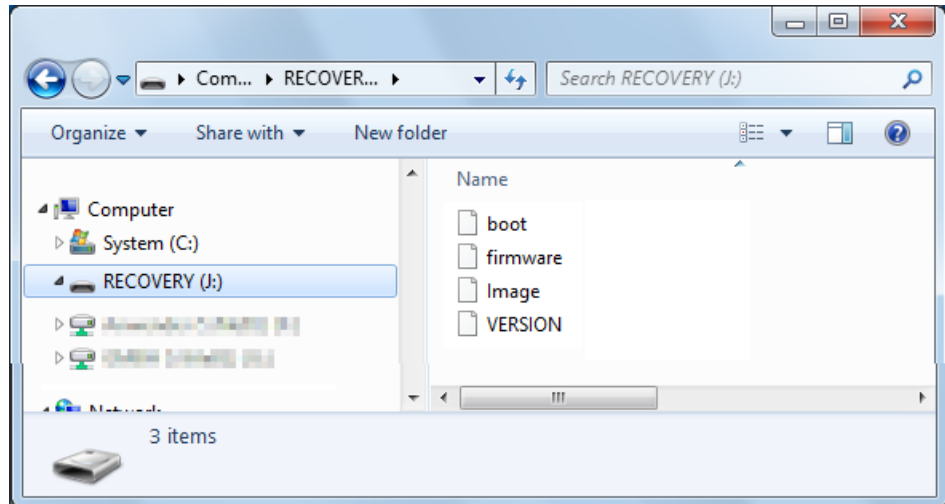


Figure 87: Recovery image on USB stick

- Remove the USB stick from your Windows PC.
- ### 4. Start recovery.
- Plug the prepared USB stick into one of the USB sockets of the device.
  - Start a new power cycle by switching the power off and then on again.
  - ⇒ The device restarts and boots from the connected USB stick. It then installs the new netFIELD OS from the USB stick. This might take a few minutes.
  - Remove the USB stick and start a new power cycle by switching the power off and then on again.
  - ⇒ You have finished the netFIELD OS recovery procedure. The device is reset to its factory settings (ETH1 LAN interface is enabled and set to DHCP, default administrator password is set to `admin`). You can now reconnect to the device as described in chapter *Establish LAN connection and login to Local Device Manager* [▶ page 28].

## 6.2 Useful CLI commands and parameters in Terminal

### 6.2.1 Network Manager

```
sudo nmcli ...
```

### 6.2.2 Show interface status

```
sudo nmcli dev status
```

### 6.2.3 Activate interface

(Re)activate interface, e.g. eth0:

```
sudo nmcli con up ifname eth0
```

### 6.2.4 Docker Compose support for Standard Docker environment

```
docker-compose <commands>
```

#### Examples

Show the version of Docker Compose:

```
docker-compose version
```

Start container(s) via Docker Compose file:

```
docker-compose -file <docker compose file.yml> up -d
```

Stop container(s) via Docker Compose file:

```
docker-compose -file <docker compose file.yml> down
```

### 6.2.5 Manage Standard Docker

```
docker <docker commands>
```

#### Examples

List all created containers of the Standard Docker instance:

```
docker ps
```

List all bridges of the Standard Docker instance:

```
docker network ls
```

### 6.2.6 Manage IoT Edge Docker

```
docker-iotedge <docker commands>
```

#### Example

To list all created containers for the IoT Edge Docker instance:

```
docker-iotedge ps -a
```

## 6.2.7 External storage support using iSCSI

Enable iSCSI service:

```
sudo systemctl enable iscsi-initiator
```

Start iSCSI service:

```
sudo systemctl start iscsi-initiator
```

Target discovery and connection administration:

```
sudo iscsiadm <parameter>
```

