Nerve Blue 2.1.2 Documentation

Getting started

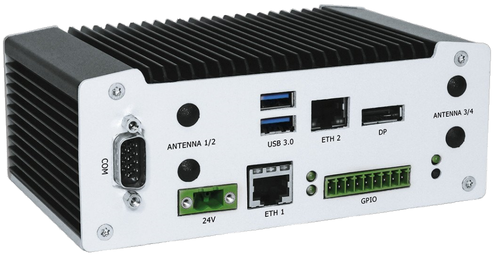
This documentation provides assistance for the usage of the Nerve Blue product.
It offers guides that can be found in the header of the page, catering to different levels of usage of the Nerve Blue product.

Note that this is the documentation of the latest Nerve Blue version. For previous versions of the product, select Previous Versions in the header.

Select a Nerve Device from the table below for a step-by-step installation:

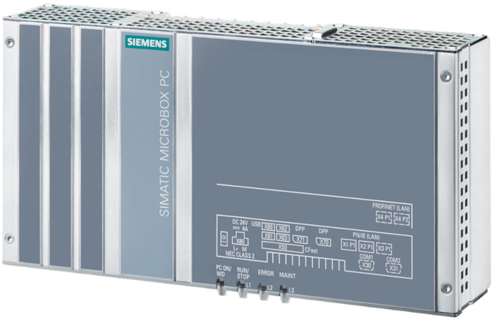
MFN 100

Kontron Kbox A-150-APL



Kontron Kbox A-250

Siemens SIMATIC IPC127E



Siemens SIMATIC IPC427E

Vecow SPC-5600-i5-8500

NOTE

Have the customer profile ready for the following chapters. It has been sent as part of the delivery. If the customer profile has not been part of the delivery, contact a sales representative or TTTech Industrial customer support at support@tttech-industrial.com.

Tutorial: Machine efficiency insight

Nerve is a platform for applications running on machines and accessing machine data. This tutorial shows how to create an application to gain insight into production and machine efficiency. It shows how to use Nerve to measure and visualize overall equipment effectiveness (OEE) of a virtual CNC mill. Going through this demo will take about 30 minutes and cover how to deploy container workloads, remotely access nodes and configure a Node-RED system to display an OEE and quality control dashboard.

Items needed for this tutorial are:

- a workstation or PC
- login credentials to the Nerve Management System and the filesharing server
- the Nerve Connection Manager app
- a flow file for Node-RED

All links, login credentials and the instructions to download files will be sent via emails. In case files or login credentials are missing, contact trynerve@tttech-industrial.com.

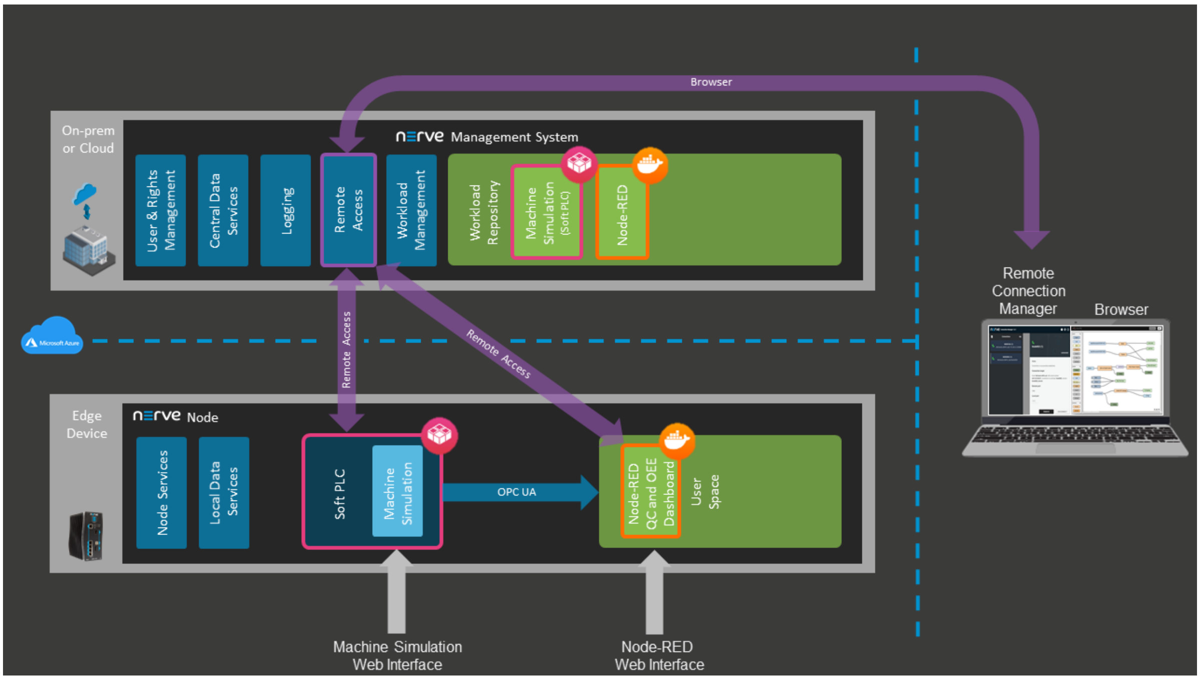
The tutorial environment is set up in the evaluation laboratory at TTTech Industrial in Vienna. The Nerve Device is an MFN 100 running an Intel Atom quad core CPU.

NOTE

In case the evaluation system is used, be aware that other users may access the same evaluation system, meaning that other nodes might be registered in the Management System. Information and workloads uploaded into the system may be visible to other users.

Tutorial architecture

In the course of this tutorial a simulation of a CNC machine (machine simulation) is installed on a Nerve Device. This machine simulation creates data which is read by the Node-RED application and visualized on a server running on the Nerve Device. The Node-RED application transforms the data received into an OEE and quality dashboard. The machine simulation, Node-RED and all dashboards are accessed on the Nerve Device directly, using the remote access feature of the Nerve cloud management platform called the Nerve Management System. In Nerve, applications managed by customers are called workloads.



Machine simulation

The digital twin simulates a machine which continuously creates parts and is subject to wear. The machine provides status information and Quality Control information through a built-in OPC UA server. The application is programmed in IEC 61131-3 and running in a soft PLC.

Node-RED

Node-RED is an open source programming tool for wiring together sensor inputs, APIs and online services. In this tutorial Node-RED reads data from the machine simulation through OPC UA, modifies it and displays it on an OEE and quality dashboard. The system is implemented based on predefined graphical modules, which are connected through drag and drop.

Refer to Node-RED documentation for more information on Node-RED. TTTech provides Node-RED as a supported third party application.

Initial setup

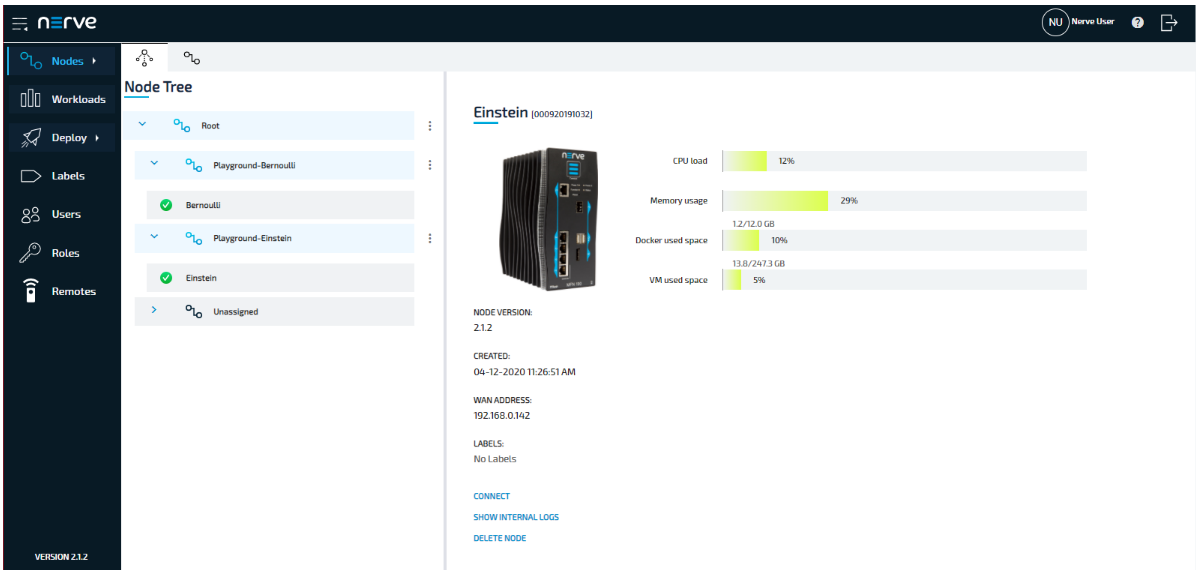
The Nerve Management System is set up so that the tutorial can be started straightaway. The necessary applications have already been configured and the remote connections are set up. The evaluation system does not limit experimentation to the extent of this tutorial. It is possible to create, deploy and test new workloads. However, the features of Local UI, Data Services and user management are not enabled in the standard evaluation system. Contact sales@tttech-industrial.com for information on how to obtain the full version.

To explore the Nerve system even further, refer to the user guide.

Viewing nodes in the node tree

The node tree is the first visible page of the Management System after logging in. It presents a means of organization for nodes that are connected to the Management System. Nodes are embedded into elements of the node tree.

Look for your node in the node tree. It is embedded in its own tree element. The node and the tree element are named after a scientist. The name has been sent in an email.



Einstein [000920191032]

CPU load 12%

Memory usage 29%

1.2/12.0 GB

Docker used space 10%

13.8/247.3 GB

VM used space 5%

NODE VERSION:
2.1.2

CREATED:
04-12-2020 11:26:51 AM

WAN ADDRESS:
192.168.0.142

LABELS:
No Labels

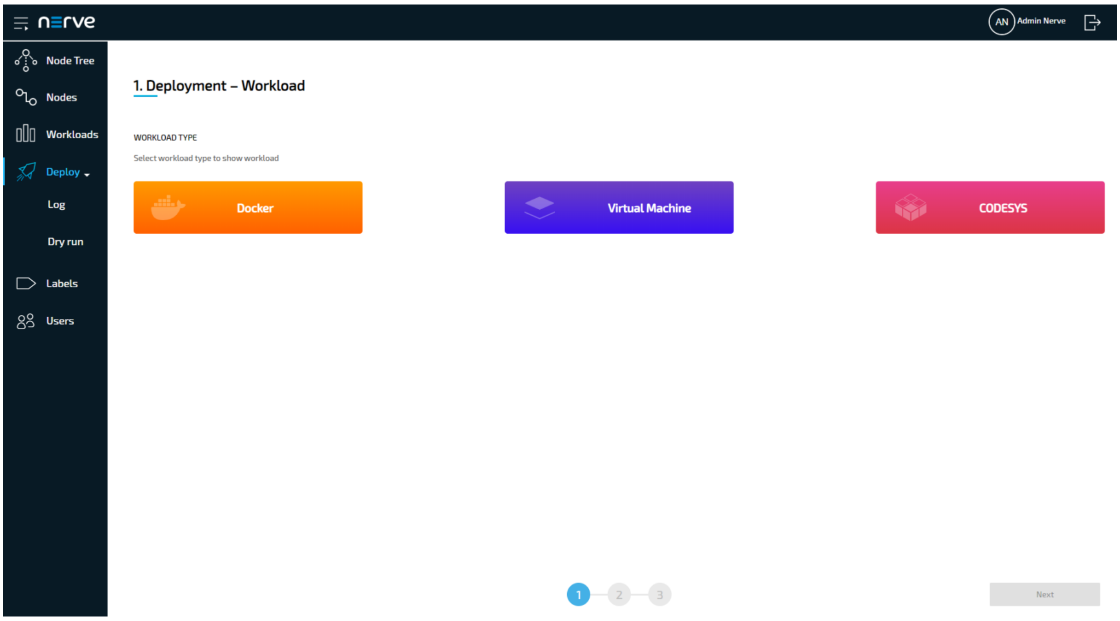
> **NOTE**
>
> Be aware that other users may access the same evaluation system. Make sure to use the designated node that was mentioned in the instruction emails.

Step 1: Deploying the machine simulation and Node-RED workloads

Deployment is the process of downloading workloads to Nerve Devices through the Nerve Management System. The instructions below show how to deploy the machine simulation as a CODESYS workload and Node-RED as a Docker workload.

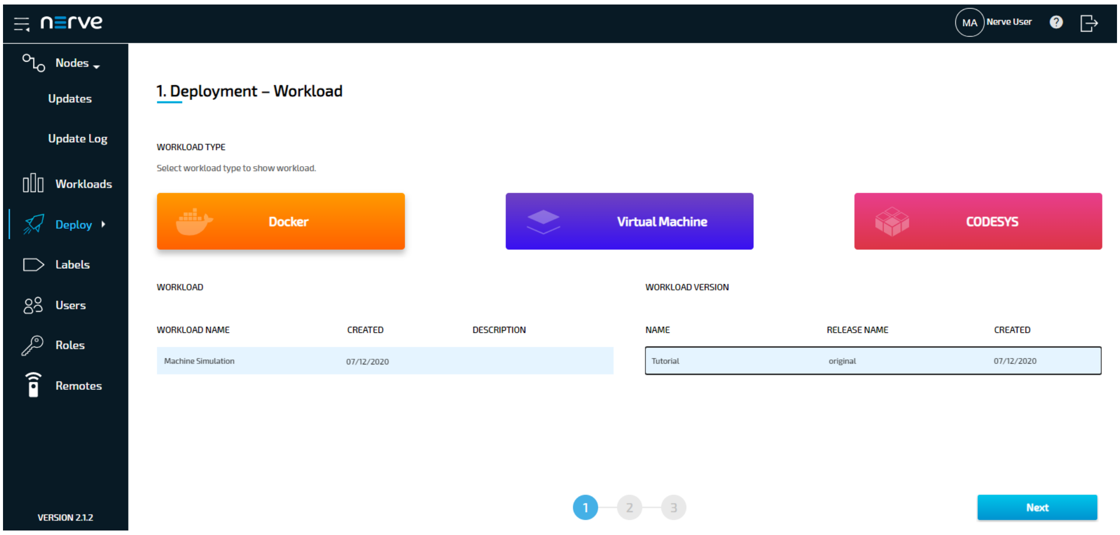
Machine simulation

1. Log in to the Management System.
2. Select **Deploy** in the navigation on the left.



3. Select the red CODESYS tab. A list of CODESYS workloads will appear below.

4. Select **Machine Simulation** in the list of workloads. A list of version of this workload will appear to the right.



WORKLOAD

WORKLOAD NAME	CREATED	DESCRIPTION
Machine Simulation	07/12/2020	

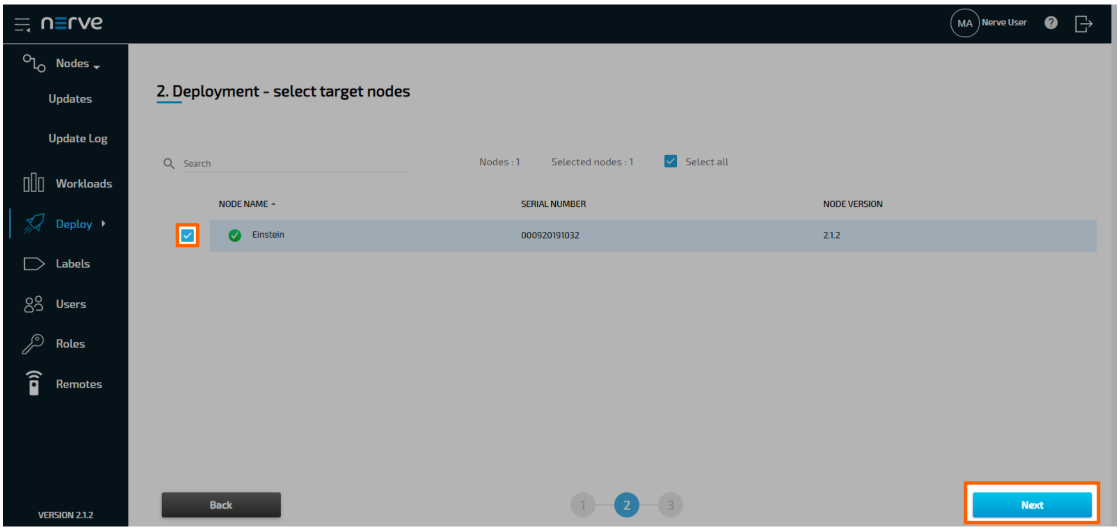
WORKLOAD VERSION

NAME	RELEASE NAME	CREATED
Tutorial	original	07/12/2020

5. Select the latest version of the workload on the right.

6. Click **Next** in the lower-right corner.
7. Tick the checkbox next to your node.

8. Select **Next** in the lower-right corner.



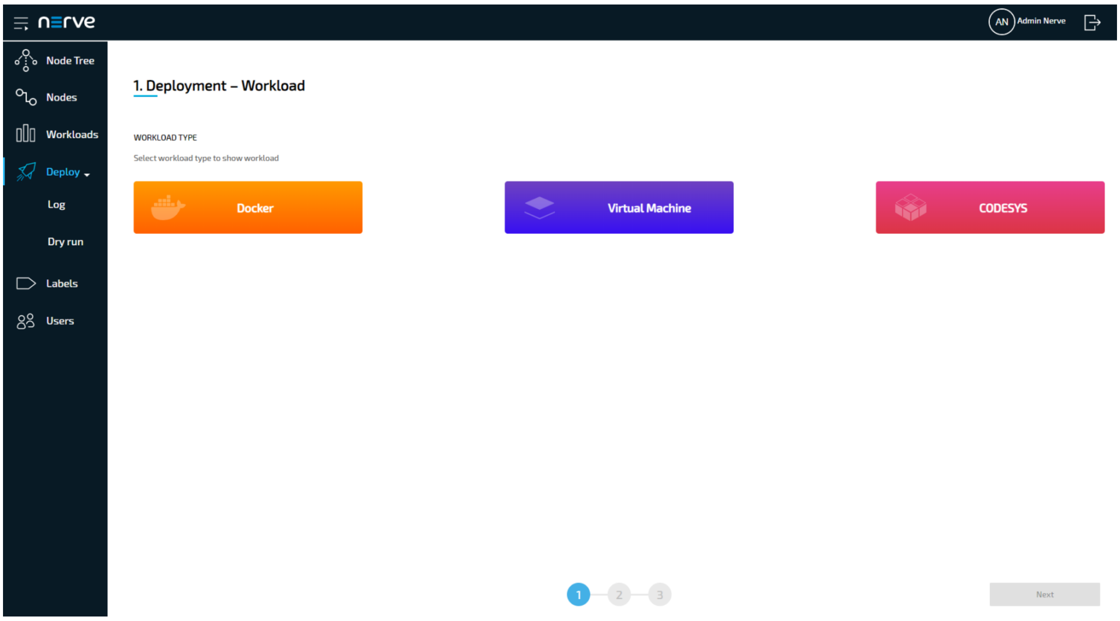
	NODE NAME	SERIAL NUMBER	NODE VERSION
☑	✓ Einstein	000920191032	2.1.2

9. Select **Deploy** to execute the deployment.
Optional: Enter a Deploy name above the Summary of the workload to make this deployment easy to identify. A timestamp is filled in automatically.

The deployment should now be visible at the top of the deployment log. Click the log entry of the deployment to see a more detailed view.

Node-RED

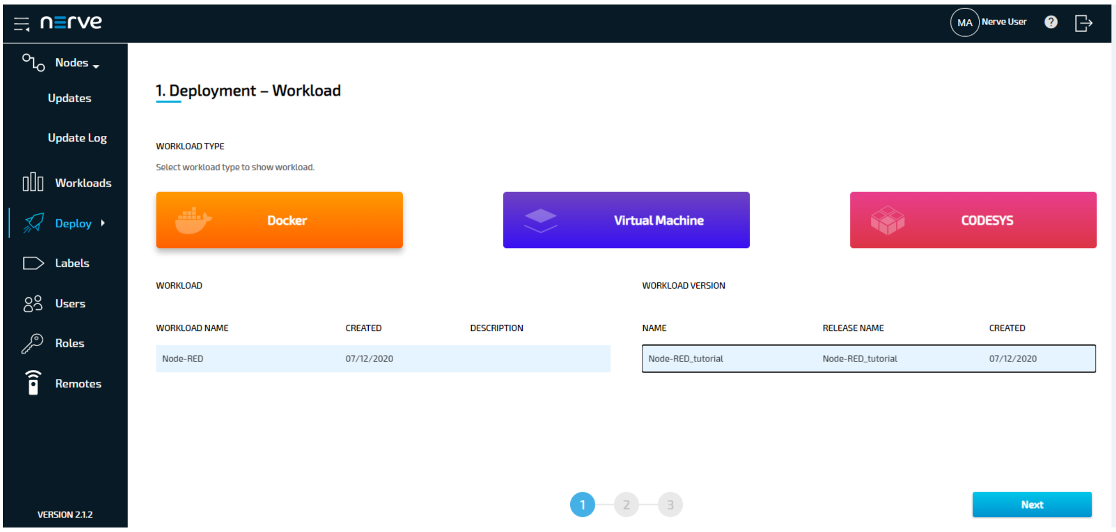
1. Select **Deploy** in the navigation on the left.



2. Select the orange Docker tab. A list of Docker workloads will appear below.

3. Select **Node-RED** in the list of workloads. A list of version of this workload will appear to the right.

4. Select the latest version of the workload.



1. Deployment – Workload

WORKLOAD TYPE

Select workload type to show workload.