Configuration files:

```
initiatorname.iscsi  
iscsid.conf
```

## 6.2.8 Enable/disable SSH Daemon (release port 22)

Disable autostart:

```
sudo systemctl disable sshd.socket
```

Stop SSH Daemon:

```
sudo systemctl stop sshd.socket
```

## 6.2.9 Follow the system log via terminal CLI

```
sudo journalctl -f
```

## 6.2.10 Configure operating mode of serial interface

Set serial interface remanently to RS-485:

```
sudo bash -c 'echo "mode=rs485" > /etc/default/uart1'  
systemctl restart uart1-mode
```

Set serial interface remanently to RS-232:

```
sudo bash -c 'echo "mode=rs232" > /etc/default/uart1'  
systemctl restart uart1-mode
```



## 7 Technical data

Category	Parameter	Value
General	Part number	1918.010
	Article name	NFX8M-D2-N32-010
	Function	IT/OT edge gateway for IIoT, Industry 4.0, Integrated Industry and automation projects with optional netFIELD Cloud connectivity
Processor	CPU	1.8 GHz quad-core ARM Cortex-A53 64Bit (NXP i.MX8M Mini)
Memory	RAM	2 GB LPDDR4
	Storage	32 GB eMMC flash, appr. 50 TBW (Terabytes Written)
Software	Operating system	netFIELD OS, based on Security Enhanced YOCTO Linux
	Standard Docker	Docker for manual and local deployment and maintenance of containers
	IoT Edge Docker	Docker for remote and automatic deployment and maintenance of containers from the netFIELD Cloud
	Local Device Manager	Web-based GUI for local device parameterization
Security	Secure boot	"High Assurance Boot" of signed software only
	Access	HTTPS, TLS
Power supply	Supply voltage	Unregulated +8 V DC ... +36 V DC IN (typical: +24 V DC) <b>Note:</b> Voltages above 36 V may cause damage to the device. Voltages below 8 V may cause a shutdown of the device.
	Current consumption without USB (typical)	110 mA at 24 V DC 220 mA at 12 V DC
	Current consumption with USB (typical)	350 mA at 24 V DC with two USB ports (at 5 V and 500 mA per port, resulting in additional 120 mA per port) <b>Note:</b> Maximum load over all three USB ports is 1 A
	Connector	5.5 x 2.5 mm coaxial input jack with "bajonet" lock
Real-time clock	Buffering	On-board battery with 3V and 120mAh: SBR1632 Lithium Fluorocarbons coin cell, RoHS compliant, 10 years service interval
Ethernet LAN interfaces	Type	ETH1: 1 Gbit/s Ethernet port (eth0 in netFIELD OS) ETH2: 100 Mbit/s Ethernet port (eth1 in netFIELD OS)
	Connector	2 x RJ45 jacks
Additional interfaces	USB	3 x USB 2.0 ports, type-A connector <b>Note:</b> Maximum allowed output current over all USB ports is 1 A
	Console	Serial UART-to-USB interface, Mini USB connector
	Serial port	RS-232 or RS-485 (2-wire/half-duplex), terminal block connector
LED indicators	Edge LED	User-programmable yellow/green duo LED (labelled "LED" on the device)
	Ethernet LAN	<b>LINK</b> (green): Ethernet Link status <b>ACT</b> (yellow): Ethernet Activity status
	Power-on	Orange LED indicating on/off state of the device
Mechanics and housing	Dimensions of device (without plugs and mounting bracket)	112 mm x 84 mm x 25 mm
	Dimensions of device with plugs and mounting bracket	Ca. 167 mm x 118 mm x 27 mm For details, see section <i>Dimensions</i> [▶ page 16]
	Weight of device with mounting bracket	Ca. 350 gr.
	Housing material	Aluminum
	Degree of protection	IP20
	Mounting type	DIN top hat rail mounting or wall mounting (with two M4 screws)