WORKLOAD

WORKLOAD NAME	CREATED	DESCRIPTION
Node-RED	07/12/2020	

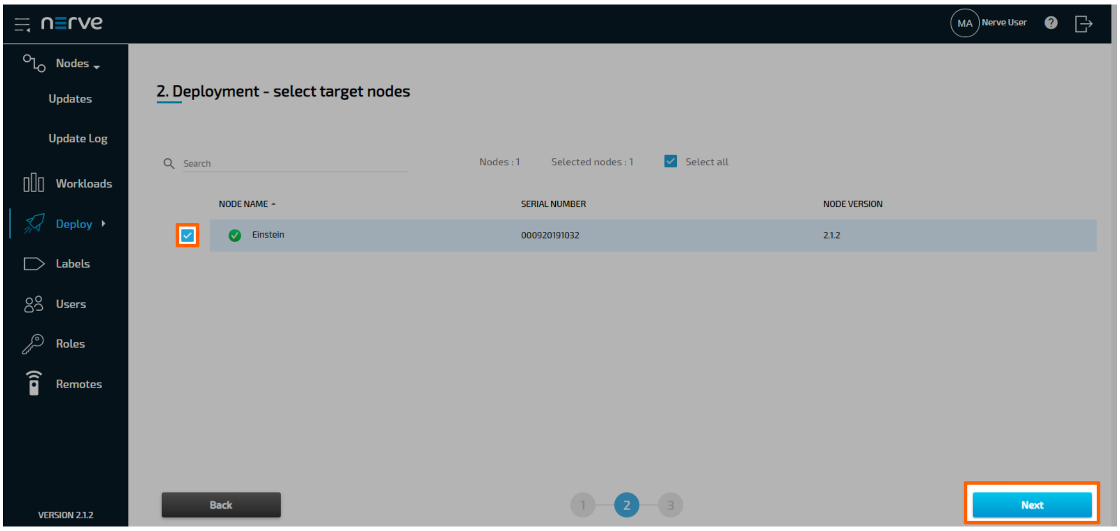
WORKLOAD VERSION

NAME	RELEASE NAME	CREATED
Node-RED_tutorial	Node-RED_tutorial	07/12/2020

5. Select **Next** in the lower-right corner.

6. Tick the checkbox next to your node.

7. Select **Next** in the lower-right corner.



2. Deployment - select target nodes

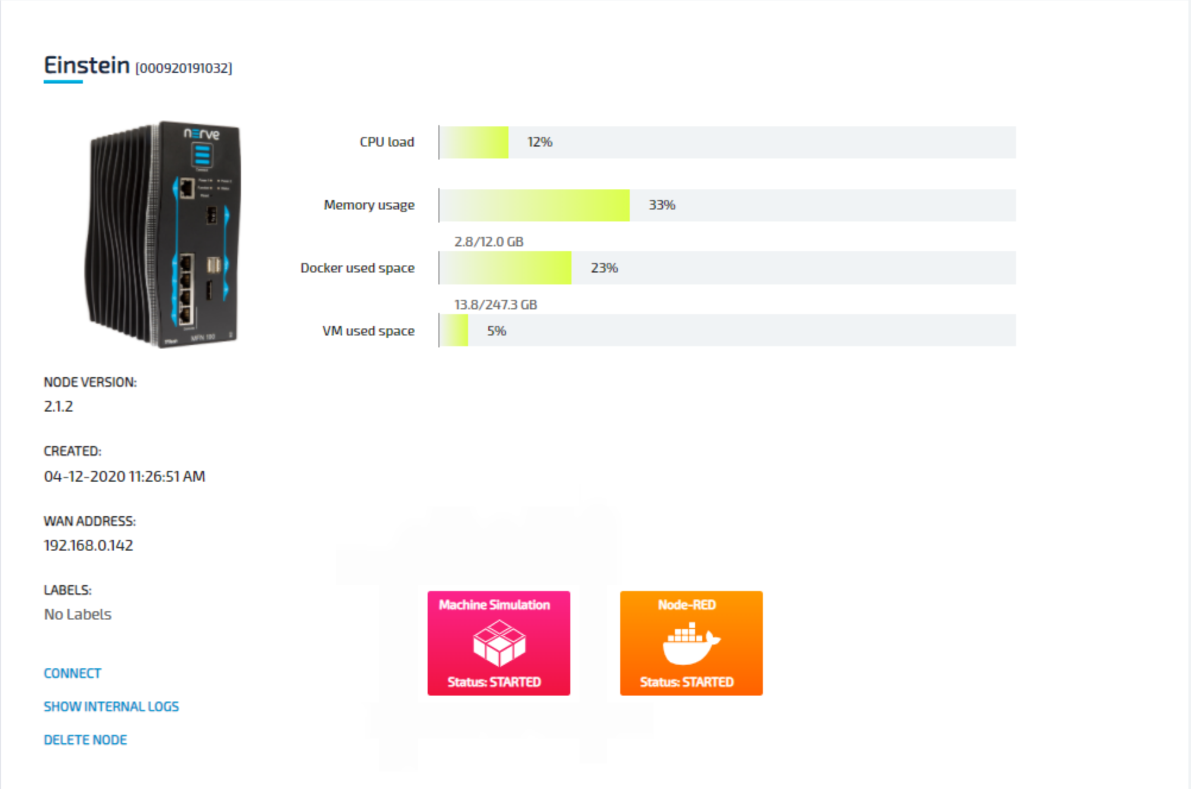
Nodes : 1 Selected nodes : 1 ☑ Select all

	NODE NAME	SERIAL NUMBER	NODE VERSION
☑	Einstein	000920191032	2.1.2

8. Select **Deploy** to execute the deployment.
Optional: Enter a Deploy name above the Summary of the workload to make this deployment easy to identify. A timestamp is filled in automatically.

The deployment should now be visible at the top of the deployment log. Click the log entry of the deployment to see a more detailed view.

To confirm if the workloads have been deployed successfully, select **Nodes** in the navigation on the left. Select your node in the node tree and confirm if two workload tiles are showing underneath the bar graph. The workloads should show the status **STARTED**.



Einstein [000920191032]

CPU load 12%

Memory usage 33%
2.8/12.0 GB

Docker used space 23%
13.8/247.3 GB

VM used space 5%

NODE VERSION:
2.1.2

CREATED:
04-12-2020 11:26:51 AM

WAN ADDRESS:
192.168.0.142

LABELS:
No Labels

CONNECT

SHOW INTERNAL LOGS

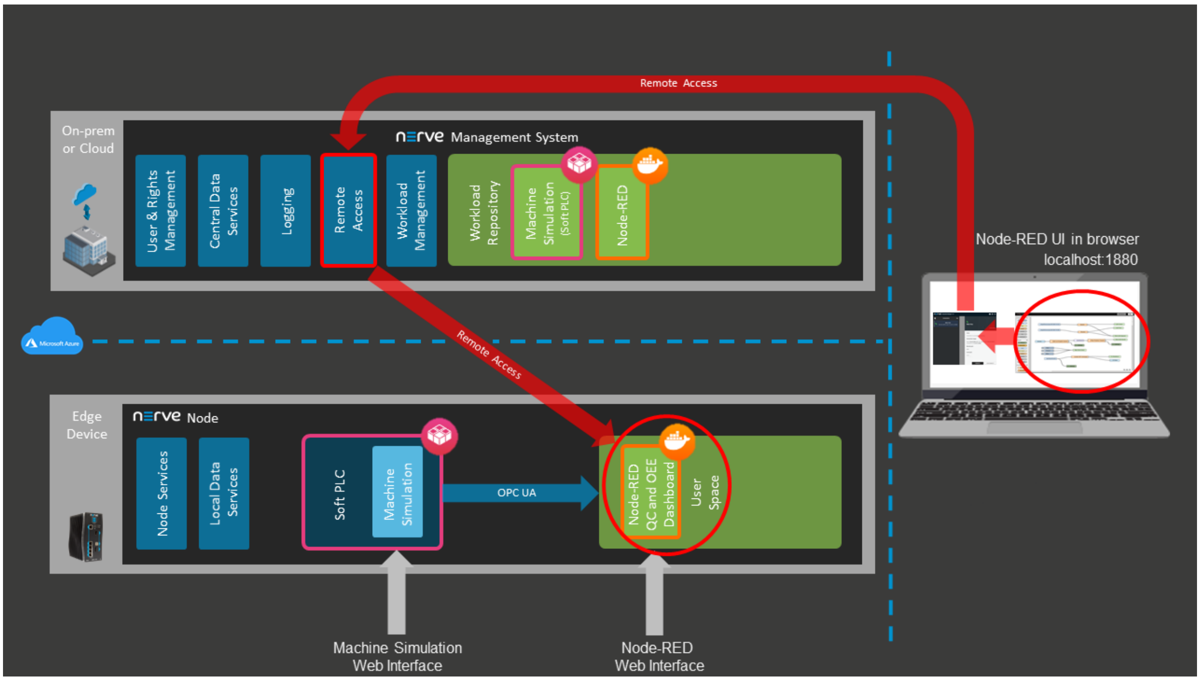
DELETE NODE

Machine Simulation
Status: STARTED

Node-RED
Status: STARTED

Step 2: Connecting to the workloads through remote access

To access the deployed workloads, it is required to create a connection between the computer used for this tutorial and the Management System, and from the Management System to the webservers of the workloads on the Nerve Device. This is done through the remote access feature, specifically through the use of remote tunnels. The remote tunnel connection has already been configured in the Management System and is ready for use. The image below illustrates the remote access feature, showing a connection to the deployed Node-RED workload on the Nerve Device.



Before continuing, install the Nerve Connection Manager that was part of the delivery, as it is required for using remote tunnels.

NOTE

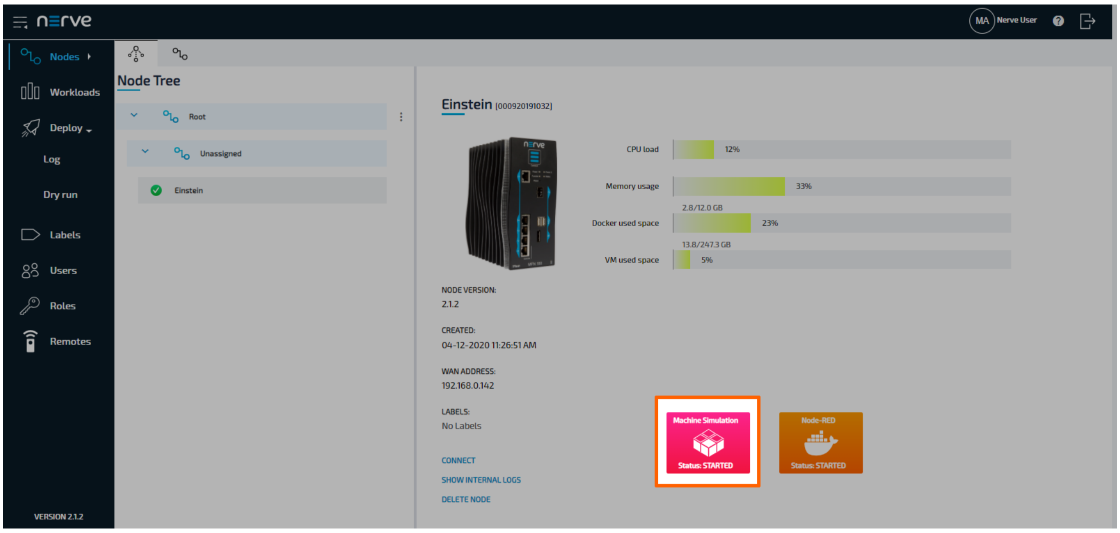
Local admin rights might be required to successfully install the Nerve Connection Manager.

Connecting to the machine simulation

The machine simulation is accessible at port 8080 on the Nerve Device through a web user interface. The preconfigured remote tunnel will use this port and port 8080 on the local computer to create a connection between the computer and the Nerve Device to access the machine simulation.

1. Select **Nodes** in the navigation on the left.
2. Select your node in the node tree.

3. Select the Machine Simulation CODESYS workload.



Einstein [000920191032]

CPU load 12%

Memory usage 33%
2.8/12.0 GB

Docker used space 23%
13.8/247.3 GB

VM used space 5%

NODE VERSION:
2.1.2

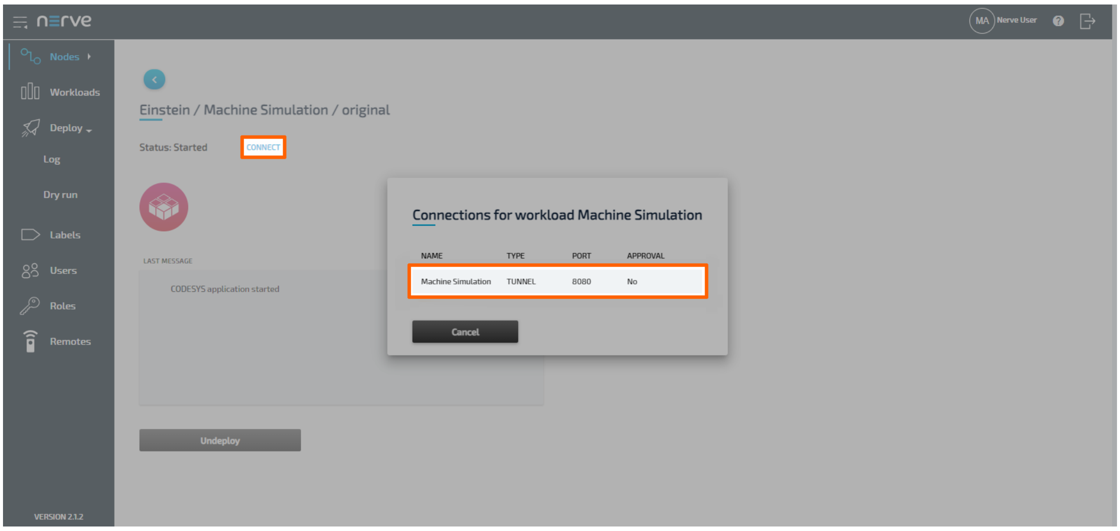
CREATED:
04-12-2020 11:26:51 AM

WAN ADDRESS:
192.168.0.142

LABELS:
No Labels

4. Select **CONNECT** next to the workload status. Available connections will appear in a window.

5. Select the **Machine Simulation** remote tunnel from the list.



Einstein / Machine Simulation / original

Status: Started

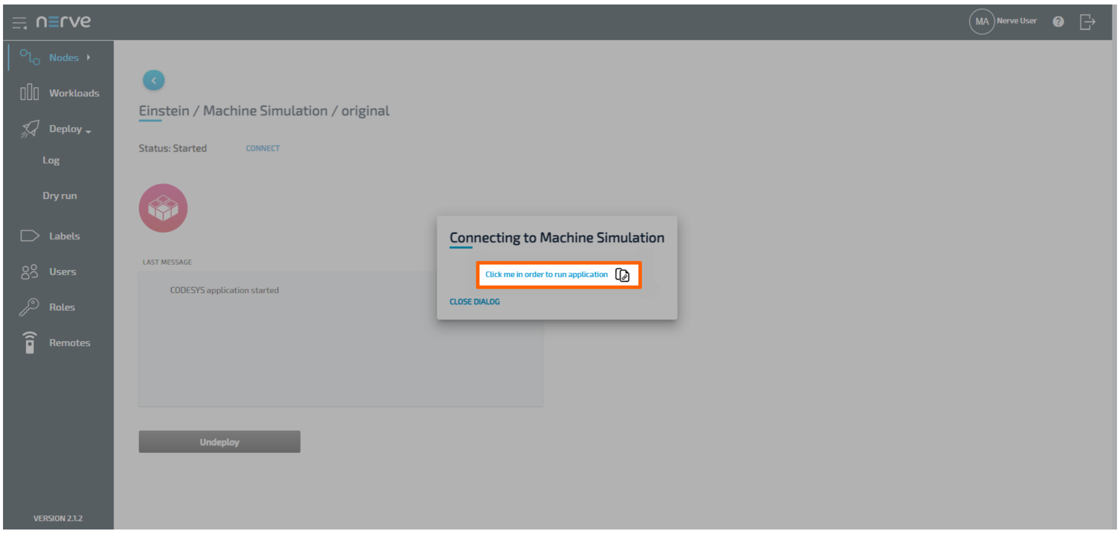
LAST MESSAGE

CODESYS application started

Connections for workload Machine Simulation

NAME	TYPE	PORT	APPROVAL
Machine Simulation	TUNNEL	8080	No

6. Select **Click me in order to run application** in the new window.



Einstein / Machine Simulation / original

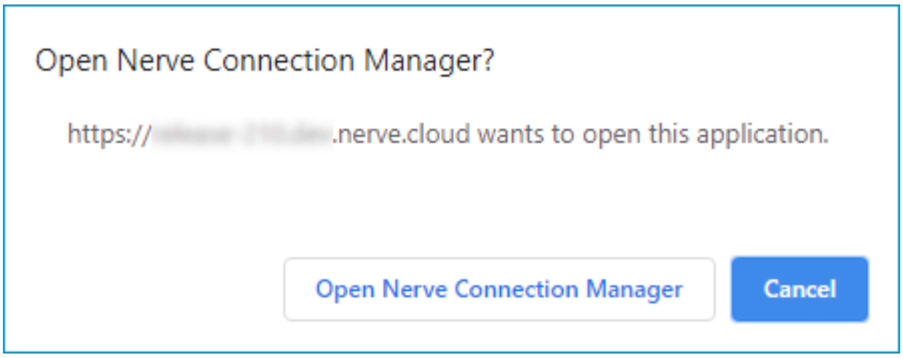
Status: Started

CODESYS application started

Connecting to Machine Simulation

Click me in order to run application

7. If the Nerve Connection Manager installed correctly, confirm the browser message that the Nerve Connection Manager shall be opened.
   Depending on the browser that is used, this message will differ. The Nerve Connection Manager will start automatically once the message is confirmed.



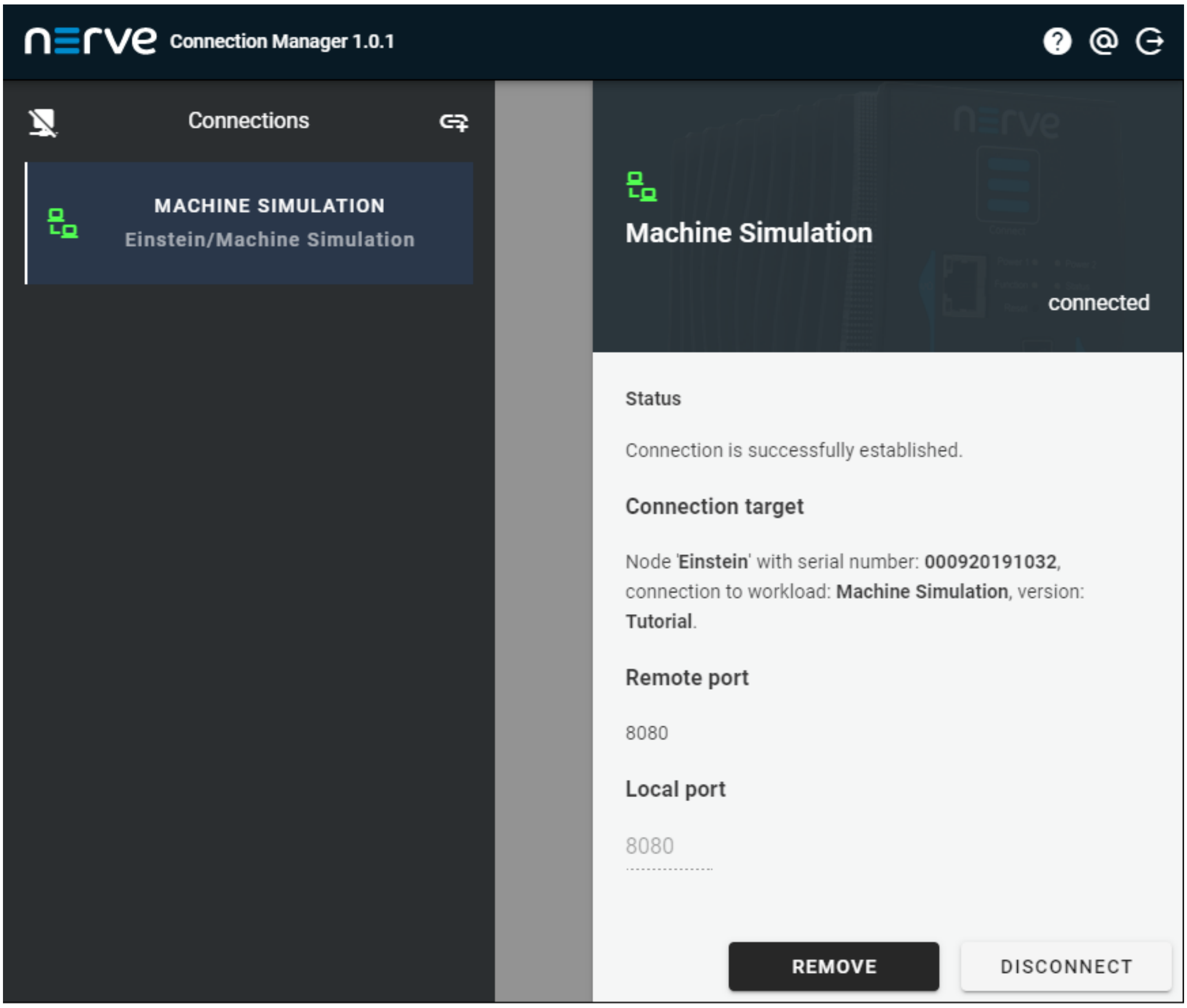
Open Nerve Connection Manager?

https://[redacted].nerve.cloud wants to open this application.

NOTE

If the Nerve Connection Manager does not start automatically, refer to Using a remote tunnel to a node or external device in the user guide.

The remote tunnel will be established once the Nerve Connection Manager starts and the remote connection will turn green in the Nerve Connection Manager. The remote tunnel is now ready to be used.



Status

Connection is successfully established.

Connection target

Node 'Einstein' with serial number: 000920191032, connection to workload: Machine Simulation, version: Tutorial.

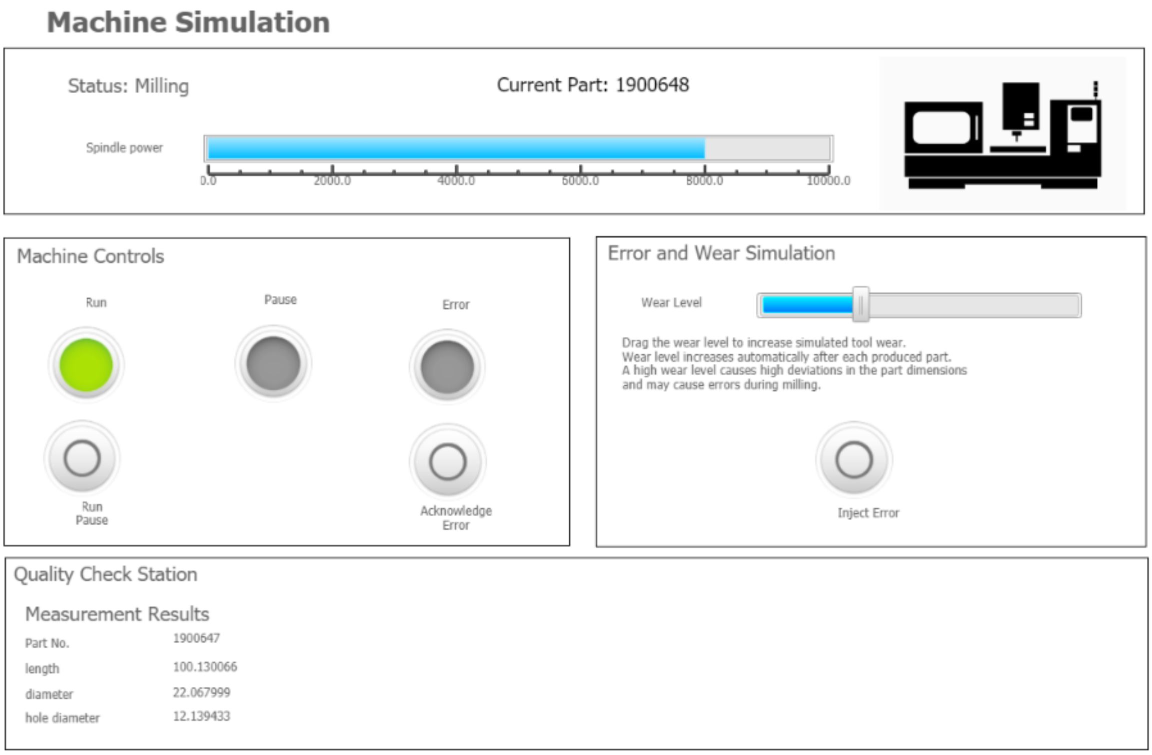
Remote port

8080

Local port

8080

Open a new browser tab and enter `localhost:8080` to open the machine simulation dashboard.



Machine Simulation

Status: Milling

Current Part: 1900648

Drag the wear level to increase simulated tool wear.
Wear level increases automatically after each produced part.
A high wear level causes high deviations in the part dimensions
and may cause errors during milling.

Quality Check Station

Measurement Results

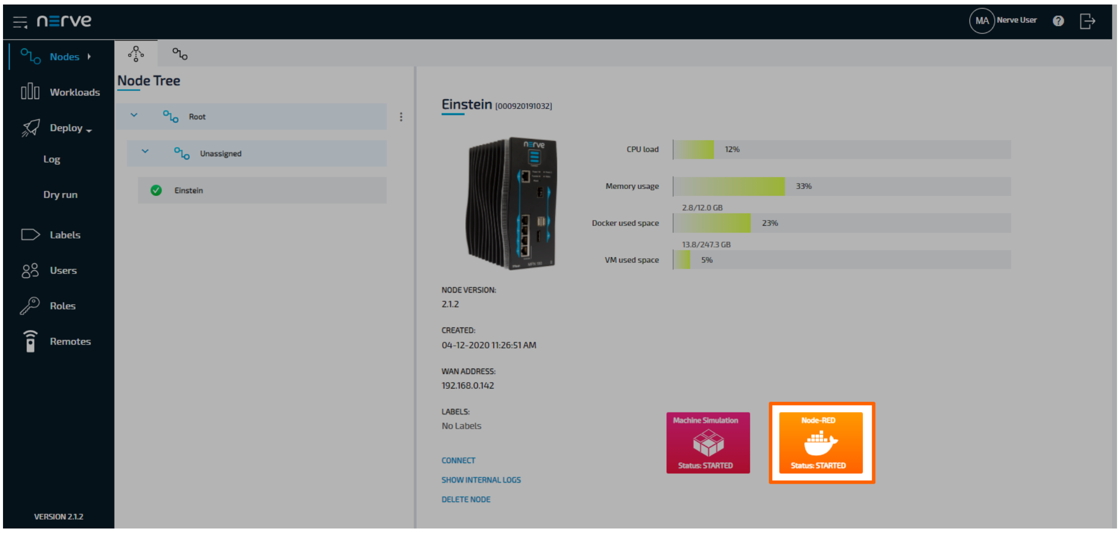
Part No.	1900647
length	100.130066
diameter	22.067999
hole diameter	12.139433

Connecting to Node-RED

Node-RED is accessible at port 1880 on the Nerve Device through a web user interface. The preconfigured remote tunnel will use this port and port 1880 on the local computer to create a connection between the computer and the Nerve Device to access the interface of Node-RED.