Category	Parameter	Value
Admissible ambient conditions	Ambient temperature	Operation: -20 °C ... +60 °C Storage: -40 °C ... +85 °C
	Relative humidity	Operation: 10 % ... 90 % Storage: 5 % ... 95 %
Compliance	Electromagnetic compatibility (EMC)	EN 55032/5, EN 61000-6-2, EN 61000-6-3
	Safety	EN/UL/IEC 62368-1
Approvals	UL	In preparation
Conformity declarations	CE	Yes
	UKCA	Yes
	FCC	Yes
	RoHS	Yes

Table 19: Technical data netFIELD Compact

## 8 Decommissioning, dismantling and disposal

### 8.1 Dismounting

---

**NOTICE****Risk of Unsafe System Operation!**

To prevent personal injury or property damage, make sure that the removal of the device from your plant during operation will not affect the safe operation of the plant.

---

**CAUTION****Risk of light burns due to hot device**

During operation, high temperatures can occur on the surface of the device and on the metallic connection sockets. If the device was in operation, let it cool down before touching it or use gloves.

---

- Remove all cables from the device.
- Press down the spring-loaded lower rail clamps at the back of the mounting bracket and lift the device from the top hat rail.

## 8.2 Disposal

### Disposal of battery

The product contains a battery that requires special handling when it is replaced or when the device is disposed of after having reached its end-of-life. The contained battery is a

*JL World SBR1632 Lithium Fluorocarbons coin cell battery (or equivalent).*



### Waste electronic equipment

According to European Directive 2006/66/EU, batteries may not be treated as household waste. As a consumer, you are legally obliged to dispose of batteries according to national and local regulations, for instance at the public collection sites for waste batteries.

If desired by the customer, Hilscher takes back this product, including the battery, and disposes of it according to the applicable regulations. Note that transportation costs must be paid for by the customer.

### Removing of battery

If you do not want to return the device to Hilscher, you have to remove the battery and dispose of it properly prior to the disposal of the device.

To remove the battery, proceed as follows:

- Unplug all cables and dismount the device from the DIN top hat rail.
- Detach the device from the mounting bracket by unscrewing the two attachment screws.

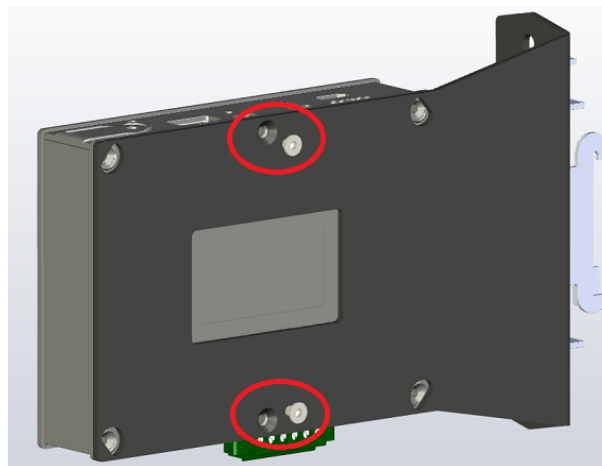


Figure 88: Detach device from mounting bracket

- Open the housing of the device by unscrewing the four fastening screws.

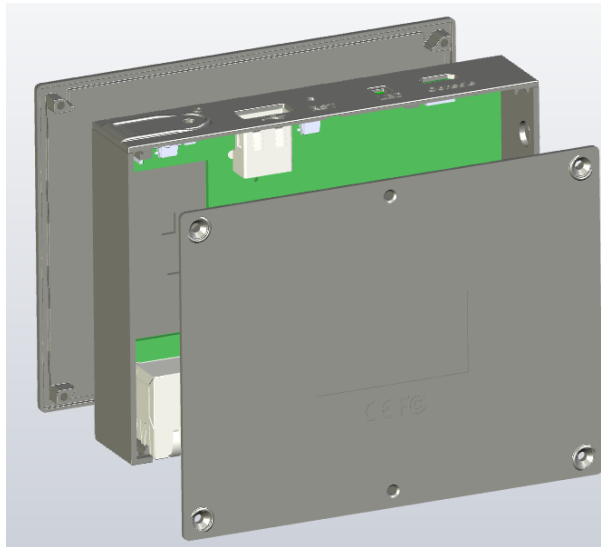


Figure 89: Open device housing

- Remove the battery from the device. The position of the battery is marked in the photo below:

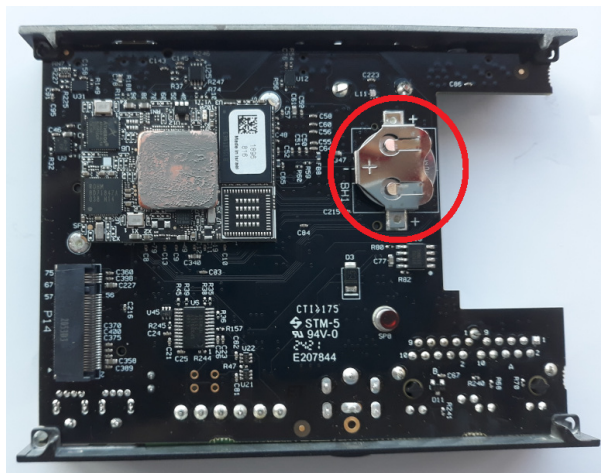


Figure 90: Position of battery in device

## Disposal of device

---



### Waste electronic equipment

---

This product must not be treated as household waste. It must be disposed of at a designated waste electronic equipment collecting point.

---

According to the European Directive 2012/16/EU “Waste Electrical and Electronic Equipment (WEEE)”, this electronic equipment may not be disposed of as household waste. You are legally obliged to dispose of all waste electronic equipment according to national and local regulations.

If desired by the customer, Hilscher takes back this product and disposes of it according to the applicable regulations. Note that transportation costs must be paid by the customer.

## 9 Legal notes

### Terms and conditions

Please read the terms and conditions under <https://www.netfield.io/termsOfUse>.

### netFIELD OS

netFIELD OS is a YOCTO project based Linux operating system released and licensed by Hilscher.

netFIELD OS is released under the

[HILSCHER netFIELD Source Code/Software LICENSE AGREEMENT](#).

netFIELD OS may include 3rd party software or software declared as Open Source. The licensing information of all included software components are provided in textual form in the netFIELD OS under the folder `/usr/share/common-licenses` and subfolders.

Open Source software is released under Open Source License usually.

The official verbatim texts can be read and checked under <https://opensource.org/licenses>.

### Copyright

© Hilscher Gesellschaft für Systemautomation mbH

All rights reserved.

The images, photographs and texts in the accompanying materials (in the form of a user's manual, operator's manual, Statement of Work document and all other document types, support texts, documentation, etc.) are protected by German and international copyright and by international trade and protective provisions. Without the prior written consent, you do not have permission to duplicate them either in full or in part using technical or mechanical methods (print, photocopy or any other method), to edit them using electronic systems or to transfer them. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. Illustrations are provided without taking the patent situation into account. Any company names and product designations provided in this document may be brands or trademarks by the corresponding owner and may be protected under trademark, brand or patent law. Any form of further use shall require the express consent from the relevant owner of the rights.

### **Important notes**

Utmost care was/is given in the preparation of the documentation at hand consisting of a user's manual, operating manual and any other document type and accompanying texts. However, errors cannot be ruled out. Therefore, we cannot assume any guarantee or legal responsibility for erroneous information or liability of any kind. You are hereby made aware that descriptions found in the user's manual, the accompanying texts and the documentation neither represent a guarantee nor any indication on proper use as stipulated in the agreement or a promised attribute. It cannot be ruled out that the user's manual, the accompanying texts and the documentation do not completely match the described attributes, standards or any other data for the delivered product. A warranty or guarantee with respect to the correctness or accuracy of the information is not assumed.