1. Select **Nodes** in the navigation on the left.
2. Select your node in the node tree.

3. Select the Node-RED Docker workload.



Einstein [000920191032]

CPU load 12%

Memory usage 33% 2.8/12.0 GB

Docker used space 23% 13.8/247.3 GB

VM used space 5%

NODE VERSION:
2.1.2

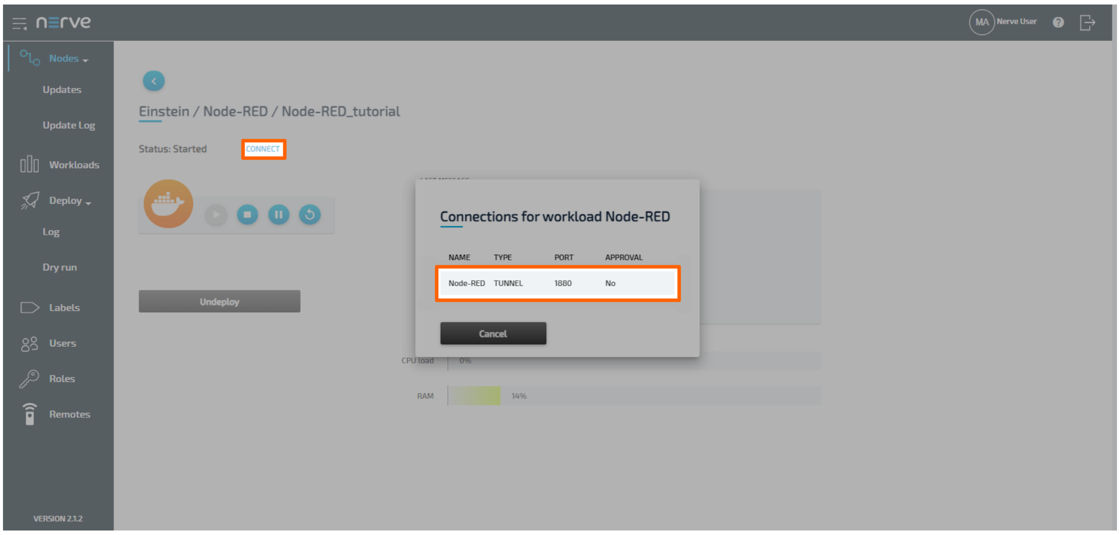
CREATED:
04-12-2020 11:26:51 AM

WAN ADDRESS:
192.168.0.142

LABELS:
No Labels

4. Select **CONNECT** next to the workload status. Available connections will appear in a window.

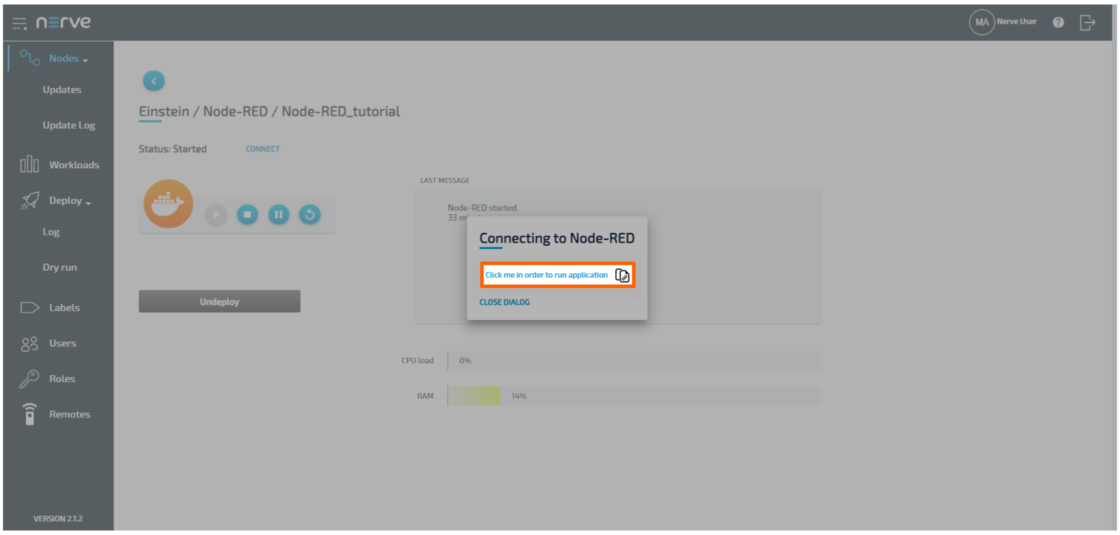
5. Select the **Node-RED** remote tunnel from the list.



Connections for workload Node-RED

NAME	TYPE	PORT	APPROVAL
Node-RED	TUNNEL	1880	No

6. Select **Click me in order to run application** in the new window.

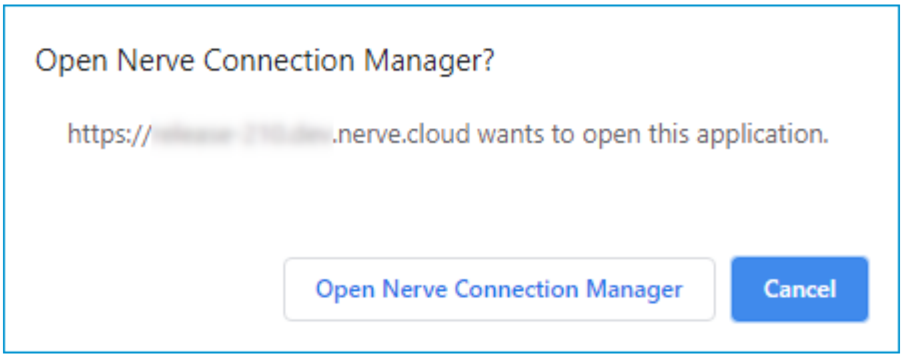


Node-RED started.
33 m

Connecting to Node-RED

Click me in order to run application

7. If the Nerve Connection Manager installed correctly, confirm the browser message that the Nerve Connection Manager shall be opened.
   Depending on the browser that is used, this message will differ. The Nerve Connection Manager will start automatically once the message is confirmed.



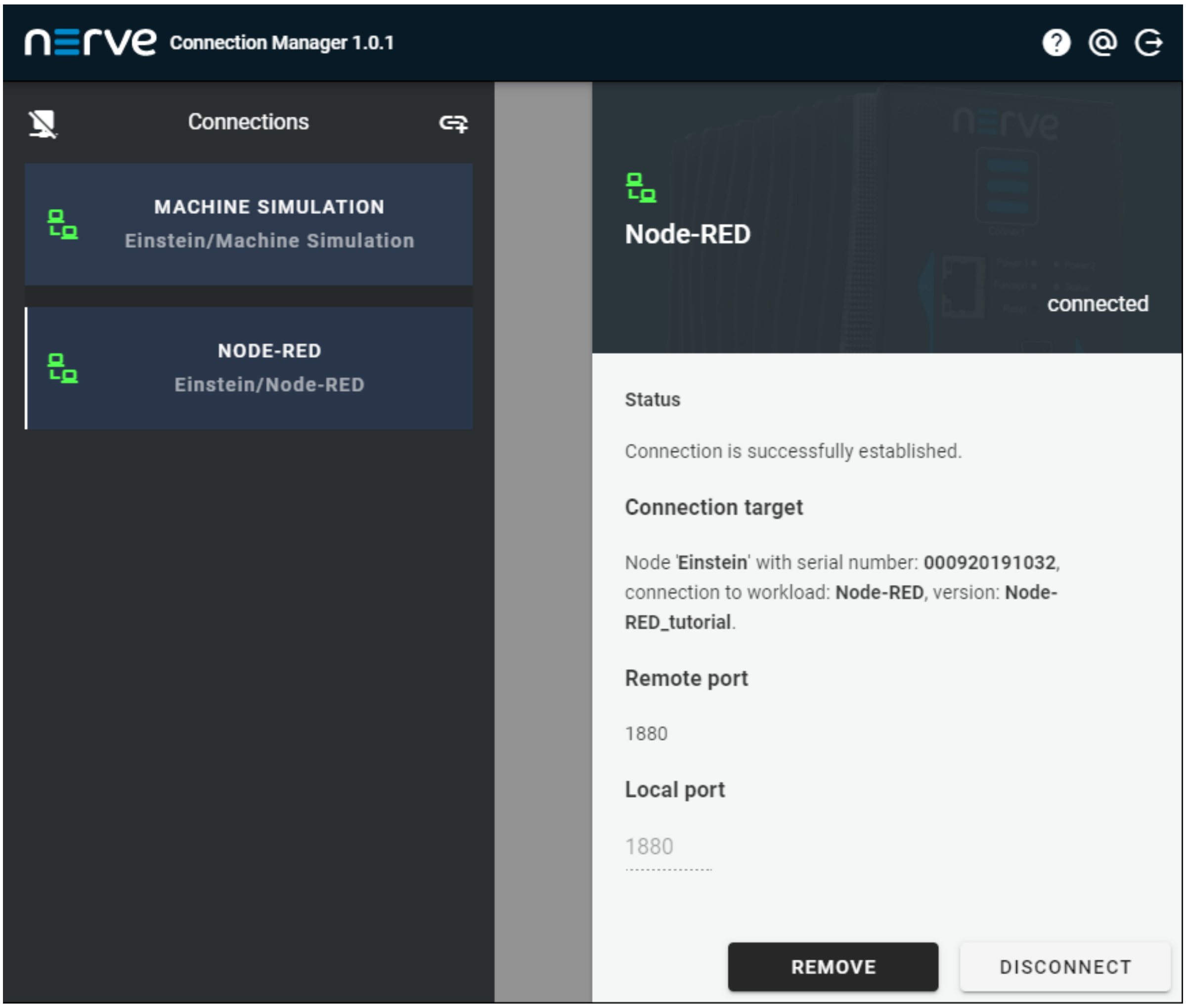
Open Nerve Connection Manager?

https://[redacted].nerve.cloud wants to open this application.

NOTE

If the Nerve Connection Manager does not start automatically, refer to Using a remote tunnel to a workload in the user guide.

The remote tunnel will be established once the Nerve Connection Manager starts and the remote connection will turn green in the Nerve Connection Manager. The remote tunnel is now ready to be used.



Status

Connection is successfully established.

Connection target

Node 'Einstein' with serial number: 000920191032, connection to workload: Node-RED, version: Node-RED_tutorial.

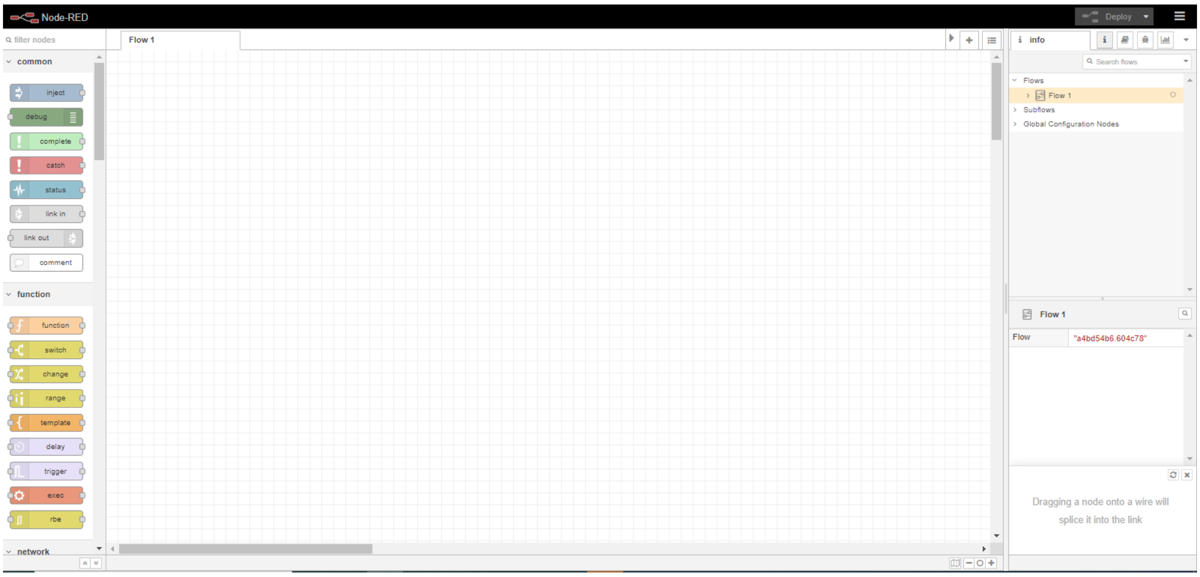
Remote port

1880

Local port

1880

Open a new browser tab and enter `localhost:1880` to open Node-RED.



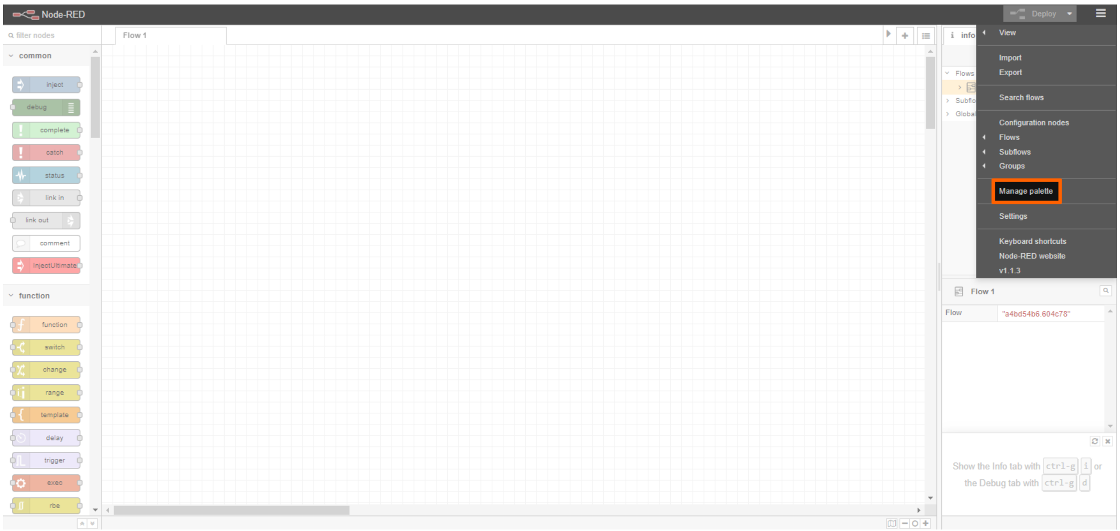
Dragging a node onto a wire will splice it into the link

Step 3: Setting up Node-RED

Node-RED is running on the node and the workstation is connected to it through a remote tunnel. Switch to the browser tab that has Node-RED opened at `localhost:1880`. In the instructions below the necessary modules are installed and the flow is imported.

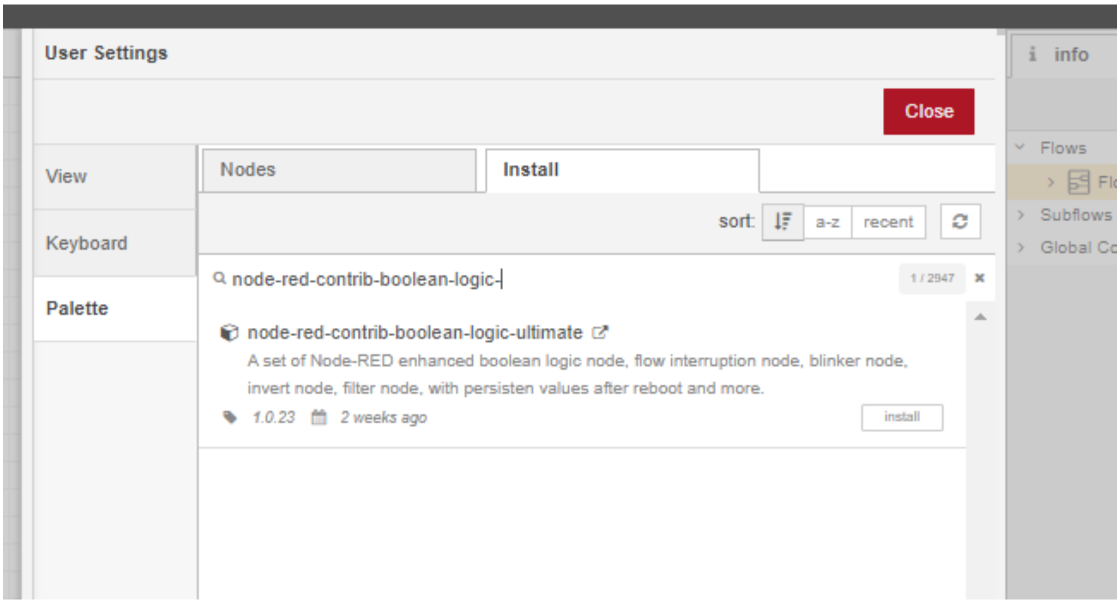
> **NOTE**
>
> Download the JSON file that has been part of the delivery and remember its download location.

1. Select the burger menu in the top right of the Node-RED application.
2. Select **Manage palette**.



Show the Info tab with ctrl-g i or the Debug tab with ctrl-g d

3. Select the **Install** tab.
4. Make sure **Palette** is selected on the left.
5. Enter `node-red-contrib-boolean-logic-ultimate` in the search bar.



node-red-contrib-boolean-logic-

node-red-contrib-boolean-logic-ultimate

A set of Node-RED enhanced boolean logic node, flow interruption node, blinker node, invert node, filter node, with persisten values after reboot and more.

1.0.23 2 weeks ago install

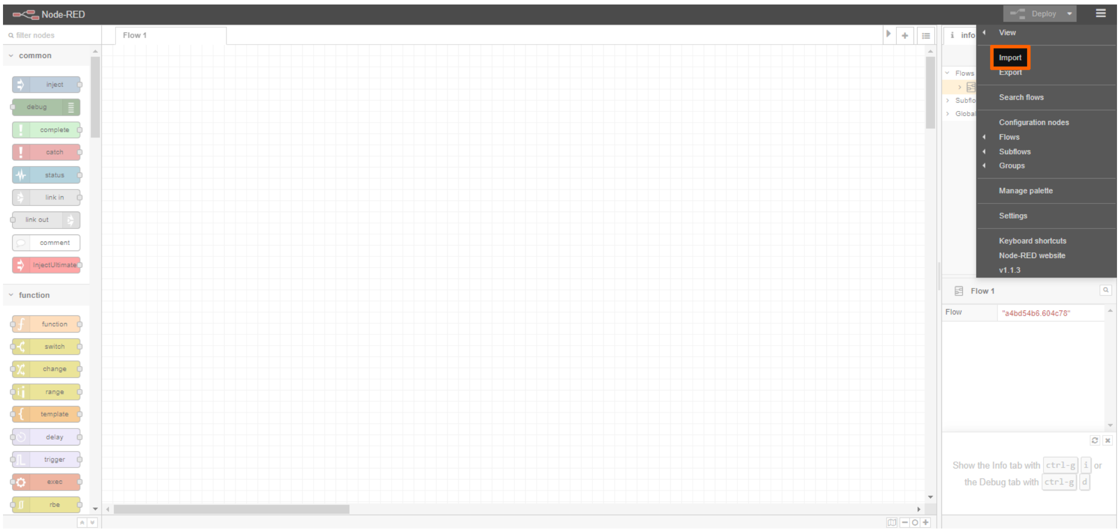
6. Select **Install** in the module that appeared.

> **NOTE**
>
> A pop-up will appear once the module is successfully installed.

7. Make sure **Palette** is selected on the left.
8. Enter `node-red-contrib-counter` in the search bar.
9. Select **Install** in the module that appeared.

> **NOTE**
>
> A pop-up will appear once the module is successfully installed.

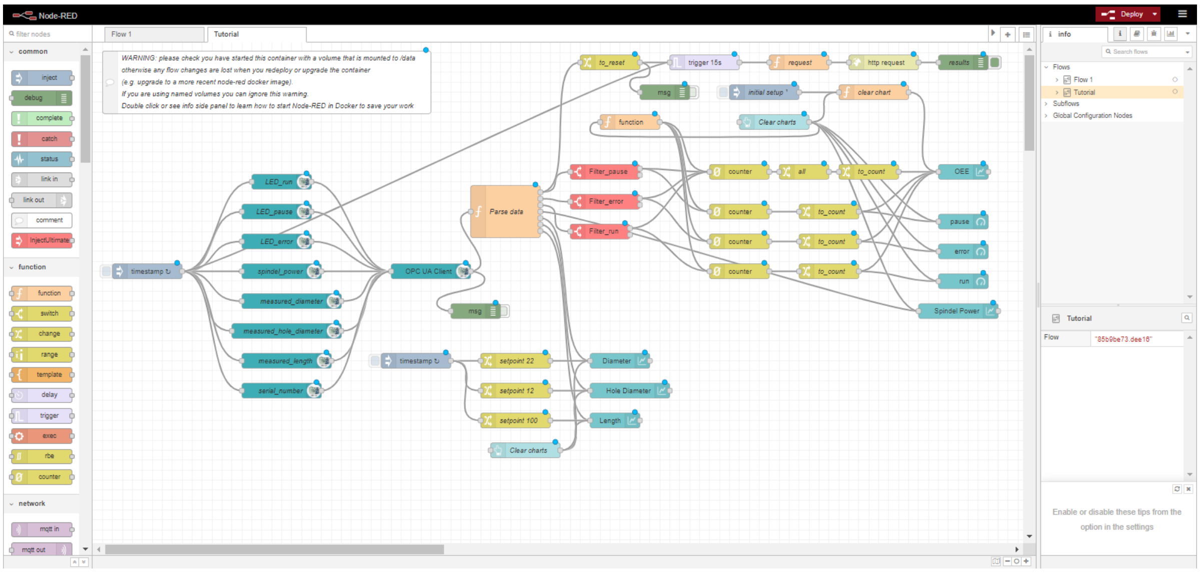
10. Select **Close** in the upper-right corner to close the **User Settings** window after all modules are installed.
11. Select the burger menu in the top-right corner.
12. Select **Import**.



13. Select **select a file to import**.

14. Look for the JSON file that has been downloaded before.
15. Select **Open**.
16. Select **Import** in the lower-right corner.

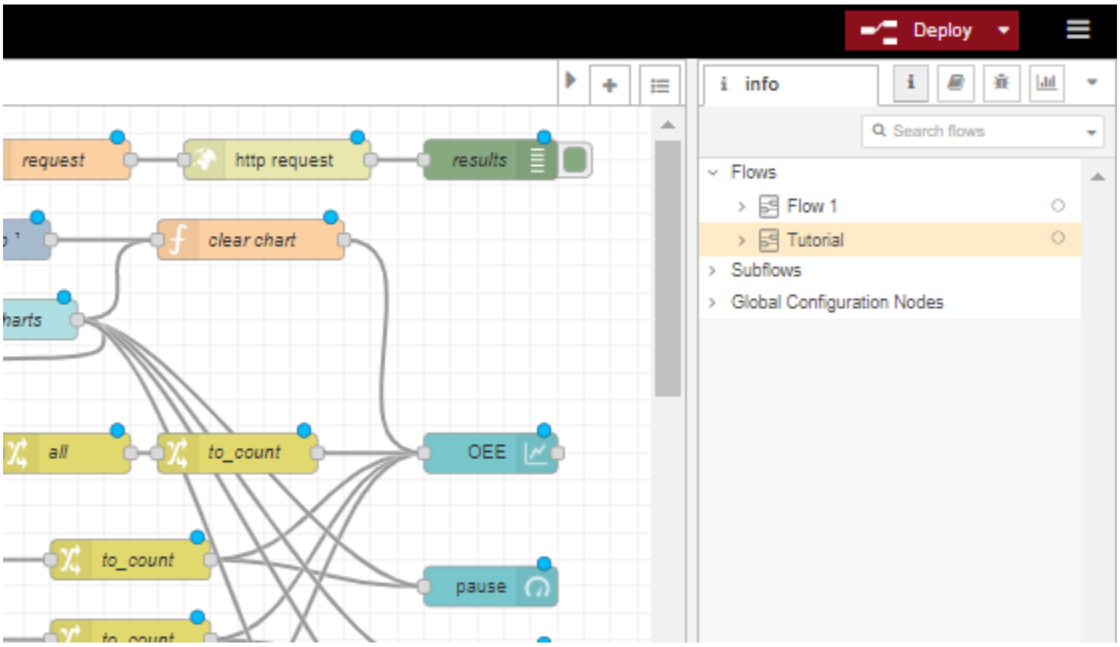
The flow will appear in a new tab inside of Node-RED labelled **Tutorial** in the top-left corner.