We reserve the right to modify our products and the specifications for such as well as the corresponding documentation in the form of a user's manual, operating manual and/or any other document types and accompanying texts at any time and without notice without being required to notify of said modification. Changes shall be taken into account in future manuals and do not represent an obligation of any kind, in particular there shall be no right to have delivered documents revised. The manual delivered with the product shall apply.

Under no circumstances shall Hilscher Gesellschaft für Systemautomation mbH be liable for direct, indirect, ancillary or subsequent damage, or for any loss of income, which may arise after use of the information contained herein.

### **Liability disclaimer**

The hardware and/or software was created and tested by Hilscher Gesellschaft für Systemautomation mbH with utmost care and is made available as is. No warranty can be assumed for the performance or flawlessness of the hardware and/or software under all application conditions and scenarios and the work results achieved by the user when using the hardware and/or software. Liability for any damage that may have occurred as a result of using the hardware and/or software or the corresponding documents shall be limited to an event involving willful intent or a grossly negligent violation of a fundamental contractual obligation. However, the right to assert damages due to a violation of a fundamental contractual obligation shall be limited to contract-typical foreseeable damage.



It is hereby expressly agreed upon in particular that any use or utilization of the hardware and/or software in connection with

- Flight control systems in aviation and aerospace;
- Nuclear fission processes in nuclear power plants;
- Medical devices used for life support and
- Vehicle control systems used in passenger transport

shall be excluded. Use of the hardware and/or software in any of the following areas is strictly prohibited:

- For military purposes or in weaponry;
- For designing, engineering, maintaining or operating nuclear systems;
- In flight safety systems, aviation and flight telecommunications systems;
- In life-support systems;
- In systems in which any malfunction in the hardware and/or software may result in physical injuries or fatalities.

You are hereby made aware that the hardware and/or software was not created for use in hazardous environments, which require fail-safe control mechanisms. Use of the hardware and/or software in this kind of environment shall be at your own risk; any liability for damage or loss due to impermissible use shall be excluded.

### **Warranty**

Hilscher Gesellschaft für Systemautomation mbH hereby guarantees that the software shall run without errors in accordance with the requirements listed in the specifications and that there were no defects on the date of acceptance. The warranty period shall be 12 months commencing as of the date of acceptance or purchase (with express declaration or implied, by customer's conclusive behavior, e.g. putting into operation permanently).

The warranty obligation for equipment (hardware) we produce is 36 months, calculated as of the date of delivery ex works. The aforementioned provisions shall not apply if longer warranty periods are mandatory by law pursuant to Section 438 (1.2) BGB, Section 479 (1) BGB and Section 634a (1) BGB [Bürgerliches Gesetzbuch; German Civil Code] If, despite of all due care taken, the delivered product should have a defect, which already existed at the time of the transfer of risk, it shall be at our discretion to either repair the product or to deliver a replacement product, subject to timely notification of defect.

The warranty obligation shall not apply if the notification of defect is not asserted promptly, if the purchaser or third party has tampered with the products, if the defect is the result of natural wear, was caused by unfavorable operating conditions or is due to violations against our operating regulations or against rules of good electrical engineering practice, or if our request to return the defective object is not promptly complied with.

### **Costs of support, maintenance, customization and product care**

Please be advised that any subsequent improvement shall only be free of charge if a defect is found. Any form of technical support, maintenance and customization is not a warranty service, but instead shall be charged extra.

### **Additional guarantees**

Although the hardware and software was developed and tested in-depth with greatest care, Hilscher Gesellschaft für Systemautomation mbH shall not assume any guarantee for the suitability thereof for any purpose that was not confirmed in writing. No guarantee can be granted whereby the hardware and software satisfies your requirements, or the use of the hardware and/or software is uninterruptable or the hardware and/or software is fault-free.