Step 4: Accessing the OEE and quality dashboard

After the Node-RED setup, data generated by the machine simulation can be viewed on the OEE and quality dashboard.

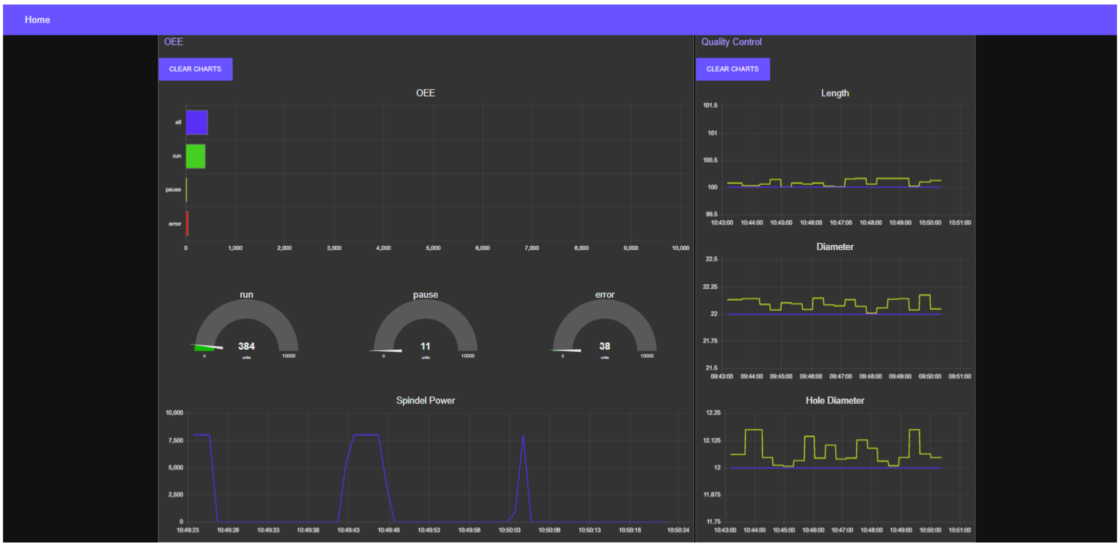
1. In the Node-RED browser tab select **Deploy** in the top-right corner to start the flow. Make sure that the **Tutorial** tab is active.



2. Open a new browser tab.

3. Enter `localhost:1880/ui` to open the OEE dashboard.

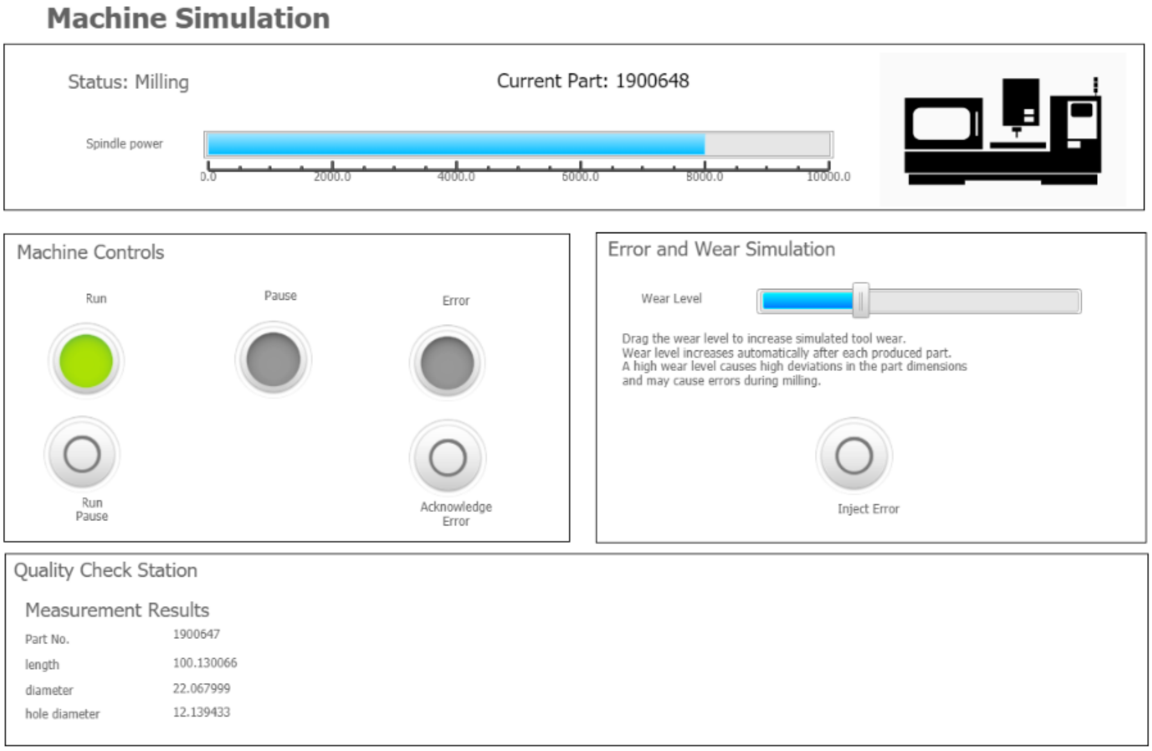
> **NOTE**
>
> It may take a moment for the dashboard to load.



Experimenting with the machine simulation

The machine simulation runs through a process of part production. The production of a part begins with the machine waiting for a part in order to load it. Once the part is loaded in, the machine starts milling and the spindle power goes up. After the machine is finished milling and the spindle power goes down again, the part is then unloaded and sent to a measuring station. The **Measurement Results** in the **Quality Check Station** are updated as soon as measuring is completed. The operation of the machine simulation and the measurements of the parts are impacted by the adjustable **Wear Level**.

The user interface of the machine simulation is split up into different parts.



Machine Simulation

Status: Milling

Current Part: 1900648

Drag the wear level to increase simulated tool wear.
Wear level increases automatically after each produced part.
A high wear level causes high deviations in the part dimensions
and may cause errors during milling.

Quality Check Station

Measurement Results

Part No.	1900647
length	100.130066
diameter	22.067999
hole diameter	12.139433

Item	Description
Status	This shows the current status of the machine simulation, the number of the current part and the spindle power when the machine is milling. The status of the machine cycles through the following way in regular operation: • **Waiting for Part** • **Loading** • **Milling** • **Unloading** • **Measuring** If the machine operation is stopped by an error or a pause, the status will display **Error** or **Pause** and the operation at which the pause or error occured in parentheses, e.g **Error (waiting for part)**.

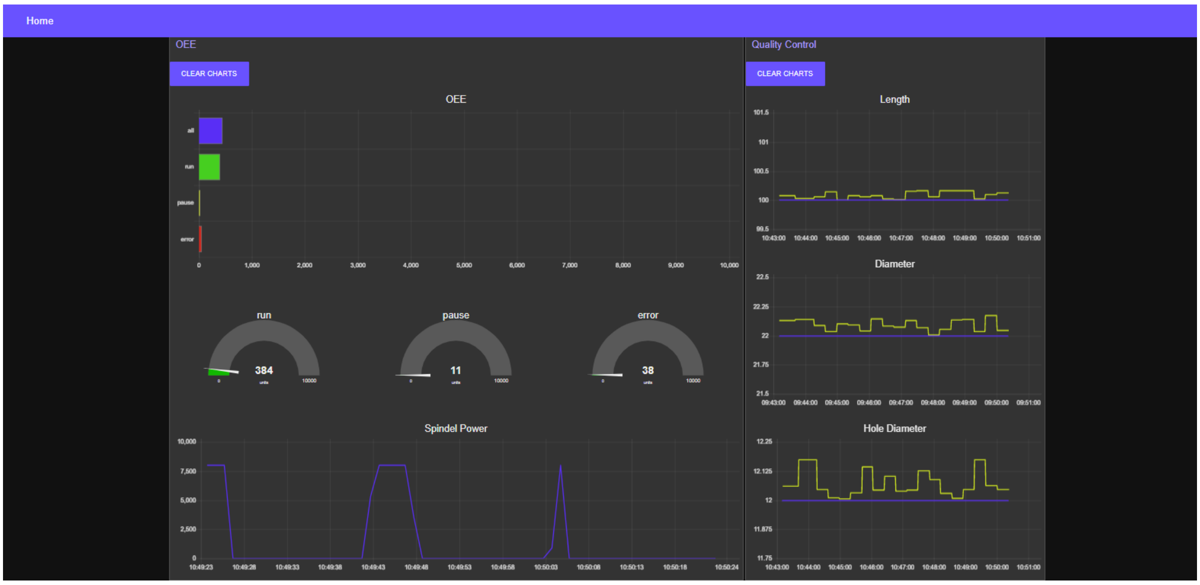
Item	Description
Machine Controls	The machine controls consist of three colored lights and two buttons. - **Run** This light is green if the machine is running as intended. - **Pause** This light is yellow if the machine has been paused by the user. - **Error** This light is red if the machine is in an error state, either caused by a high wear level or by injecting an error manually. - **Run/Pause** Use this button to pause and unpause the machine simulation. - **Acknowledge Error** If the machine is in an error state, this button must be pressed to acknowledge the error and have the machine resume its operation.
Error and Wear Simulation	The slider here emulates the wear of the system. It increases gradually when the machine is running. The higher the wear level, the higher the probability of an error occuring in the simulation. The slider can be adjusted to preference to simulate different levels of wear. Manually inject an error by pressing the **Inject Error** button.
Quality Check Station	After the completion of a part, the **Measurement Results** are updated, showing new values for the **Part No.**, **length**, **diameter** and **hole diameter** of a part. The higher the wear level, the higher are the deviations of the part measurements.

The data of the machine operation is amassed and displayed in an OEE dashboard by Node-RED.

Node-RED and the OEE dashboard

The data flow in Node-RED reads from an OPC UA Client and parses the data into the OEE statistics and quality check data. The dashboard consists of three parts:

- **OEE**
  This part of the dashboard shows the amount of the time the machine has spent in **run**, **pause** and **error** states in two different visualizations.
- **Spindle Power**
  This is a graph of how high spindle power is over time.
- **Quality Control**
  Length, **Diameter** and **Hole Diameter** are displayed in graphs, showing the measurements of the parts produced. Deviations caused by a high wear level can be analyzed here.



This concludes the tutorial for Nerve. For questions, requests and further evaluation, contact trynerve@tttech-industrial.com.

User Guide

User Guide

The user guide for Nerve Blue covers the features and configurations a user has with Nerve Blue using the two user interfaces of Nerve Blue: the Local UI and the Management System. It is supported by the device guide that contains device specific information required for working with Nerve Blue and the instructions in the user guide.

If there are any questions about the software installed and the features provided, contact a sales representative or TTTech Industrial customer support at support@tttech-industrial.com.

Recommended chapters after node registration

Once a node has been registered following the Getting started chapter, the following topics might be of interest:

- Management System overview
- Adding a user
- Adding a new role
- Provisioning a CODESYS workload
- Provisioning a Docker workload
- Provisioning a Virtual machine workload
- Deploying a workload
- First steps with CODESYS programming
- Nerve Data Services introduction
- Node internal networking

Hardware specifics

The user guide focuses on how to operate the Nerve Blue software. As such it will not contain any device specific information. Whenever device specific information is required, the user guide will link to the device guide.

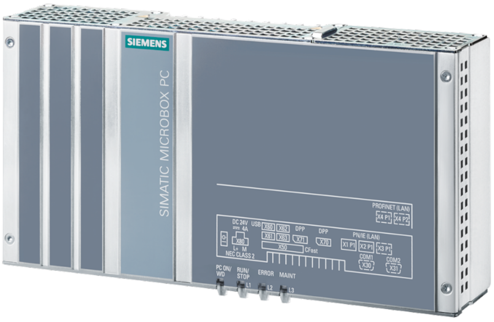
Refer to the chapter of the Nerve Device in the device guide in order to setup and install the hardware. and find out device specific information:

MFN 100

Kontron Kbox A-150-APL

Kontron Kbox A-250

Siemens SIMATIC IPC127E



Siemens SIMATIC IPC427E

Vecow SPC-5600-i5-8500

License activation

The license for using Nerve Blue has to be activated on the node before the product can be used. A ticket ID that is required for the activation of the license has been sent as part of the delivery, along with a link to a web depot. With this ticket ID and the web depot link, the product can be activated in two ways:

- **Online activation**
  The node must have access to the internet.
- **Offline activation**
  The node does not have access to the internet.

Note that offline activation requires a workstation with a connection to the internet.

> **NOTE**
>
> Make sure to connect a workstation to the physical port of the Nerve Device associated with host access and configure the network adapter of the workstation. The IP address has to be in the same range as the IP address of the host access interface with a `255.255.255.0` subnet mask. Refer to the device guide for more information per Nerve Device.

Online license activation

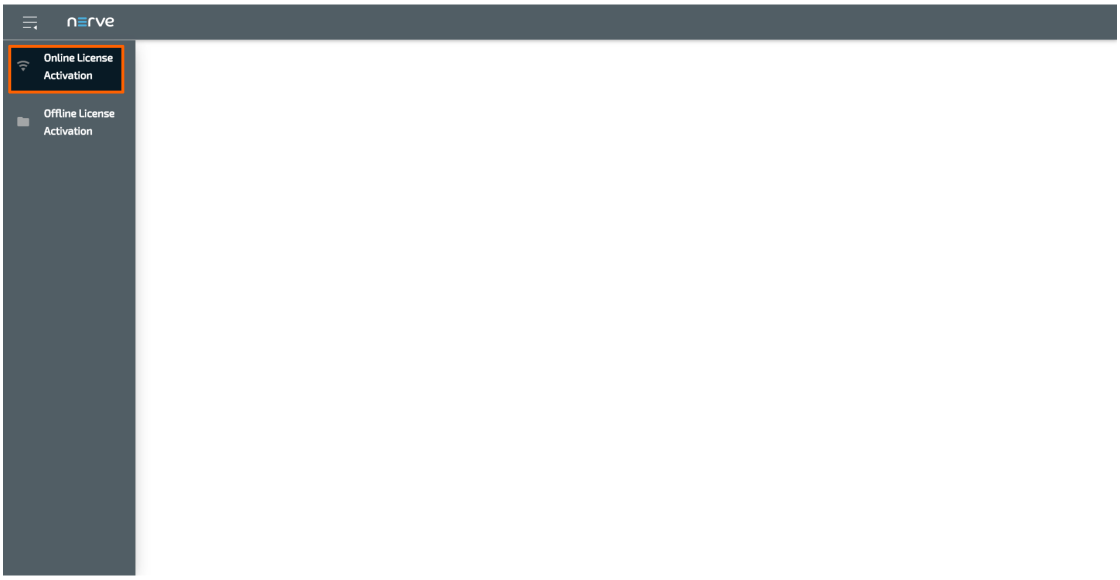
If the node has internet access, the node will connect to the licensing server.

1. Follow the link below to reach the UI for activating the license. This link is Nerve Device specific.

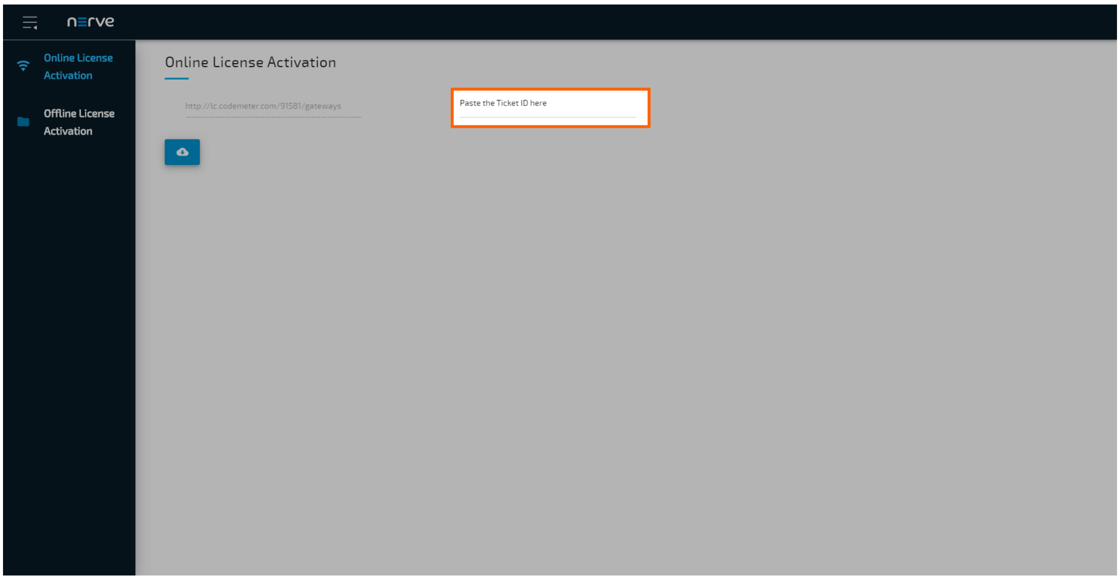
Nerve Device	Physical port	License activation UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.

Nerve Device	Physical port	License activation UI
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

2. Select **Online License Activation** in the navigation on the left.



3. Enter the ticket ID on the right side under **Paste the Ticket ID here**.

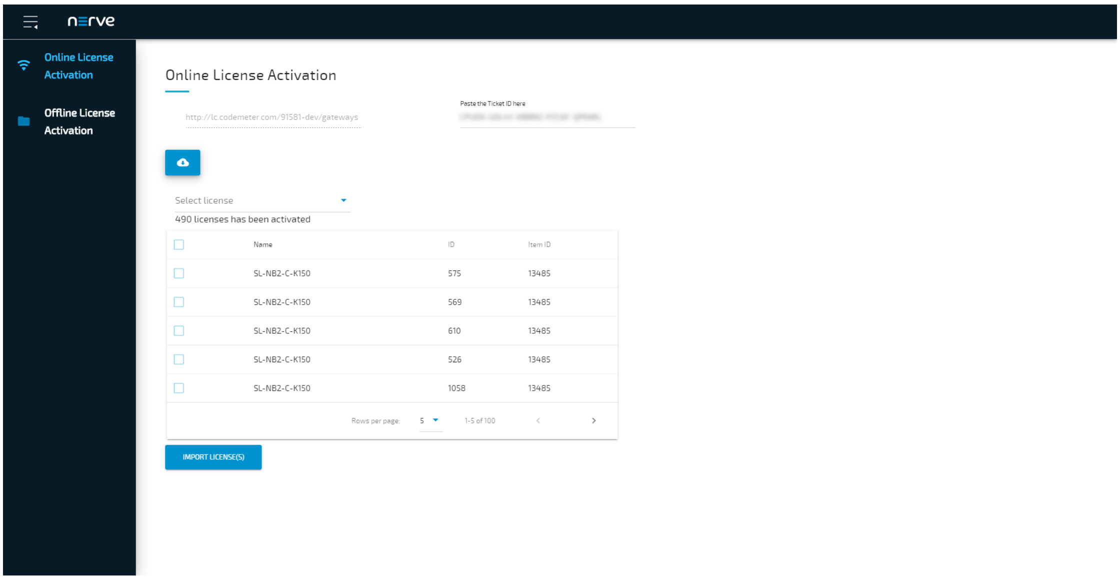


Online License Activation

http://lc.codemeter.com/91581/gateways

Paste the Ticket ID here

4. Click the download button on the left side. Available licenses will appear below.



Online License Activation

http://lc.codemeter.com/91581-dev/gateways

Paste the Ticket ID here
[redacted]

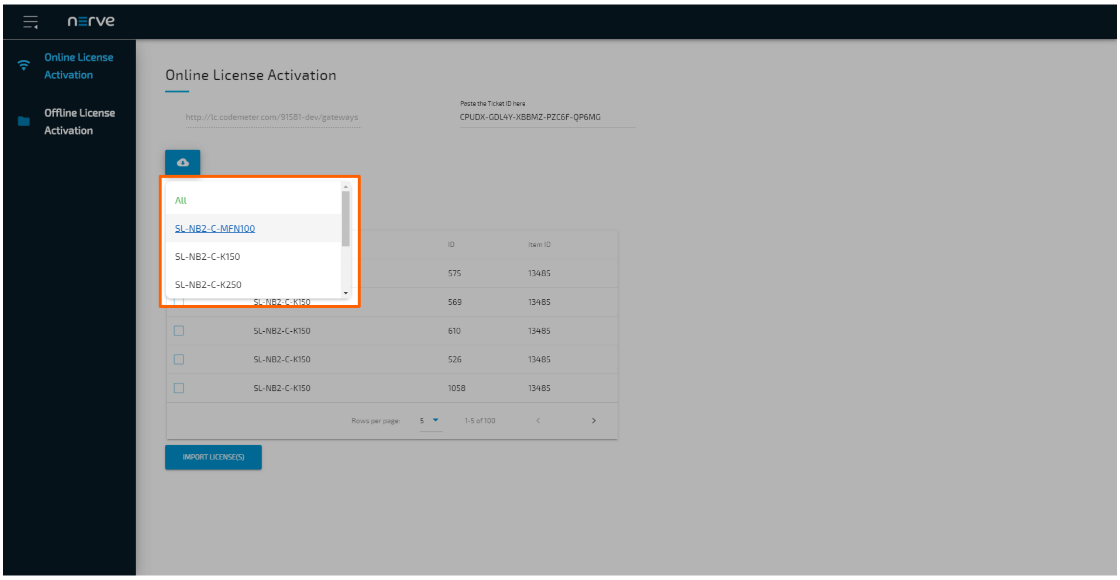
Select license

490 licenses has been activated

□	Name	ID	Item ID
□	SL-NB2-C-K150	575	13485
□	SL-NB2-C-K150	569	13485
□	SL-NB2-C-K150	610	13485
□	SL-NB2-C-K150	526	13485
□	SL-NB2-C-K150	1058	13485

Rows per page: 5 1-5 of 100 ‹ ›

5. Select the used Nerve Device from the drop-down menu. The MFN 100 is selected as an example in the screenshots.


nerve
Online License Activation
Offline License Activation
Online License Activation

http://lc.codemeter.com/91581-dev/gateways

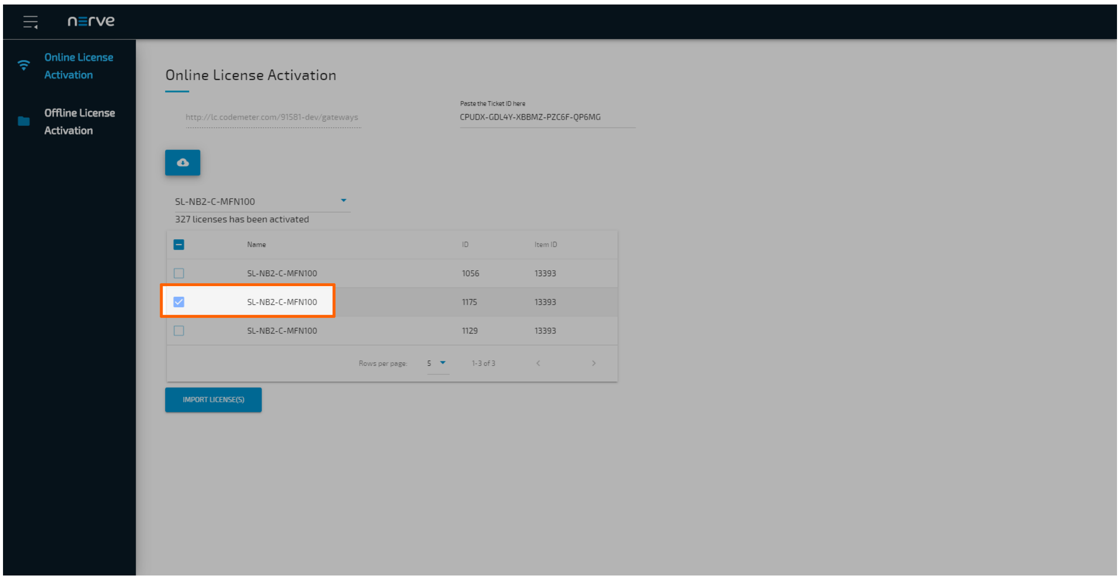
Paste the Ticket ID here: CPUDX-GDL4Y-XBBMZ-PZC6F-QP6MG