It cannot be guaranteed that patents and/or ownership privileges have not been infringed upon or violated or that the products are free from third-party influence. No additional guarantees or promises shall be made as to whether the product is market current, free from deficiency in title, or can be integrated or is usable for specific purposes, unless such guarantees or promises are required under existing law and cannot be restricted.

### **Confidentiality**

The customer hereby expressly acknowledges that this document contains trade secrets, information protected by copyright and other patent and ownership privileges as well as any related rights of Hilscher Gesellschaft für Systemautomation mbH. The customer agrees to treat as confidential all of the information made available to customer by Hilscher Gesellschaft für Systemautomation mbH and rights, which were disclosed by Hilscher Gesellschaft für Systemautomation mbH and that were made accessible as well as the terms and conditions of this agreement itself.

The parties hereby agree to one another that the information that each party receives from the other party respectively is and shall remain the intellectual property of said other party, unless provided for otherwise in a contractual agreement.

The customer must not allow any third party to become knowledgeable of this expertise and shall only provide knowledge thereof to authorized users as appropriate and necessary. Companies associated with the customer shall not be deemed third parties. The customer must obligate authorized users to confidentiality. The customer should only use the confidential information in connection with the performances specified in this agreement.

The customer must not use this confidential information to his own advantage or for his own purposes or rather to the advantage or for the purpose of a third party, nor must it be used for commercial purposes and this confidential information must only be used to the extent provided for in this agreement or otherwise to the extent as expressly authorized by the disclosing party in written form. The customer has the right, subject to the obligation to confidentiality, to disclose the terms and conditions of this agreement directly to his legal and financial consultants as would be required for the customer's normal business operation.

**Export provisions**

The delivered product (including technical data) is subject to the legal export and/or import laws as well as any associated regulations of various countries, especially such laws applicable in Germany and in the United States. The products / hardware / software must not be exported into such countries for which export is prohibited under US American export control laws and its supplementary provisions. You hereby agree to strictly follow the regulations and to yourself be responsible for observing them. You are hereby made aware that you may be required to obtain governmental approval to export, reexport or import the product.

## List of figures

Figure 1:	netFIELD OS architecture .....	9
Figure 2:	netFIELD OS container management .....	10
Figure 3:	netFIELD OS inter-container communication .....	11
Figure 4:	Positions on netFIELD Compact .....	15
Figure 5:	Dimensions in millimeters .....	16
Figure 6:	Adapter plug .....	17
Figure 7:	COMBICON plug .....	18
Figure 8:	Spring-loaded rail clamps .....	25
Figure 9:	Device mounted on DIN top hat rail.....	25
Figure 10:	Remove rail clamp screws.....	26
Figure 11:	Wall mounted.....	26
Figure 12:	Factory IP address settings of LAN interfaces .....	28
Figure 13:	Host name on device label (example) .....	30
Figure 14:	Setting IP address under Windows for direct LAN connection .....	31
Figure 15:	Login Device Manager.....	32
Figure 16:	Enter current password dialog.....	33
Figure 17:	Enter new password dialog .....	33
Figure 18:	Re-Authentication dialog .....	34
Figure 19:	System time value .....	35
Figure 20:	Change System Time dialog .....	35
Figure 21:	“Basic” onboarding screen in Local Device Manager .....	38
Figure 22:	Research Hardware ID .....	40
Figure 23:	Add device mask in netFIELD Portal.....	41
Figure 24:	Activation Code in portal.....	42
Figure 25:	Advanced Onboarding tab in device.....	43
Figure 26:	Example of an API Key permitting to onboard devices .....	45
Figure 27:	Copy key to clipboard .....	46
Figure 28:	Overview Local Device Manager.....	47
Figure 29:	System page in Local Device Manager .....	49
Figure 30:	Change host name dialog.....	50
Figure 31:	Networking page.....	52
Figure 32:	Details of LAN interface (eth0) .....	53
Figure 33:	IPv4 Settings .....	54
Figure 34:	Manual IPv4 Settings.....	55
Figure 35:	Open Firewall configuration page.....	58
Figure 36:	Elements on Firewall configuration page.....	59
Figure 37:	Add Zone dialog .....	60
Figure 38:	Add services .....	63
Figure 39:	Add custom services dialog.....	64
Figure 40:	Add forward port dialog .....	65