- All
- SL-NB2-C-MFN100
- SL-NB2-C-K150
- SL-NB2-C-K250

		ID	Item ID
		575	13485
□	SL-NB2-C-K150	569	13485
□	SL-NB2-C-K150	610	13485
□	SL-NB2-C-K150	526	13485
□	SL-NB2-C-K150	1058	13485

Rows per page: 5 1-5 of 100

6. Tick the checkbox next to a license for the selected Nerve Device. The license will have the device name in its name.


nerve
Online License Activation
Offline License Activation
Online License Activation

http://lc.codemeter.com/91581-dev/gateways

Paste the Ticket ID here: CPUDX-GDL4Y-XBBMZ-PZC6F-QP6MG

SL-NB2-C-MFN100

327 licenses has been activated

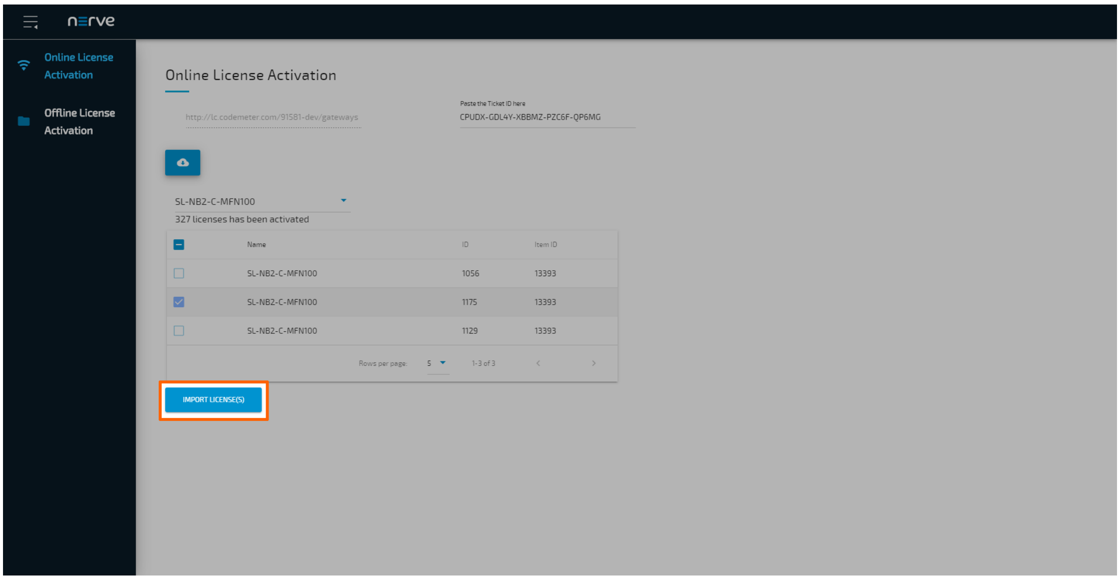
▬	Name	ID	Item ID
□	SL-NB2-C-MFN100	1056	13393
☑	SL-NB2-C-MFN100	1175	13393
□	SL-NB2-C-MFN100	1129	13393

Rows per page: 5 1-3 of 3

NOTE

Make sure to always select a single license per device. Selecting multiple licenses when activating one device will waste purchased licenses.

7. Select **IMPORT LICENSE(S)** below the list of licenses.



Online License Activation

http://lc.codemeter.com/91581-dev/gateways

Paste the Ticket ID here: CPUDX-GDL4Y-XBBMZ-PZC6F-QP6MG

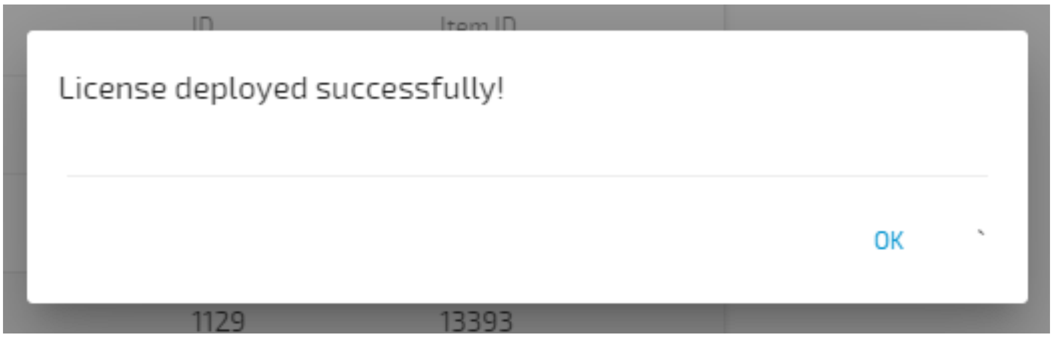
SL-NB2-C-MFN100

327 licenses has been activated

	Name	ID	Item ID
□	SL-NB2-C-MFN100	1056	13393
☑	SL-NB2-C-MFN100	1175	13393
□	SL-NB2-C-MFN100	1129	13393

Rows per page: 5 1-3 of 3

8. Select **OK** when importing the license file is successful.



License deployed successfully!

NOTE

It takes around 30 seconds until the Local UI can be accessed after license activation. Refresh the page until the Local UI login page is displayed.

With the license activated, return to the previous device guide page and continue with **Accessing the Local UI and registering the device**.

Offline license activation

In case of the node not having internet access, the license can be activated with a file-based method. However, note that a workstation with an internet connection is required for connecting to the licensing server in order to upload and download files.

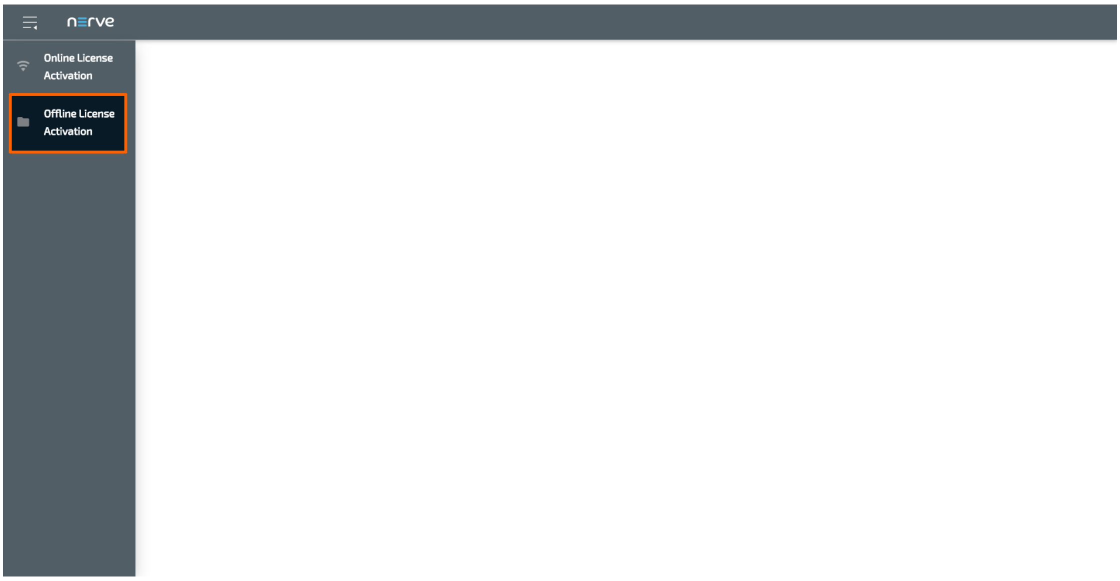
NOTE

Make sure to always select a single license per device. Selecting multiple licenses when activating one device will waste purchased licenses.

1. Follow the link below to reach the UI for activating the license. This link is Nerve Device specific.

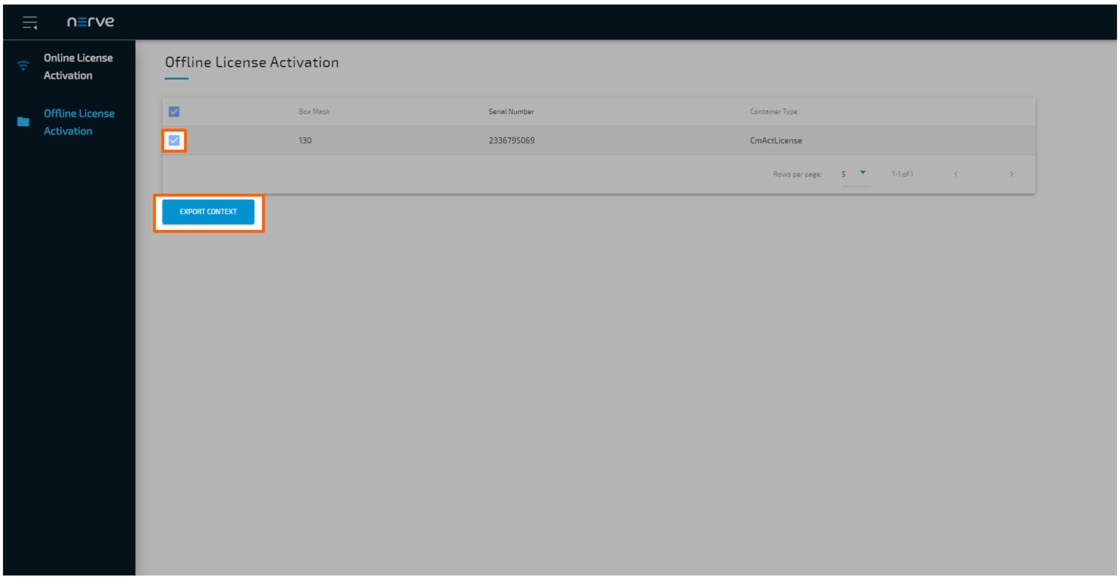
Nerve Device	Physical port	License activation UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

2. Select **Offline License Activation** in the navigation on the left.



3. Tick the checkbox on the left to select the container.

4. Select **EXPORT CONTEXT** to download the context file named `<licensenumber>-WibuCmRaC`.



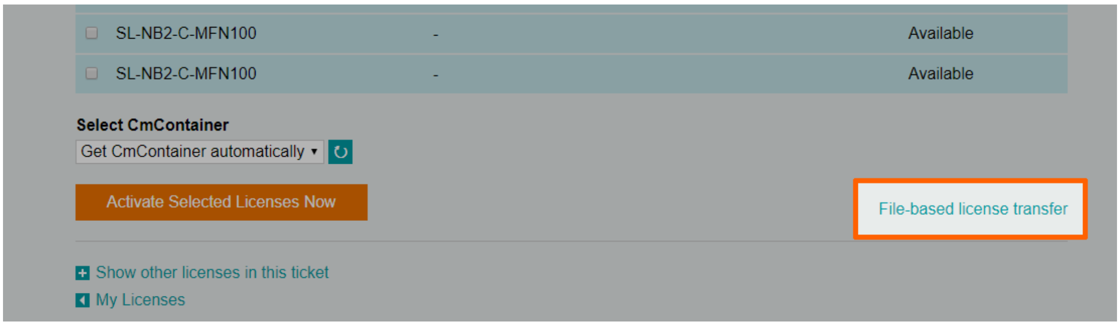
Offline License Activation

☑	Box Mask	Serial Number	Container Type
☑	130	2336795069	CmActLicense

5. Follow the link to the web depot. The link has been sent as part of the delivery.

6. Select a license for the appropriate Nerve Device. The license will have the device name in its name.

7. Select **File-based license transfer** below the list of licenses.

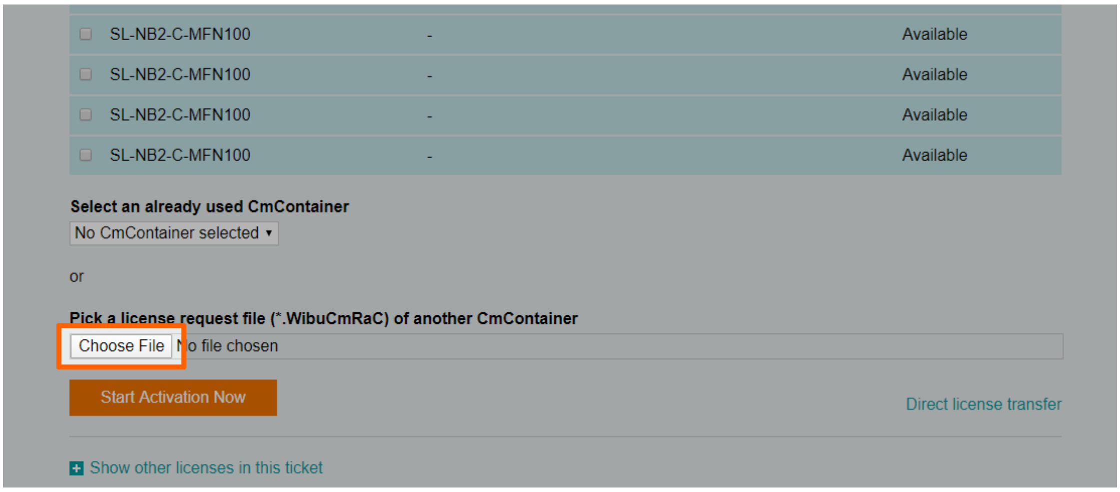

Get CmContainer automatically ▾
Activate Selected Licenses Now
File-based license transfer
Show other licenses in this ticket
My Licenses

□	SL-NB2-C-MFN100	-	Available
□	SL-NB2-C-MFN100	-	Available

Select CmContainer

8. Select a license for the appropriate Nerve Device. The license will have the device name in its name.

9. Select **Choose File** below the list of licenses under **Pick a license request file (*.WibuCmRaC) of another CmContainer** to open the file browser.



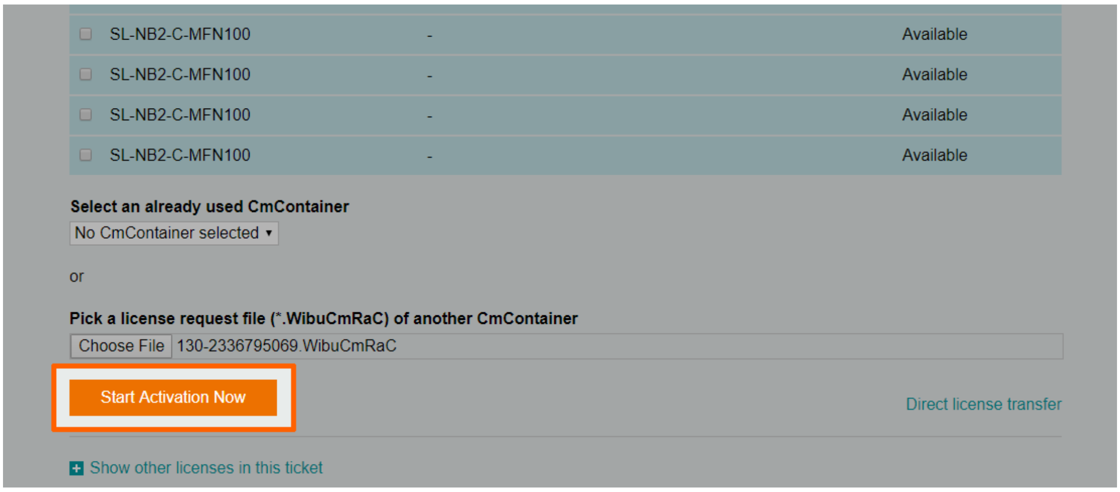
□	SL-NB2-C-MFN100	-	Available
□	SL-NB2-C-MFN100	-	Available
□	SL-NB2-C-MFN100	-	Available
□	SL-NB2-C-MFN100	-	Available

Select an already used CmContainer

or

Pick a license request file (*.WibuCmRaC) of another CmContainer

10. Upload the `<licensenumber>-WibuCmRaC` file that was download before.
11. Select **Start Activation Now**.



□	SL-NB2-C-MFN100	-	Available
□	SL-NB2-C-MFN100	-	Available
□	SL-NB2-C-MFN100	-	Available
□	SL-NB2-C-MFN100	-	Available

Select an already used CmContainer

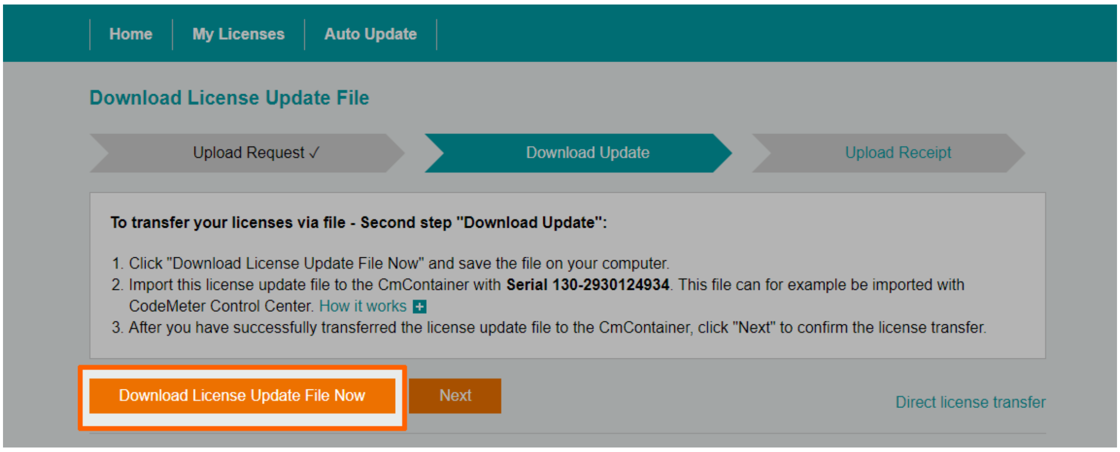
No CmContainer selected ▾

or

Pick a license request file (*.WibuCmRaC) of another CmContainer

Choose File 130-2336795069.WibuCmRaC

12. Select **Download License File Update now** in the next window to download the license update file named `<licensenumber>-WibuCmRaU`.

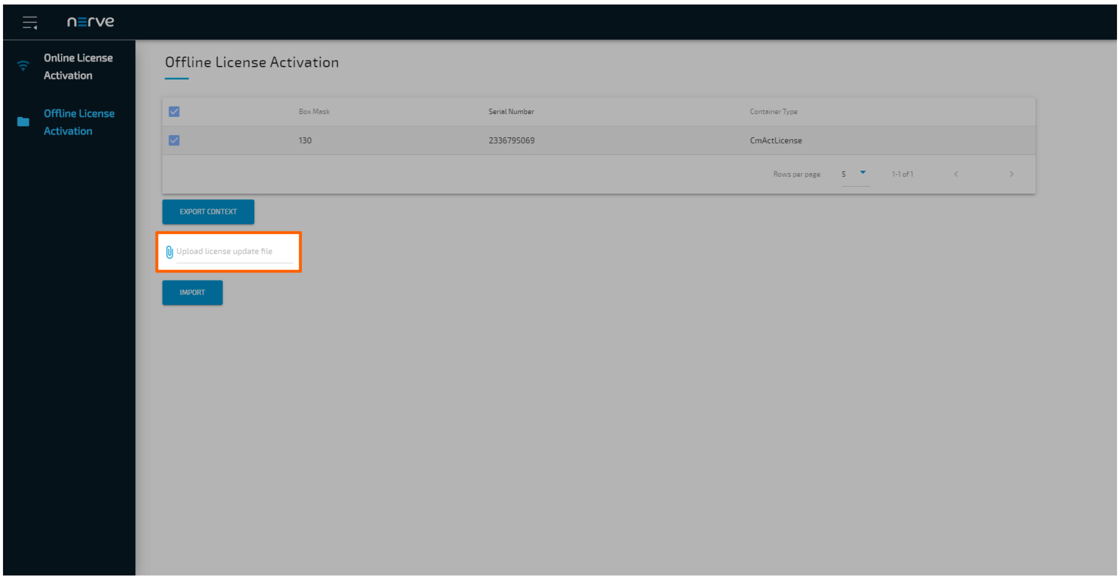


Download License Update File

To transfer your licenses via file - Second step "Download Update":

1. Click "Download License Update File Now" and save the file on your computer.
2. Import this license update file to the CmContainer with **Serial 130-2930124934**. This file can for example be imported with CodeMeter Control Center. How it works ⊞
3. After you have successfully transferred the license update file to the CmContainer, click "Next" to confirm the license transfer.

13. Switch back to the UI for activating the license.
14. Select the **Upload license update file** field to open the file browser.



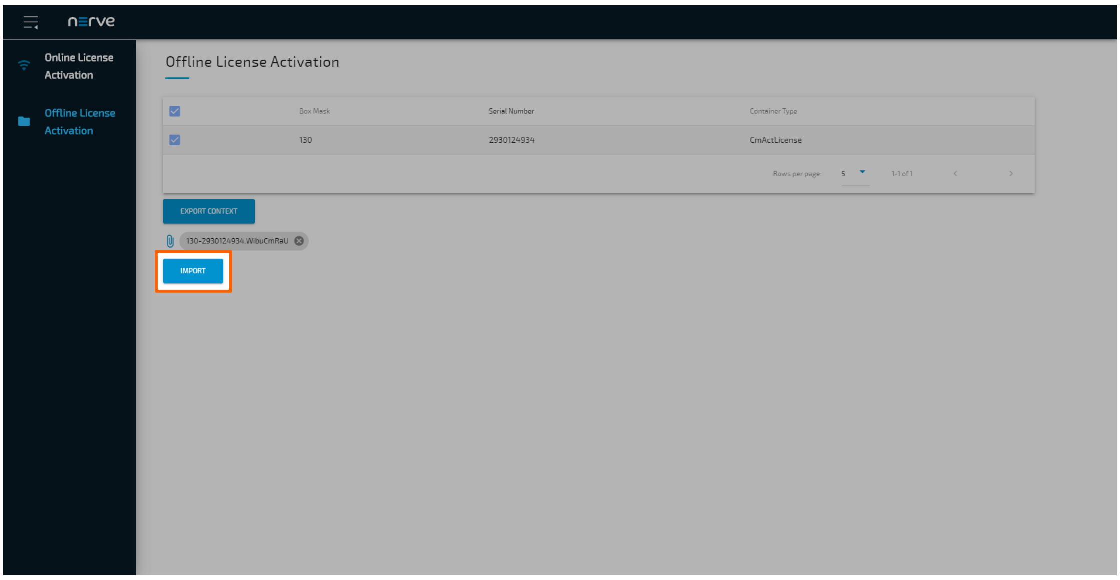
nerve
Online License Activation
Offline License Activation
EXPORT CONTEXT
Upload license update file
IMPORT
Rows per page: 5
1-1 of 1

Offline License Activation

☑	Box Mask	Serial Number	Container Type
☑	130	2336795069	CmActLicense

15. Navigate to where the `<licensenumber>-WibuCmRaU` file is saved and select it.

16. Select **IMPORT LICENSE(S)** to import the license.

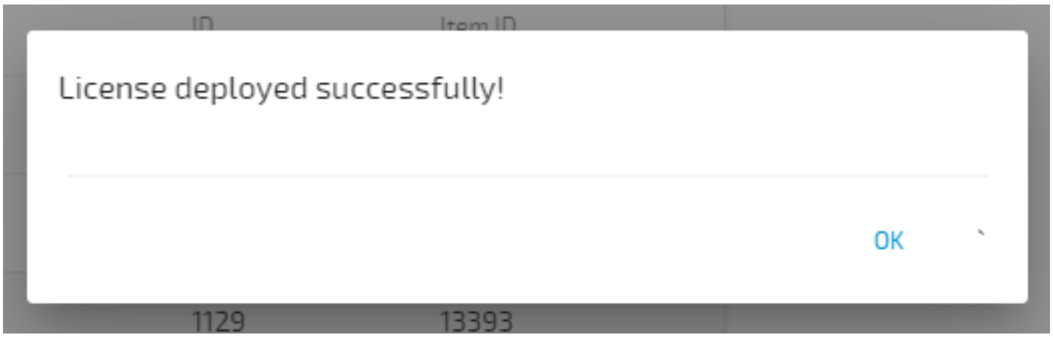


Offline License Activation

☑	Box Mask	Serial Number	Container Type
☑	130	2930124934	CmActLicense

130-2930124934.WibuCmRaU

17. Select **OK** when importing the license file is successful.



License deployed successfully!

NOTE

It takes around 30 seconds until the Local UI can be accessed after license activation. Refresh the page until the Local UI login page is displayed.

With the license activated, return to the previous device guide page and continue with **Accessing the Local UI and registering the device**.

Local UI

The Local UI is provided by a web server that is running locally on the Nerve Device. Compared to the Management System, the Local UI covers features that only concern the node itself.
The Local UI gives access to the following features:

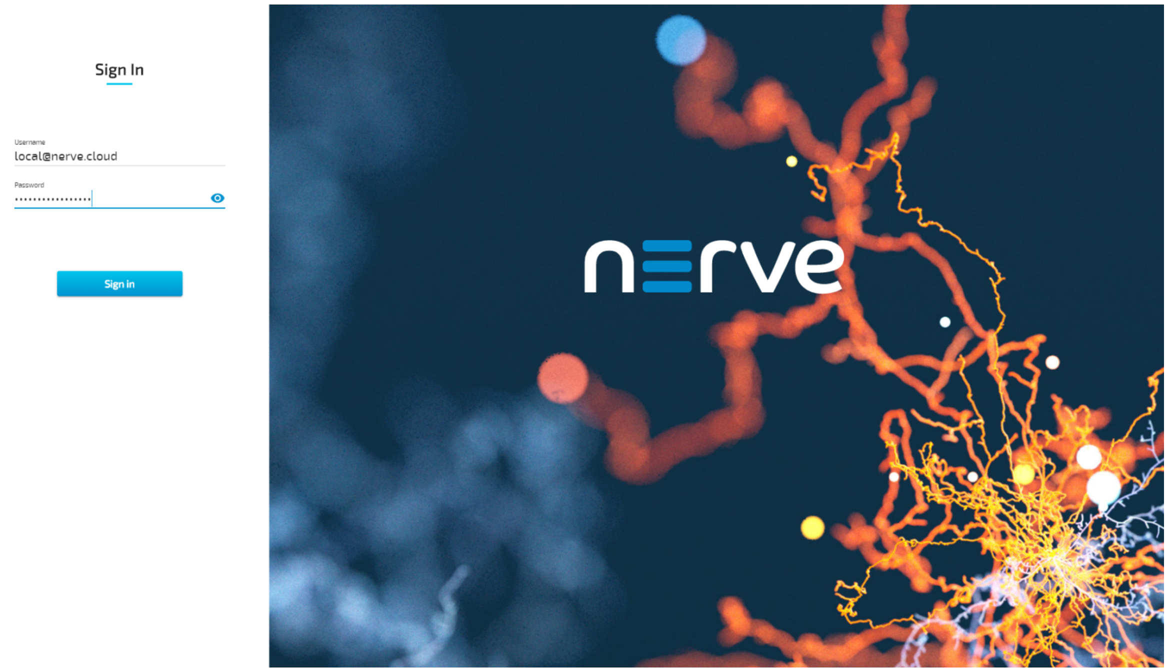
- Network configuration
- Node registration and password management
- Workload management
- Local workload deployment
- Local workload repository
- Management of remote connections

Connecting to the Local UI

Connecting to the Local UI depends on the Nerve Device. Refer to the table below on how to reach the Local UI for each Nerve Device. Make sure to connect a workstation to the physical port of the Nerve Device associated with host access and configure the network adapter of the workstation. The IP address has to be in the same range as the IP address of the host access interface with a `255.255.255.0` subnet mask.

Nerve Device	Physical port	License activation UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

Once a connection is established, log in with the credentials for the Local UI to reach the Local UI dashboard.



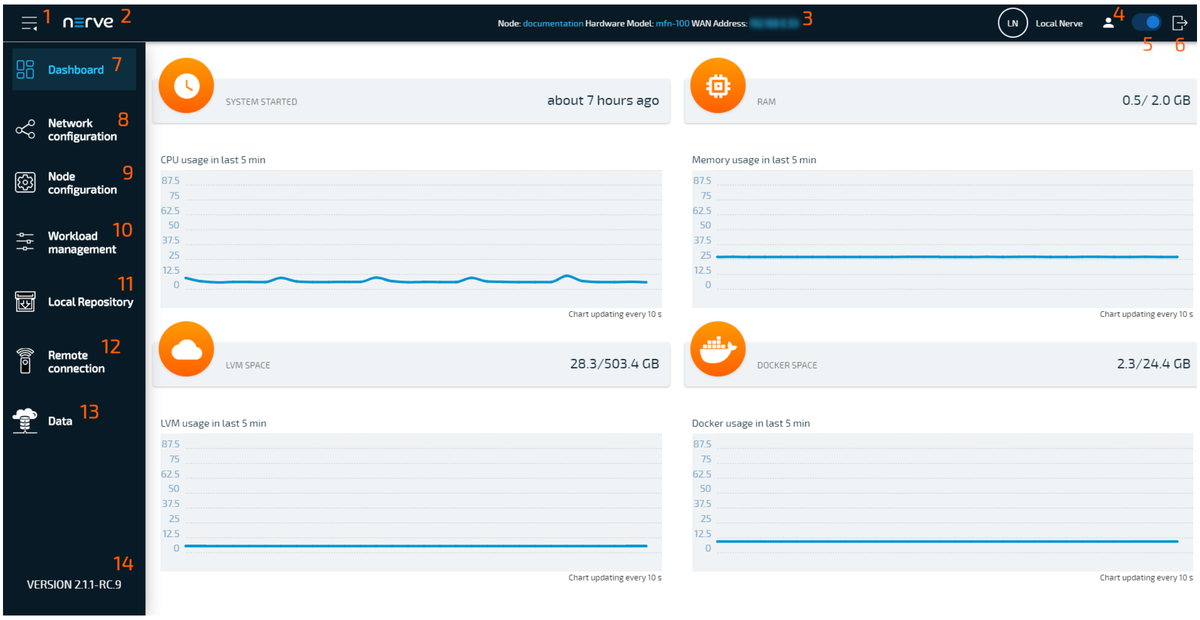
Sign In

Username: local@nerve.cloud

Password: ••••••••••••••••

Local UI dashboard

The dashboard of the Local UI is the default screen after the log in. Usage statistics of the Nerve Device are displayed in the window with more options in the menu on the left side.



Node: documentation Hardware Model: mfn-100 WAN Address: [redacted]

SYSTEM STARTED: about 7 hours ago

RAM: 0.5/ 2.0 GB

LVM SPACE: 28.3/503.4 GB

DOCKER SPACE: 2.3/24.4 GB

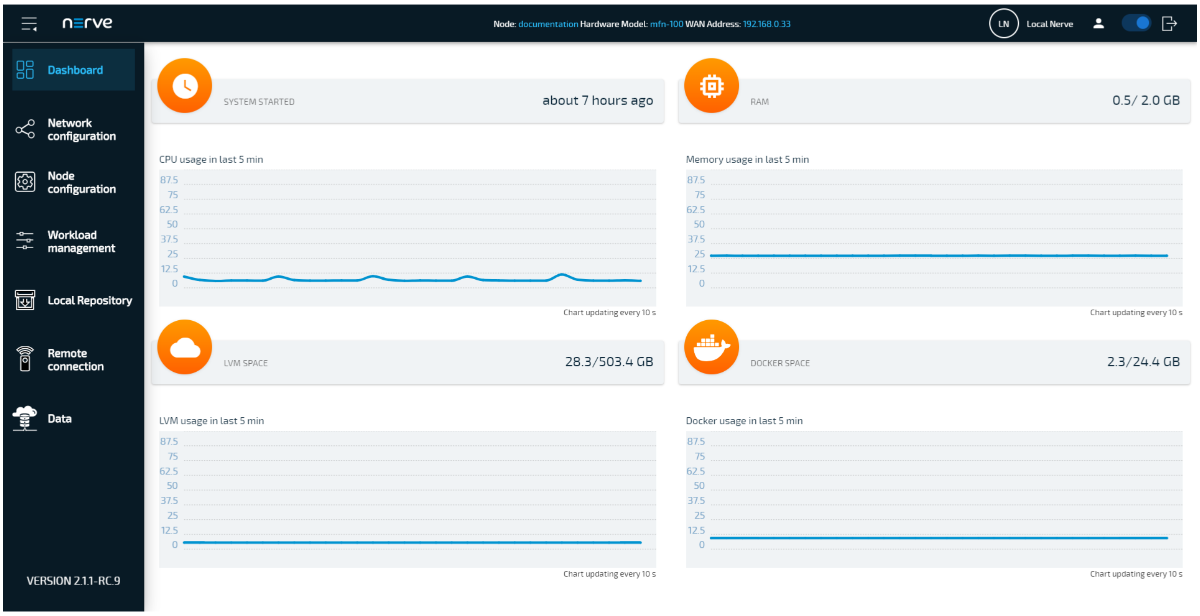
VERSION 2.1.1-RC.9

Term	Description
Burger menu (1)	Expand and collapse the left-hand menu by clicking here.
Nerve logo (2)	Click here to return to the dashboard and reload the page.
Node details (3)	Details of the node are displayed here, showing the name of the node in the Management System, the hardware model and the WAN address of the node.
Change password (4)	Clicking here leads to an area where the password to the Local UI can be changed.

Term	Description
Connect button (5)	Select the slider here to disconnect the node from the network. Blue indicates an active connection to the network. If the button is white, then the node is offline.
Log out (6)	Click here to log out of the Local UI.
Dashboard (7)	Select this to display the dashboard containing the system metrics — graphs showing available resources of the Nerve Device and their usage over time.
Network configuration (8)	This menu allows to configure the Ethernet ports of the Nerve Device.
Node configuration (9)	Configure data such as the Management System URL that the node will connect to or the serial number of the node here. The information required to register a node in the Management System is configured in this menu. Also, passwords can be changed here.
Workload management (10)	Select this menu for control options for deployed workloads and manual deployment of workloads.
Local Repository (11)	Find settings for the configuration of a local workload repository here.
Remote connection (12)	Manage incoming and active remote connections here.
Data (13)	Access the instance of the Nerve Data Services in the Local UI here. Refer to Nerve Data Services for more information.
Node version (14)	This is the currently installed version of the node.

System metrics

The graphs in the Local UI dashboard show available resources of the Nerve Device and their usage over time. The y-axis displays percentages and the x-axis is updated every 10 seconds, showing a time span of 5 minutes. The percentages displayed are always in relation to the maximum of the available resource:



Node: documentation Hardware Model: mfn-100 WAN Address: 192.168.0.33

SYSTEM STARTED about 7 hours ago

RAM 0.5/ 2.0 GB

LVM SPACE 28.3/503.4 GB

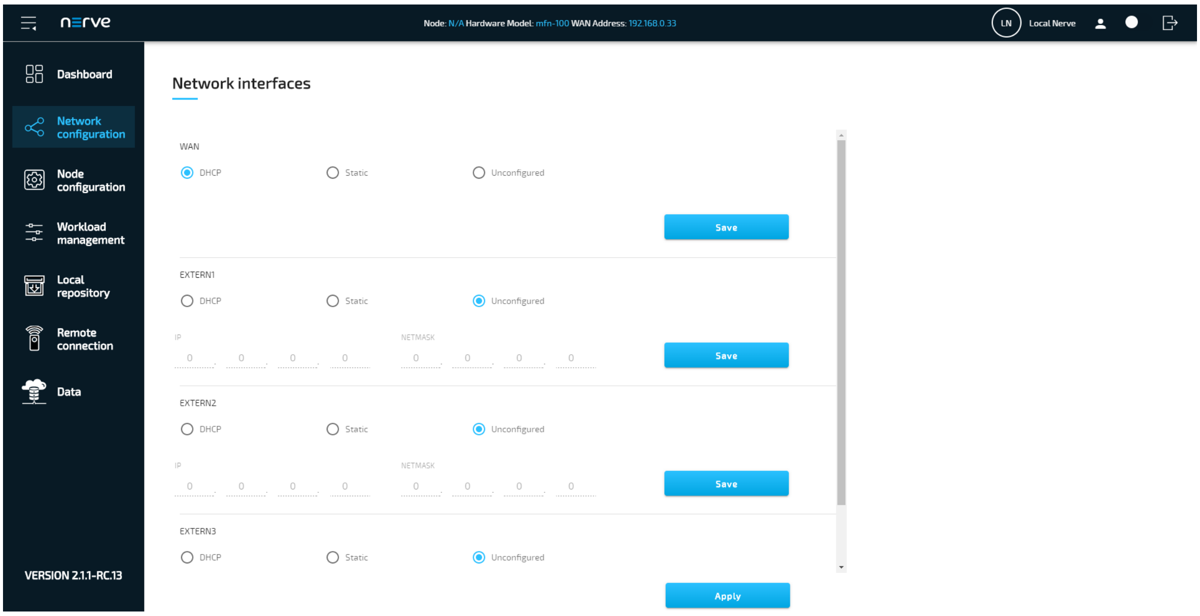
DOCKER SPACE 2.3/24.4 GB

VERSION 2.1.1-RC.9

Item	Description
SYSTEM STARTED	This shows how long the Nerve Device has been running. If the device is restarted, this value is reset.
CPU usage in last 5 min	The graph here shows the percentage of processing power that is being used. This includes CPUs that have been assigned to VMs and Docker containers.
RAM	This shows how much memory is used (left value) and how much memory is available in total (right value). Example: 0.3/1.9 GB Note that the total amount of memory the Nerve Device offers is not shown here. This is the memory that is available for the host.
Memory usage in last 5 min	Similar to CPU usage, the graph shows the percentage of memory used. This includes memory that has been assigned to VMs or Docker containers.
LVM SPACE	Virtual machines have their dedicated virtual partition (Logical Volume Manager). The values show how much of this partition is used (left value) and how much is available in total (right value). Example: 86.5/238.5 GB
LVM usage in the last 5 min	This graph shows the percentage of space that is being used by the Logical Volume Manager.
DOCKER SPACE	Similar to LVM space, Docker containers have their dedicated virtual partition. The values show how much of this partition is used (left value) and how much is available in total (right value). Example: 2.9/40.3 GB
Docker usage in the last 5 min	This graph shows the percentage of space for Docker containers that is being used.

Local network configuration

From the Local UI, the Ethernet ports of the Nerve Device can be configured. Select **Network configuration** in the navigation on the left to reach this menu. The example below is of the MFN 100. The page is specific to the Nerve Device. The number and names of interfaces may differ.



Node: N/A Hardware Model: mfn-100 WAN Address: 192.168.0.33

Network interfaces

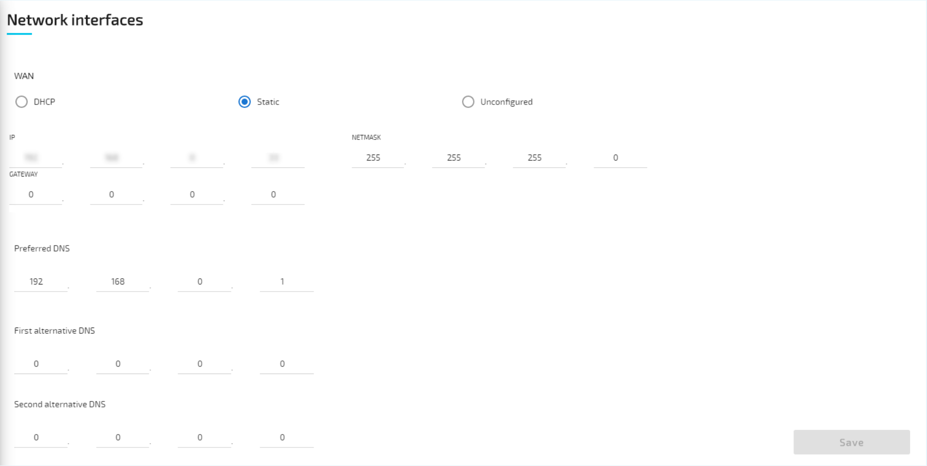
WAN
● DHCP ○ Static ○ Unconfigured

EXTERN1
○ DHCP ○ Static ● Unconfigured
IP 0 . 0 . 0 . 0 NETMASK 0 . 0 . 0 . 0

EXTERN2
○ DHCP ○ Static ● Unconfigured
IP 0 . 0 . 0 . 0 NETMASK 0 . 0 . 0 . 0

EXTERN3
○ DHCP ○ Static ● Unconfigured

Item	Description
DHCP	The IP address of the port will be assigned by the DHCP server. If an IP address has been assigned, it will be displayed here.

Item	Description
Static	By selecting **Static**, the IP address of the port needs to be manually defined. Enter the IP address and subnet mask under **IP** and **NETMASK** to set a static IP address. For the WAN interface, the **GATEWAY**, the preferred **DNS** as well as two alternative DNS can also be set.  Network interfaces WAN ○ DHCP ● Static ○ Unconfigured IP [redacted] NETMASK 255 . 255 . 255 . 0 GATEWAY 0 . 0 . 0 . 0 Preferred DNS 192 . 168 . 0 . 1 First alternative DNS 0 . 0 . 0 . 0 Second alternative DNS 0 . 0 . 0 . 0 Save
Unconfigured	If **Unconfigured** is checked, the port is disabled for the host but can still be used for virtual machines with bridged interfaces.

NOTE

For more information on networking and interfaces refer to the networking chapter.

Node configuration

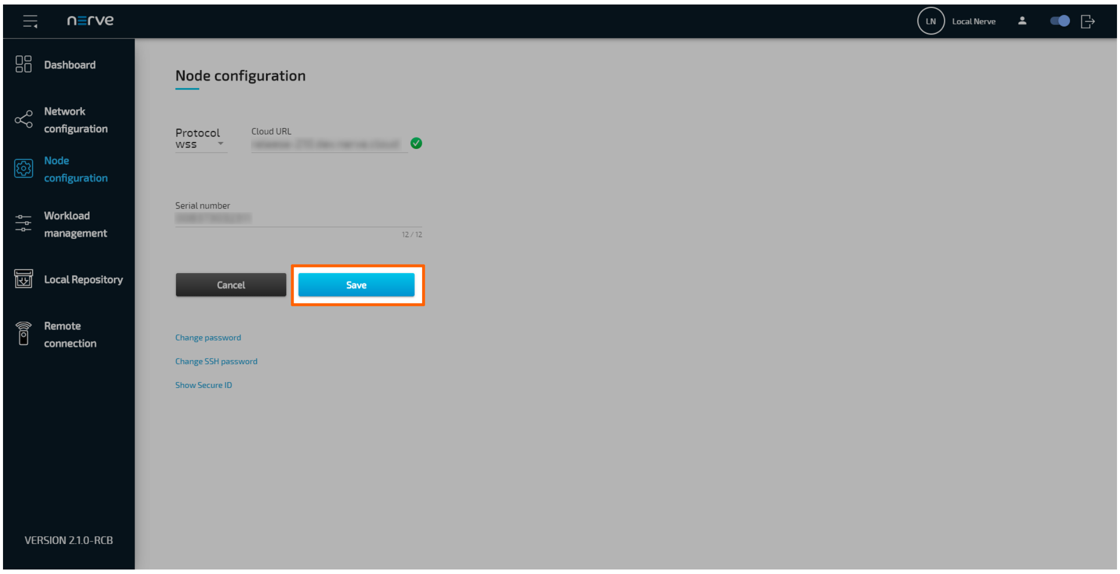
The first part of registering a node in the Management System is performed in the Local UI. Node details and information required for registering a node in the Management System are configured under **Node configuration**.

1. Select **Node configuration** in the navigation on the left.
2. Enter the following information:

Protocol	Select **wss** or **ssl** from the drop-down menu: ◦ **wss** Selecting this will use the WebSocket Secure protocol for the registration of the node, meaning that port 443 will be used for communication between the node and the Management System. ◦ **ssl** Selecting this will use the Secure Sockets Layer protocol for the registration of the node, meaning that port 8883 will be used for communication between the node and the Management System. Note that port 8883 of the company firewall has to be open when using the SSL protocol.
Cloud URL	Enter the URL of the Management System without the protocol, e.g `example.nerve.cloud`.

Serial number	Enter a serial number with a minimum of 12 characters. Note that this serial number can be freely defined. It is required for node registration in the Management System and serves as a means of identification. Entering the serial number that is printed on the Nerve Device is not required but recommended.

3. Click **Save**.



Node configuration

Protocol
WSS

Cloud URL
[redacted]

Serial number
[redacted]
12 / 12

With the serial number saved in the node configuration, the secure ID has been generated and can be displayed by selecting **Show Secure ID**. This secure ID is required when adding the node in the Management System. Refer to Adding a node on how to add nodes to the Management System.

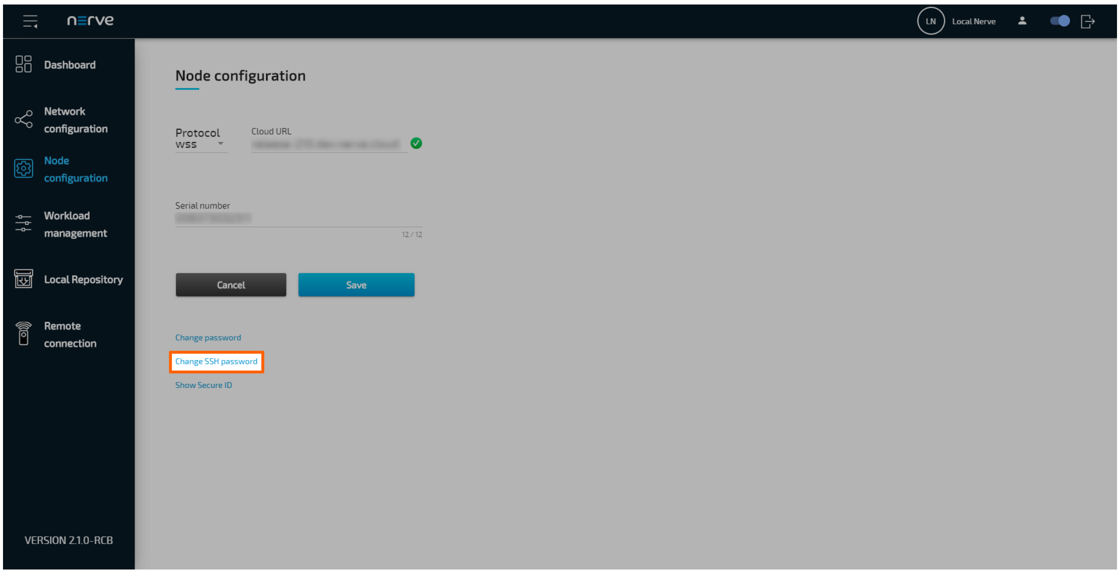
Changing the password for host access

The password for host access via SSH can be changed. The default password for host access is found in the customer profile.

> **NOTE**
>
> Access to the Linux host system of Nerve Blue is provided in order to enable advanced use cases. Using host access requires expert Linux knowledge as system internal changes can be performed. Note that changes may impact the Nerve Blue system.