Figure 41:	Network Proxy configuration.....	66
Figure 42:	Proxy Settings dialog window.....	67
Figure 43:	Using one Proxy server for all protocols.....	68
Figure 44:	Separate HTTP/HTTPS/FTP configuration .....	69
Figure 45:	Restart dialog after changing proxy server configuration .....	70
Figure 46:	Synchronize proxy settings with netFIELD Portal.....	71
Figure 47:	Connectivity Check tab.....	73
Figure 48:	Basic Onboarding page .....	75
Figure 49:	Offboarding “Basic” .....	76
Figure 50:	Offboarding “Advanced” .....	77
Figure 51:	Web Server Settings tab.....	78
Figure 52:	Default MQTT Settings .....	79
Figure 53:	Docker Network Settings .....	81
Figure 54:	Default docker network configuration .....	84
Figure 55:	Remote Access tab .....	86
Figure 56:	Login tab.....	87
Figure 57:	Notification on Sign In dialog .....	87
Figure 58:	Standard Docker.....	88
Figure 59:	Expand concise container details .....	89
Figure 60:	Container parameters with terminal window.....	90
Figure 61:	Image Search dialog of Standard Docker.....	91
Figure 62:	Run Image dialog .....	92
Figure 63:	Expand image details .....	93
Figure 64:	Image details .....	94
Figure 65:	IOT Edge Docker.....	95
Figure 66:	Container details expanded.....	97
Figure 67:	Container parameters.....	98
Figure 68:	IoT image expanded.....	99
Figure 69:	Details of netFIELD Proxy image .....	100
Figure 70:	Accounts.....	101
Figure 71:	Create new account.....	101
Figure 72:	Edit account.....	102
Figure 73:	Web Server Certificate page .....	104
Figure 74:	Terminal.....	105
Figure 75:	OS update page .....	107
Figure 76:	Selected OS update image.....	108
Figure 77:	Upload finished message .....	109
Figure 78:	OS update “Disconnected” message.....	109
Figure 79:	Logs.....	111
Figure 80:	Services page.....	112
Figure 81:	Service details and settings page.....	113

---

Figure 82:	Service types: Timers .....	114
Figure 83:	Create timer dialog .....	114
Figure 84:	Formatting USB stick .....	116
Figure 85:	Format USB STICK dialog window.....	117
Figure 86:	Formatted USB stick.....	117
Figure 87:	Recovery image on USB stick .....	118
Figure 88:	Detach device from mounting bracket.....	124
Figure 89:	Open device housing.....	125
Figure 90:	Position of battery in device.....	125

## List of tables

Table 1:	List of revisions .....	4
Table 2:	Terms and abbreviations .....	6
Table 3:	Positions on netFIELD Compact device .....	15
Table 4:	Supply voltage connector .....	17
Table 5:	Serial interface .....	18
Table 6:	Serial port settings of console .....	20
Table 7:	LEDs LAN interface .....	21
Table 8:	Setting EDG LED state in MQTT topic .....	22
Table 9:	Tasks for commissioning the netFIELD Compact for netFIELD Cloud usage .....	23
Table 10:	Tasks for commissioning the netFIELD Compact for Standard Docker usage .....	24
Table 11:	Available Firewall zones .....	60
Table 12:	Elements in Add Zone dialog .....	61
Table 13:	Columns/elements in Allowed Services table .....	62
Table 14:	Columns/elements in Forward Ports table .....	64
Table 15:	Control elements in main toolbar .....	65
Table 16:	Default MQTT Client Settings .....	80
Table 17:	Standard Docker Network Settings .....	82
Table 18:	Standard Docker Network Settings .....	83
Table 19:	Technical data netFIELD Compact .....	121