1. Select **Node configuration** in the navigation on the left.

2. Select **Change SSH password**



Node configuration

Protocol
wss

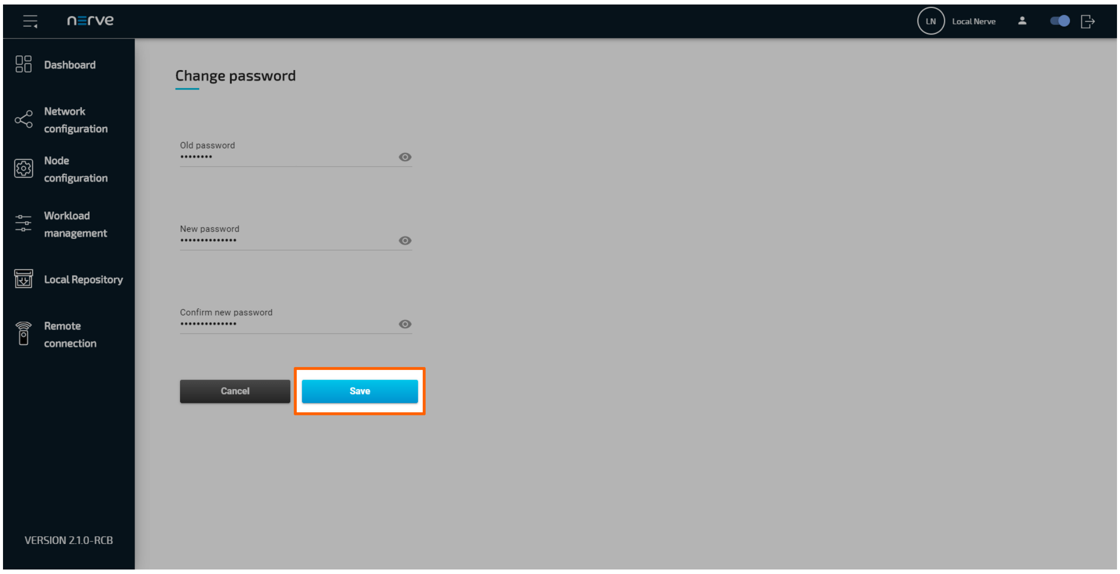
Cloud URL [redacted]

Serial number [redacted]
12 / 12

3. Enter the following information:

Item	Description
Old password	Enter the old password for host access.
New password	Enter the new password here. The new password must be 8 characters or longer and it can only consist of alphanumeric characters.
Confirm new password	Enter the new password again. Both passwords must match in order to proceed.

4. Select **Save** to set the new password.



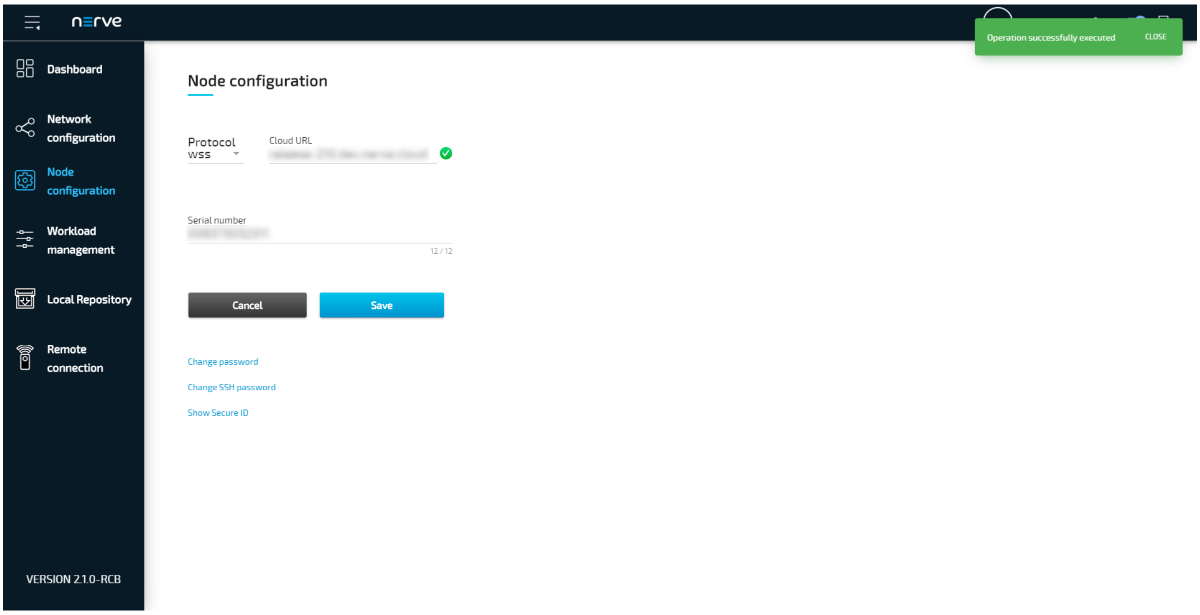
Change password

Old password
••••••••

New password
••••••••••••••

Confirm new password
••••••••••••••

If the process was successful, the Local UI will display the dashboard with a green notice confirming the change in the upper-right corner.



Operation successfully executed

Node configuration

Protocol: wss

Cloud URL: [redacted]

Serial number: [redacted] 12 / 12

Change password

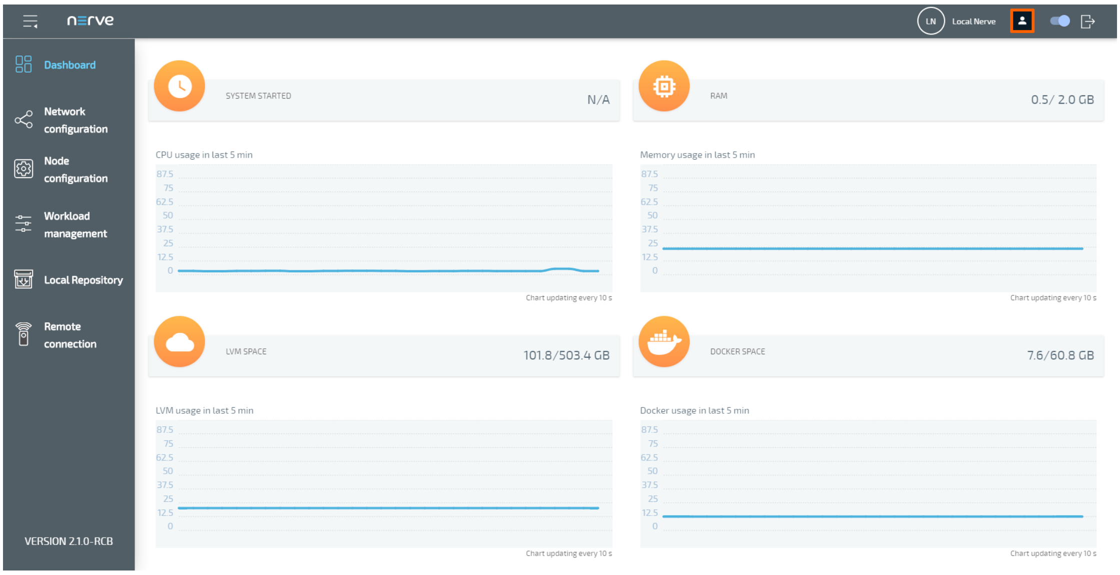
Change SSH password

Show Secure ID

Changing the password for the Local UI

The password to the Local UI can be changed. The default password for Local UI access can be found in the customer profile.

1. Select the user icon (**Change password**) in the upper-right.



SYSTEM STARTED N/A

RAM 0.5/ 2.0 GB

LVM SPACE 101.8/503.4 GB

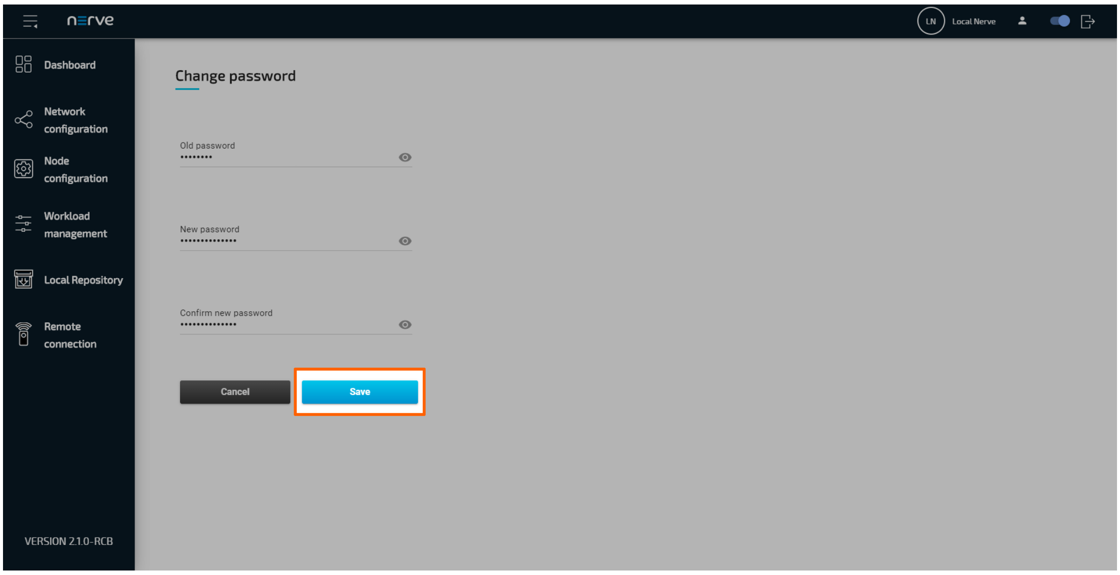
DOCKER SPACE 7.6/60.8 GB

> **NOTE**
>
> Alternatively, it is also possible to change the password in the Node configuration menu. Select **Node configuration** in the navigation on the left and click **Change password** to reach the password form.

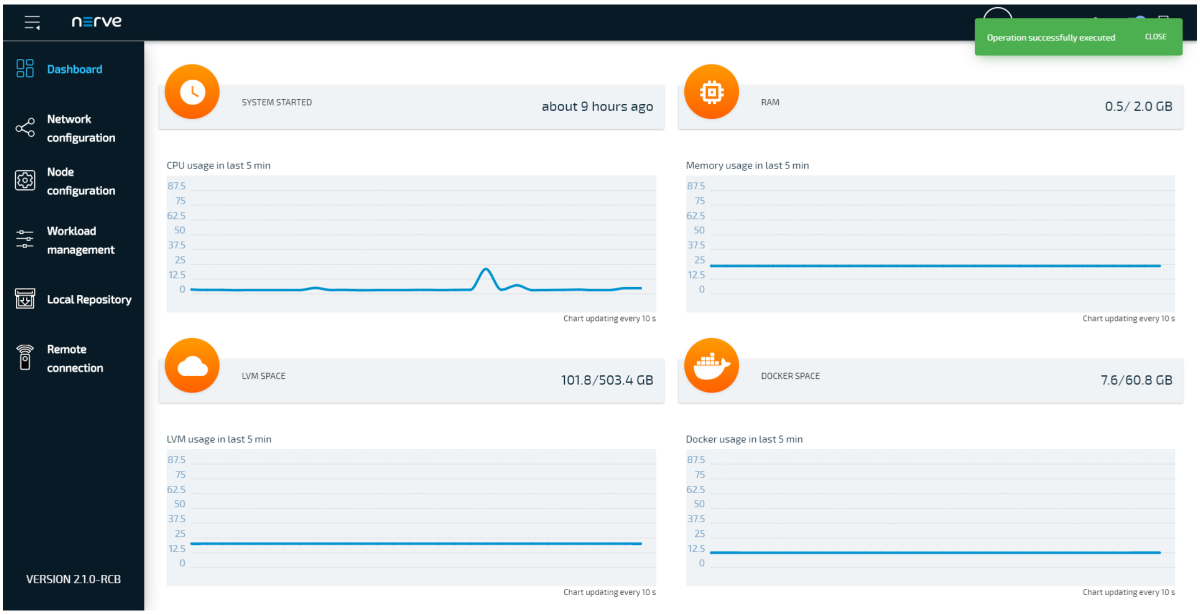
2. Enter the following information:

Item	Description
Old password	Enter the old password to the Local UI.
New password	Enter the new password here. The new password must be 8 characters or longer and it can only consist of alphanumeric characters.
Confirm new password	Enter the new password again. Both passwords must match in order to proceed.

3. Select **Save** to set the new password.

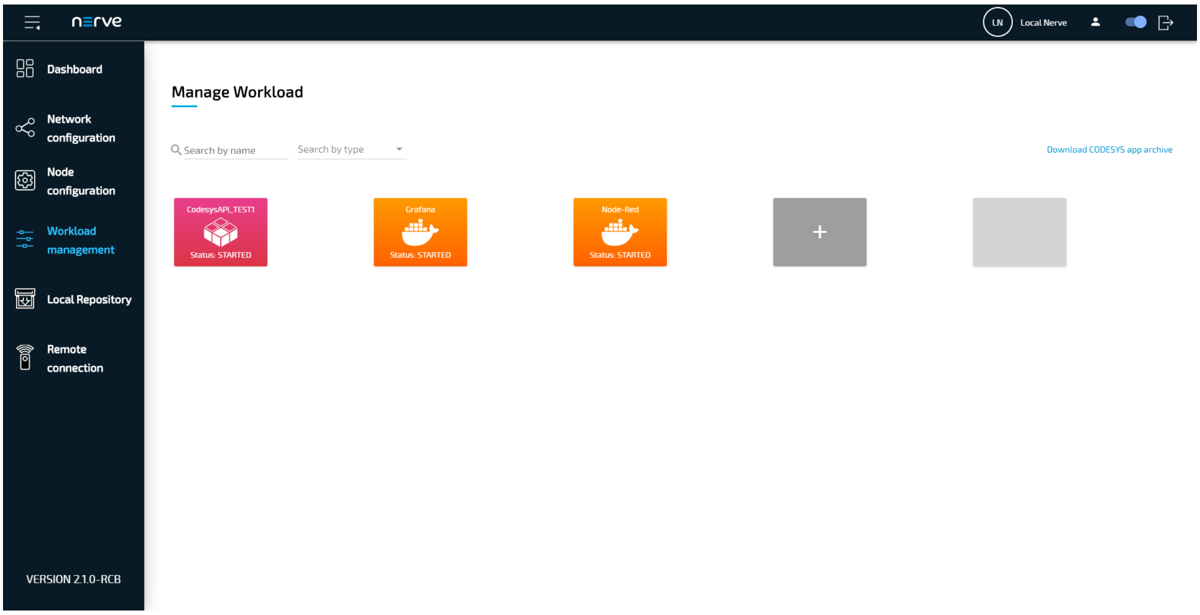


If the process was successful, the Local UI will display the dashboard with a green notice confirming the change in the upper-right corner.

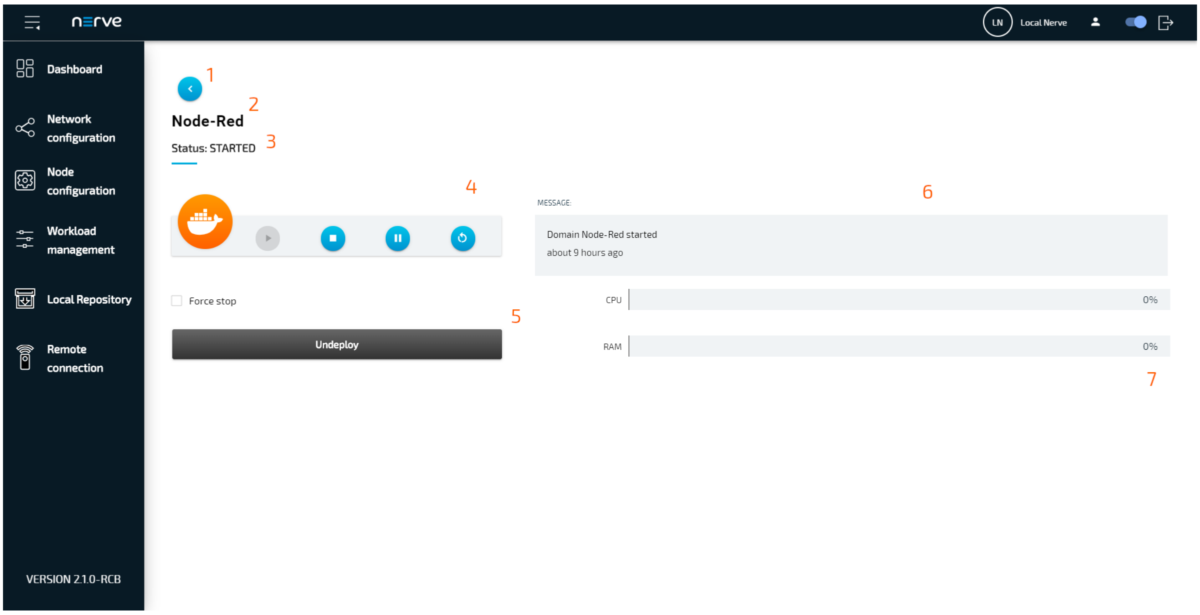


Workload management

Deployed workloads can be controlled in the Local UI. Select **Workload management** in the navigation menu on the left-hand side for an overview of all deployed workloads. Then, select a workload to reach the interface for controlling workloads running on the Nerve Device. If there is a large number of workloads deployed, filter the list by name or by type. Enter a search query under **Search by name** to filter the list by name. Select **All**, **VM**, **Codesys** and **Docker** from the drop-down menu to filter workloads by type.



Note that CODESYS workloads can only be controlled in the Local UI, as operation of a CODESYS workload may have an impact on machine operation and therefore should not be controlled remotely. Virtual machine workloads and Docker workloads can also be controlled in the Management System. Refer to the workload control section in the node tree chapter on how to control workloads in the Management System.



Domain Node-Red started
about 9 hours ago

Item	Description
Back button (1)	Click here to return to overview of workloads.
Workload name (2)	This is the name of the workload. The name of the workload version is not displayed.

Item	Description
Workload status (3)	The current status of the workload is displayed here. The possible statuses are the following: • **Idle** This is the initial state of the workload before it is started. • **Creating** This is a transitional state of the workload when it is being created on the node. • **Starting** This is a transitional state when the workload is being started. • **Restarting** This is a transitional state when the workload is being restarted. • **Started** The workload is running and operating. • **Suspending** This is a transitional state when the workload is being suspended. • **Suspended** The workload has been paused. • **Resuming** This is a transitional state when the workload is being resumed from the suspended state. • **Stopping** This is a transitional state when the workload is being stopped. • **Stopped** The workload has been stopped. • **Removing** This is a transitional state when the workload is being undeployed. • **Error** An unknown error has occured.
Control panel (4)	There are five control options for workloads here: • **Play** If the workload is in a stopped state, clicking **Play** will start the workload. • **Stop** If the workload is running, clicking **Stop** will stop the workload. • **Suspend** Clicking **Suspend** will pause the workload. It can be continued by clicking **Play**. • **Restart** This will restart the workload. • **Force stop** Setting a value here will force the workload into the stopped state once **Stop** is clicked and the set time has expired. Note that the control panel for CODESYS workloads only offers **Play**, **Stop** and **Force stop**. Also, stopping a CODESYS workload will reset it to its initial values unless the CODESYS application has been written using the retain variables library. Refer to Enabling retain variables for more information.
Undeploy (5)	Clicking here removes the workload from the node. To deploy the workload again, it has to be deployed through the Local UI or the Management System.

Item	Description
Message window (6)	The message window displays the latest message the workload has sent out including how much time has passed since it was sent out. The type of message that is displayed here depends on the workload. Here is a list of messages that are valid for VMs and Docker containers: • "Domain creating." • "ERROR during creating! <errormessage>" • "Domain starting." • "ERROR during starting! <errormessage>" • "Domain <domainname> started." • "Domain stopping." • "ERROR during stopping! <errormessage>" • "Domain <domainname> stopped." • "Domain suspending." • "ERROR during suspending! <errormessage>" • "Domain <domainname> suspended." • "Domain resuming." • "ERROR during resuming! <errormessage>" • "Domain restarting." • "ERROR during restarting." • "Domain removing!!!" • "ERROR during removing." • "ERROR!!! Domain stopping." In the messages above, <domainname> is a placeholder for the name of the VM or Docker. In case of Docker containers, <errormessage> signifies a message that is generated by the Docker container if an error occurs. For VMs, there is an additional set of messages: • "Failed to connect to hypervisor." • "Failed to create domain." • "Domain <domainname> created." • "Cannot start <domainname> domain because it may already be running!" • "Failed to resume <domainname> domain!" <errormessage> • "Failed to start domain <domainname>. " <errormessage> In this case, <errormessage> is a message that is fetched from the libvirt library. CODESYS workloads have the following set of messages: • "Preparing files for installation" • "Starting CODESYS application" • "CODESYS application started" • "Stopping CODESYS application" • "CODESYS application stoppped" • "Removing CODESYS application file" • "An unexpected error has occurred. <errormessage>" Here, <errormessage> is a message that is sent between the node and CODESYS.

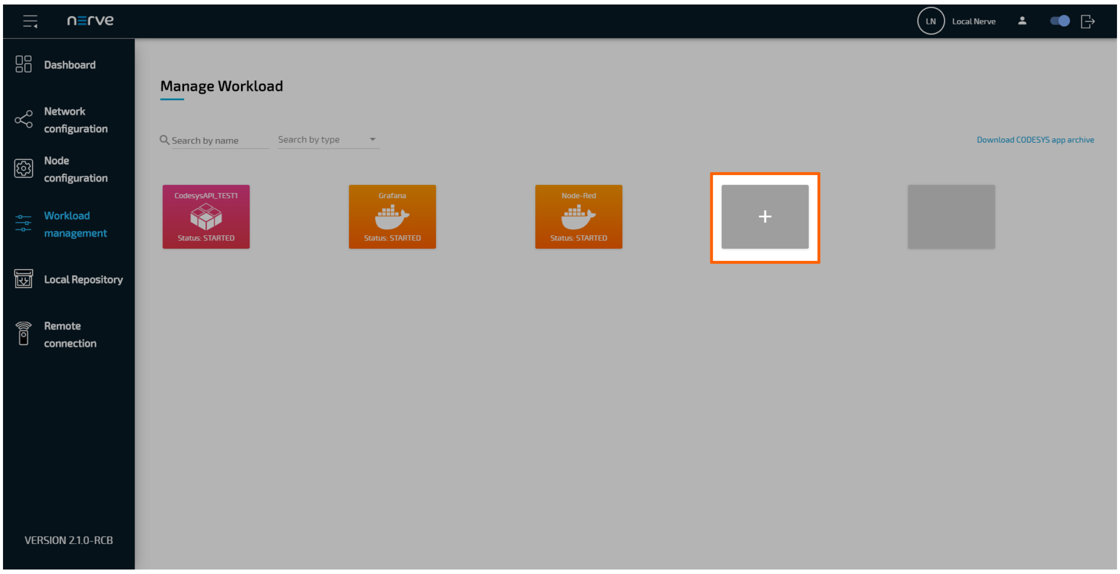
Item	Description
Usage statistics (7)	Virtual Machine workloads and Docker workloads have their assigned resources they can use. The use of these resources is displayed with bar graphs: • **CPU (VM and Docker)** The percentage here shows the usage of CPU resources in relation to the assigned CPUs. Example: A VM is assigned one CPU core out of four and the core is at 75 % usage capacity. The graph will be at 75 %. • **RAM (Docker only)** Similar to the CPU usage statistic, the percentage here shows the usage of system memory resources in relation to the assigned memory. If the assigned memory is at a 100 % usage capacity, the graph will be at 100 %. If no memory has been assigned, the graph will show the percentage of used memory in relation to the total available memory of the host.

All changes performed in the Local UI are reflected in the Management System.

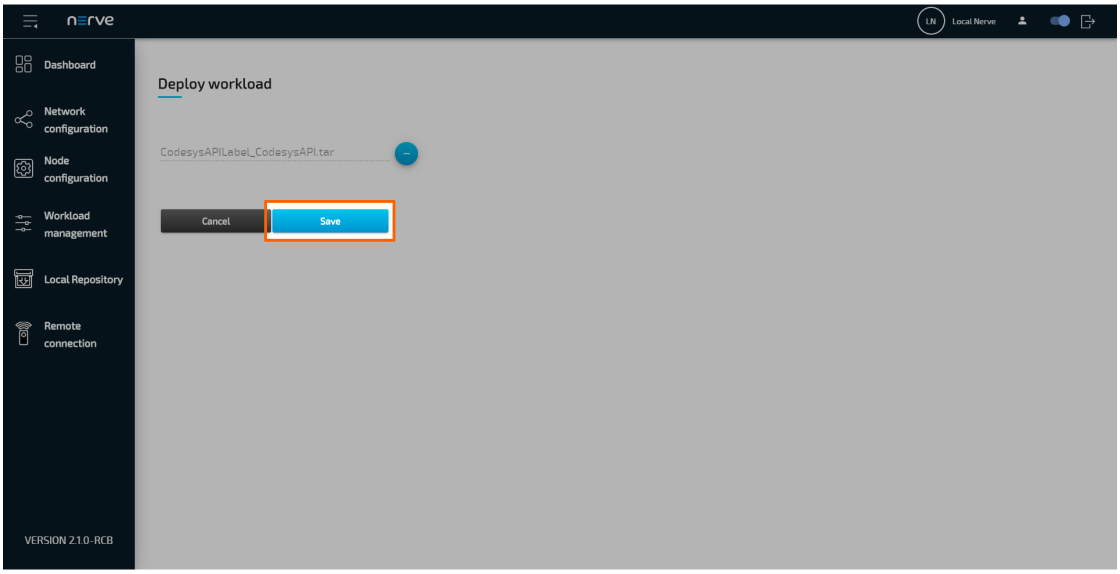
Local workload deployment

Workloads that have been provisioned in the Management System can be exported and then deployed directly through the Local UI. Refer to the workloads chapter on how to export a workload before following the instructions below.