# Contacts

## HEADQUARTER

### Germany

Hilscher Gesellschaft für  
Systemautomation mbH  
Rheinstraße 15  
65795 Hattersheim  
Phone: +49 (0) 6190 9907-0  
Fax: +49 (0) 6190 9907-50  
E-mail: [info@hilscher.com](mailto:info@hilscher.com)

### Support

Phone: +49 (0) 6190 9907-990  
E-mail: [hotline@hilscher.com](mailto:hotline@hilscher.com)

## SUBSIDIARIES

### China

Hilscher Systemautomation (Shanghai) Co. Ltd.  
200010 Shanghai  
Phone: +86 (0) 21-6355-5161  
E-mail: [info@hilscher.cn](mailto:info@hilscher.cn)

### Support

Phone: +86 (0) 21-6355-5161  
E-mail: [cn.support@hilscher.com](mailto:cn.support@hilscher.com)

### France

Hilscher France S.a.r.l.  
69800 Saint Priest  
Phone: +33 (0) 4 72 37 98 40  
E-mail: [info@hilscher.fr](mailto:info@hilscher.fr)

### Support

Phone: +33 (0) 4 72 37 98 40  
E-mail: [fr.support@hilscher.com](mailto:fr.support@hilscher.com)

### India

Hilscher India Pvt. Ltd.  
Pune, Delhi, Mumbai, Bangalore  
Phone: +91 8888 750 777  
E-mail: [info@hilscher.in](mailto:info@hilscher.in)

### Support

Phone: +91 8108884011  
E-mail: [info@hilscher.in](mailto:info@hilscher.in)

### Italy

Hilscher Italia S.r.l.  
20090 Vimodrone (MI)  
Phone: +39 02 25007068  
E-mail: [info@hilscher.it](mailto:info@hilscher.it)

### Support

Phone: +39 02 25007068  
E-mail: [it.support@hilscher.com](mailto:it.support@hilscher.com)

### Japan

Hilscher Japan KK  
Tokyo, 160-0022  
Phone: +81 (0) 3-5362-0521  
E-mail: [info@hilscher.jp](mailto:info@hilscher.jp)

### Support

Phone: +81 (0) 3-5362-0521  
E-mail: [jp.support@hilscher.com](mailto:jp.support@hilscher.com)

### Republic of Korea

Hilscher Korea Inc.  
13494, Seongnam, Gyeonggi  
Phone: +82 (0) 31-739-8361  
E-mail: [info@hilscher.kr](mailto:info@hilscher.kr)

### Support

Phone: +82 (0) 31-739-8363  
E-mail: [kr.support@hilscher.com](mailto:kr.support@hilscher.com)

### Austria

Hilscher Austria GmbH  
4020 Linz  
Phone: +43 732 931 675-0  
E-mail: [sales.at@hilscher.com](mailto:sales.at@hilscher.com)

### Support

Phone: +43 732 931 675-0  
E-mail: [at.support@hilscher.com](mailto:at.support@hilscher.com)

### Switzerland

Hilscher Swiss GmbH  
4500 Solothurn  
Phone: +41 (0) 32 623 6633  
E-mail: [info@hilscher.ch](mailto:info@hilscher.ch)

### Support

Phone: +41 (0) 32 623 6633  
E-mail: [support.swiss@hilscher.com](mailto:support.swiss@hilscher.com)

### USA

Hilscher North America, Inc.  
Lisle, IL 60532  
Phone: +1 630-505-5301  
E-mail: [info@hilscher.us](mailto:info@hilscher.us)

### Support

Phone: +1 630-505-5301  
E-mail: [us.support@hilscher.com](mailto:us.support@hilscher.com)