1. Export a workload in the Management System.
2. Switch to the Local UI.
3. Select **Workload management** in the navigation on the left side.
4. Click the plus tile in the middle.



5. Select the plus symbol to open the file browser.
6. Add the TAR file containing the exported workload.
7. Select **Save** to deploy the workload.



CodesysAPILabel_CodesysAPI.tar

Setting a local repository

If desired, the required files for deploying workloads can be stored in a local repository that does not require internet connection. In doing so, the workload image files for deployment are taken from a user defined repository instead of the workload repository in the Management System.
Refer to Exporting a Workload for information on how to obtain the workload images. Also note that every workload needs to be provisioned in the Management System once before the image files can be transferred to a local repository.

A web server that services static files is required. Popular web servers like Nginx or Apache HTTP Server can be used, as well as a Network-attached storage (NAS) device. Recommended are:

- the NodeJS http-server for Linux
- the built-in Internet Information Services (IIS) for Windows

The IIS for Windows needs to be activated first in newer versions of Windows like Windows 8 and Windows 10.

1. Open the **Control Panel**
2. Navigate to **Control Panel > Programs > Programs and Features**.
3. Select **Turn Windows features on or off**
4. Tick the checkbox next to **Internet Information Services**.
   Note that ticking the checkbox will only enable the components required to publish a web site. Expand the folder to select other needed features.
5. Select **OK**.

With this the IIS can be used on a Windows machine. Take the following steps to enable the hosting of the workload images. The steps below are not carried out in detail as they are only serving as a guideline.

1. Remove all files from the document root. The default location of the document root is `C:\inetpub\wwwroot`.
2. Copy the TAR files of the workloads that have been exported from the Management System to the document root.
3. Enable directory listing.
4. Test the web server by opening http://localhost in a web browser.
5. Test if a computer can be reached from another computer in the network.

NOTE

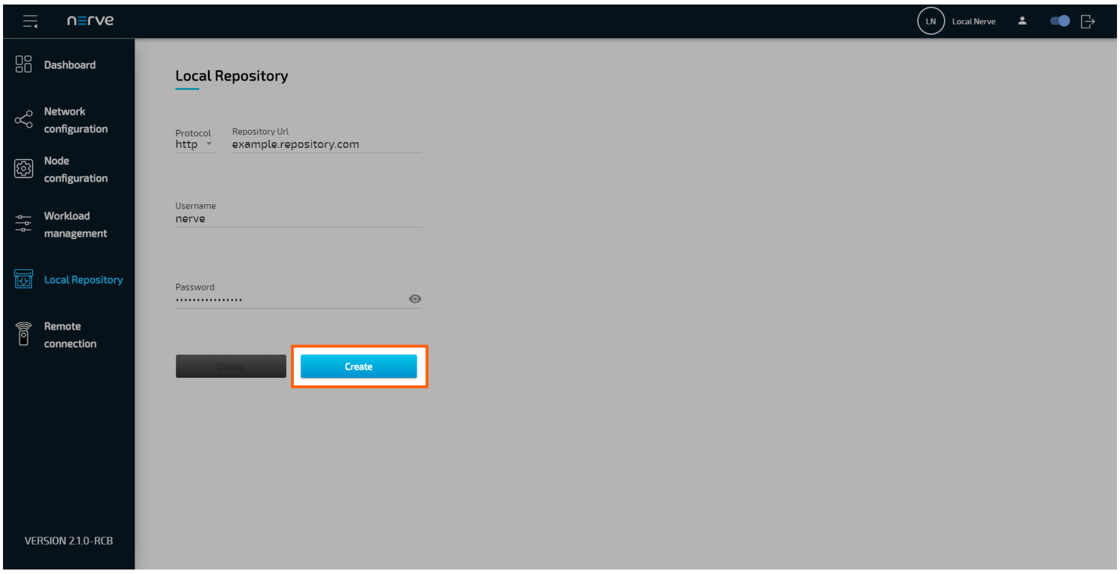
Note that all served files can be accessed from all computers connected to the same network.

For more information, refer to the IIS documentation. Follow the instructions below to enable a local repository for a node:

1. Select **Local Repository** in the navigation on the left.
2. Enter the following information:

Protocol	Select the protocol for communication with the local repository: **http** or **https**.
Repository URL	Enter the URL through which the web server hosting the workload images can be accessed.
Username	If login credentials have been defined for the web server, enter the username for accessing the web server hosting the workload images.
Password	If login credentials have been defined for the web server, enter the password for accessing the web server hosting the workload images.

3. Select **Create** to finish the setup.



Local Repository

Protocol: http

Repository Url: example.repository.com

Username: nerve

Password: ....

If a local repository is defined in the Local UI, the Management System will look for the workload images in the local repository first when a workload is deployed to this node. If the workload image is not present in the local repository, the workload repository in the Management System will be used instead. To revert back to using the repository of the Management System, select **Local Repository** in the navigation on the left and select **Delete**.

Managing remote connections

The behavior of nodes for remote connections is defined when a remote connection is configured in the Management System. Refer to Remote connections for more information.

Settings	Tick the checkbox next to **Always allow remote connections** to automatically approve all remote connection requests. This setting is only valid when **Local acknowledgment** is set to **Yes** in the remote connection configuration in the Management System.

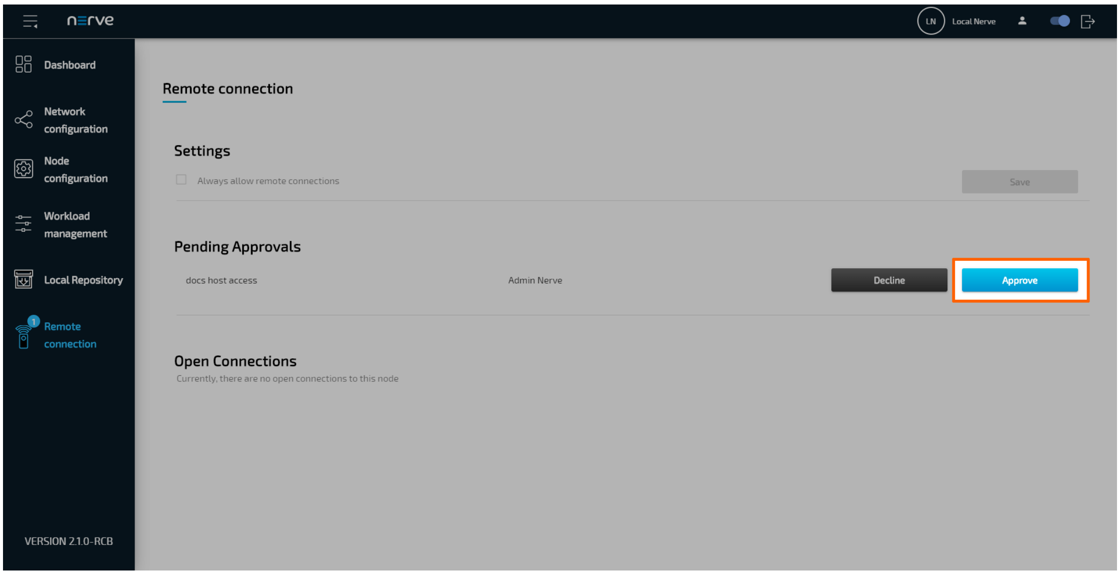
Pending Approvals	Incoming connection requests are displayed here. Select **Approve** to accept the remote connection. Clicking **Cancel** denies the incoming request. This setting is only valid when **Local acknowledgment** is set to **Yes** in the remote connection configuration in the Management System.
Open Connections	Connection requests that have been approved and are currently open are listed here. To the left, the name of the remote connection is shown. This name is also shown under **Remotes** in the Management System. In the middle, the user that established the remote connection is displayed. To the right, there is a button to terminate the remote connection.

Approving a remote connection

Local approval for remote connections can be configured in the Management System. Set **Local acknowledgment** to **Yes** when configuring a remote connection. When set to **Yes**, every remote connection has to be approved in the Local UI before it can be established. If approval for an incoming remote connection is pending, a notification bubble appears next to **Remote connections** in the navigation on the left.

1. Select **Remote connections** in the navigation on the left.
2. Search for incoming remote connections under **Pending Approval**.

3. Click **Approve** on the right for the remote connection that shall be established.



Remote connection

Settings

☐ Always allow remote connections

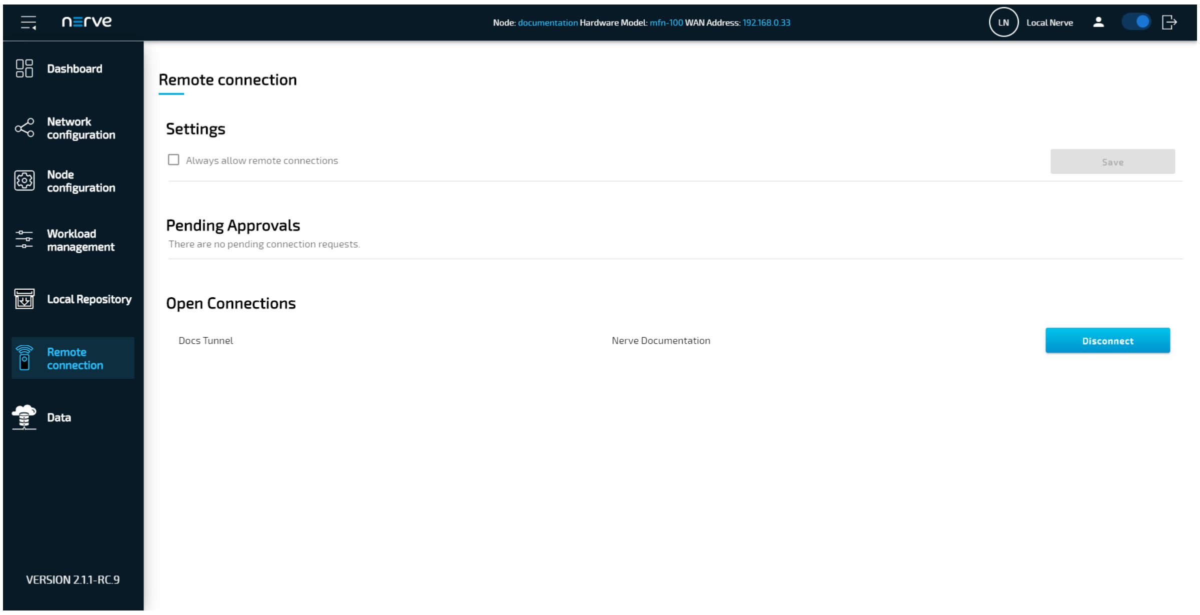
Pending Approvals

docs host access — Admin Nerve

Open Connections

Currently, there are no open connections to this node

Once approved, the open connection will be displayed under **Open Connections** below. Shown are the name of the remote connection and the user that is currently using the remote connection. If the same remote connection is used by multiple users, multiple entries of the same remote connection are shown with different users.



Node: documentation Hardware Model: mfn-100 WAN Address: 192.168.0.33

Remote connection

Settings

□ Always allow remote connections

Pending Approvals

There are no pending connection requests.

Open Connections

Docs Tunnel Nerve Documentation

Nerve Management System

Nerve Management System

The Nerve Management System is a web-based service that permits management of Nerve Blue nodes that are registered. It can be used to:

- Monitor nodes
- Deploy and control workloads on a node
- Manage workloads

NOTE

Google Chrome or Firefox Version 63 or later are recommended for the usage of the Management System.

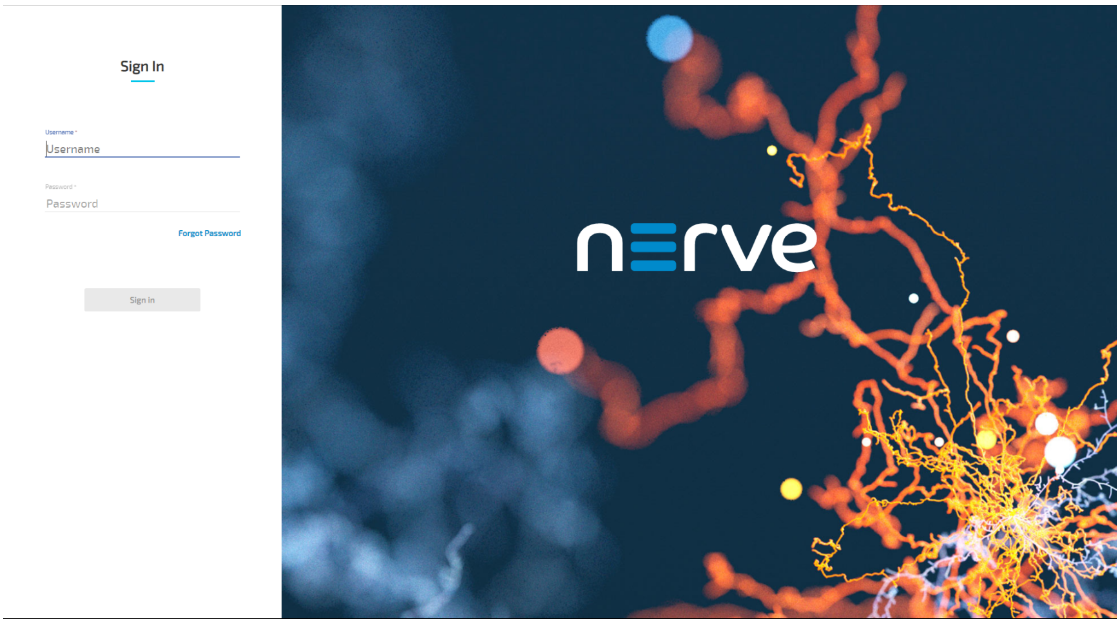
Logging in to the Management System

The Management System is either hosted by TTTech Industrial or installed on premise. The URL of the Management System changes accordingly. Find the URL of the Management System in the customer profile if it is hosted by TTTech Industrial.

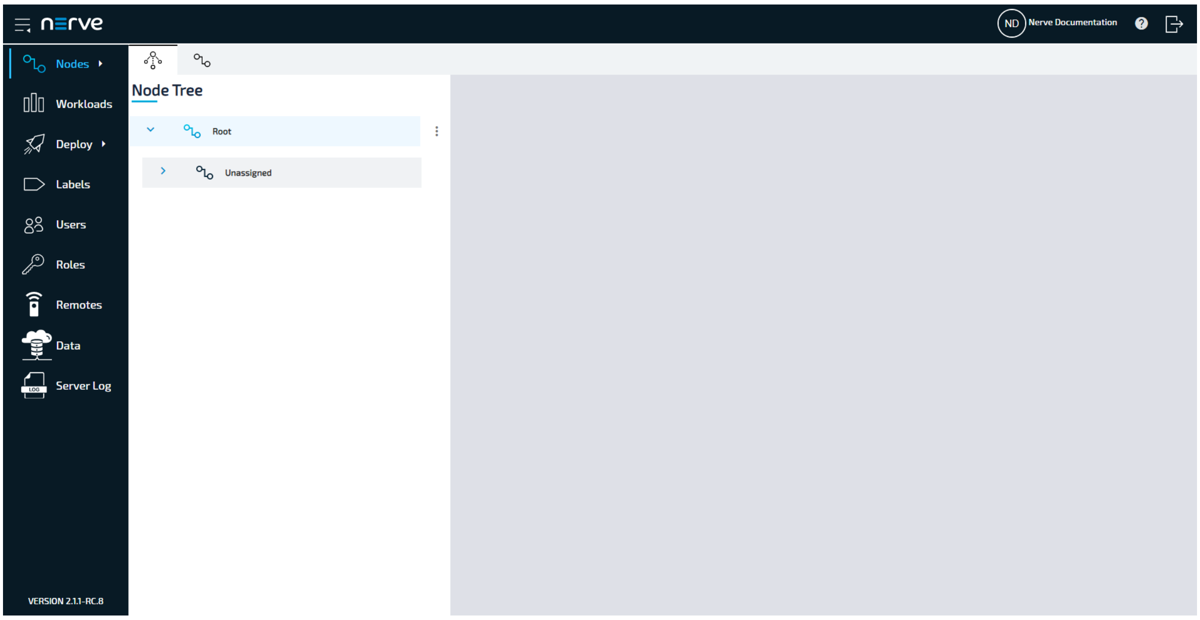
1. Go to the URL of the Management System.

2. Log in with the credentials for the Management System.

> **NOTE**
>
> The initial login credentials of the Management System can be found in the customer profile. If a customer profile has not been part of the delivery, contact a sales representative or TTTech Industrial customer support at support@tttech-industrial.com.

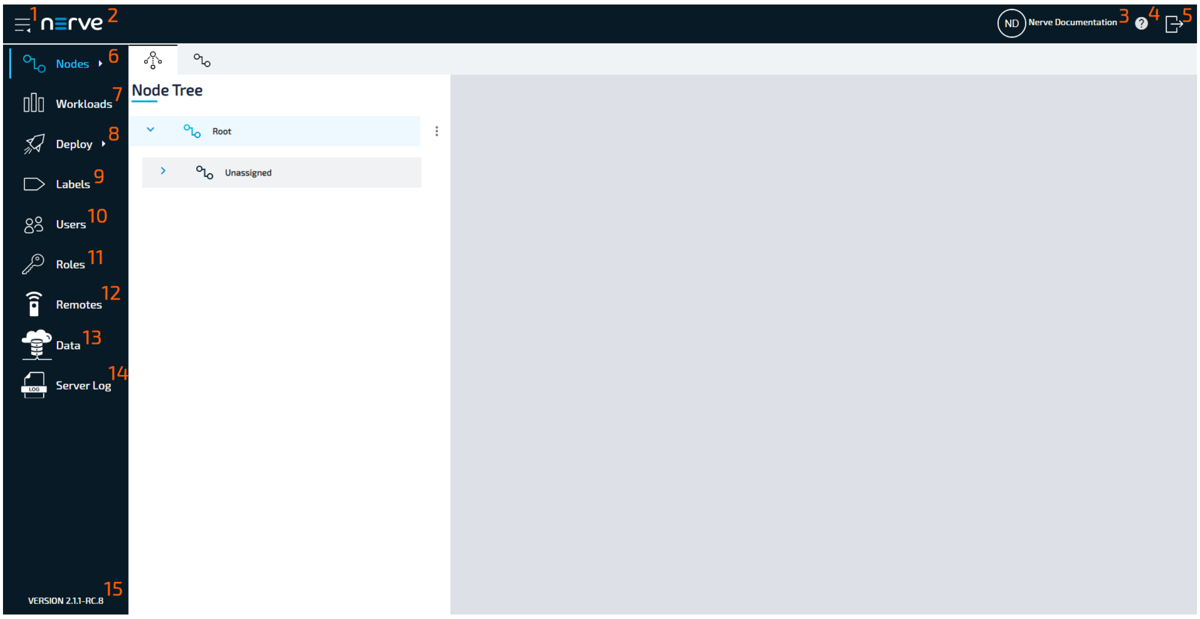


The node tree is the landing page of the Management System.



Menu structure of the Management System

As a landing page the Management System shows the node tree. Navigate the site by using the menu bar on the left side.



Item	Description
Burger Menu (1)	Clicking here will expand and collapse the menu bar on the left side. The expanded version of the menu adds names to the menu icons.
Nerve logo (2)	Return to the node tree by clicking the Nerve logo.
User display (3)	Access the user details of the active user from here.
Documentation link (4)	Click the question mark to open the Nerve Blue documentation in a new browser tab.
Log out button (5)	Clicking this icon will log out the active user from the Management System.
Nodes (6)	The nodes menu has two sub-menus in the navigation on the left and two tabs in the default view. **Tabs** • **Node Tree Tab** This is the default view of the Management System. It displays all registered nodes in a node tree. Add, delete and move tree elements freely here. It mainly serves an organizational purpose and does not impact the functionality of the nodes. Nodes and workloads can be managed in the node details view, which is reached through the node tree. • **Node List Tab** Selecting the nodes tab displays a list of all available nodes that have been registered in the Management System. Add and remove nodes, as well as edit their details here. **Sub menus** ◦ **Updates** Available updates for nodes can be found here. ◦ **Update Log** Past node updates can be viewed here. When an update is performed, this view also shows the current progress of the update.

Item	Description
Workloads (7)	All workloads that have been provisioned in the Management System are listed here. Workloads can be added, deleted, disabled and edited through this menu. New versions of workloads can also be added here.
Deploy (8)	Workloads that have been provisioned previously can be deployed using this menu. However, there are two sub menus available here: • **Log** Find a list of all workloads that have been deployed or are currently being deployed. Also, view details of all deployments and delete log entries from the list. • **Dry Run** A dry run is a simulation of a workload deployment. It allows to test out if the deployment of a workload could be successful. However, note that a successful dry run is not a guarantee for successful deployment.
Labels (9)	This is a list of all labels that have been defined in the Management System. Add, delete, edit and merge labels here.
Users (10)	This is the user management menu. It lists all registered users and allows to edit profiles and add new users.
Roles (11)	Manage user roles and and permissions here.
Remotes (12)	Currently active remote connections are displayed here. Refer to Remote Connections for more information.
Data (13)	Access the instance of the Nerve Data Services in the Management System here. This feature is disabled by default and must first be activated by an admin user. Refer to Nerve Data Services for more information.
Server Logs (14)	Look at internal server logs here. These internal server logs are aimed at Nerve service technicians in case of error and failure. Data is stored with Elasticsearch and the logs are visualized with the Kibana application.
Management System version (15)	This is the current version of the Management System. Select the download symbol to view available updates for the Management System.

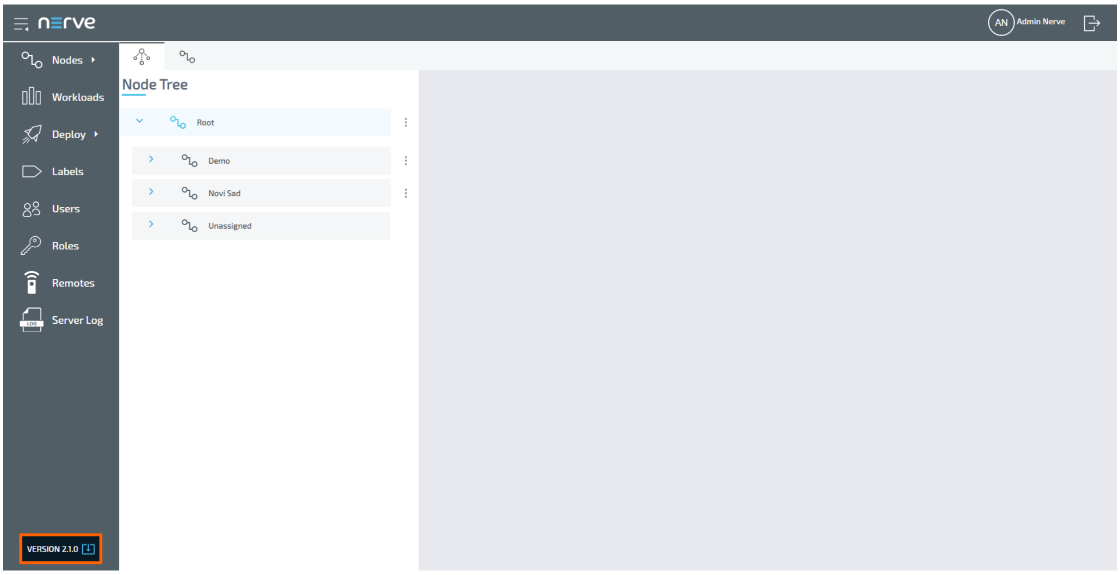
NOTE

The available features of the Management System depend on the user role. Refer to Roles and Permissions for more information.

Updating the Management System

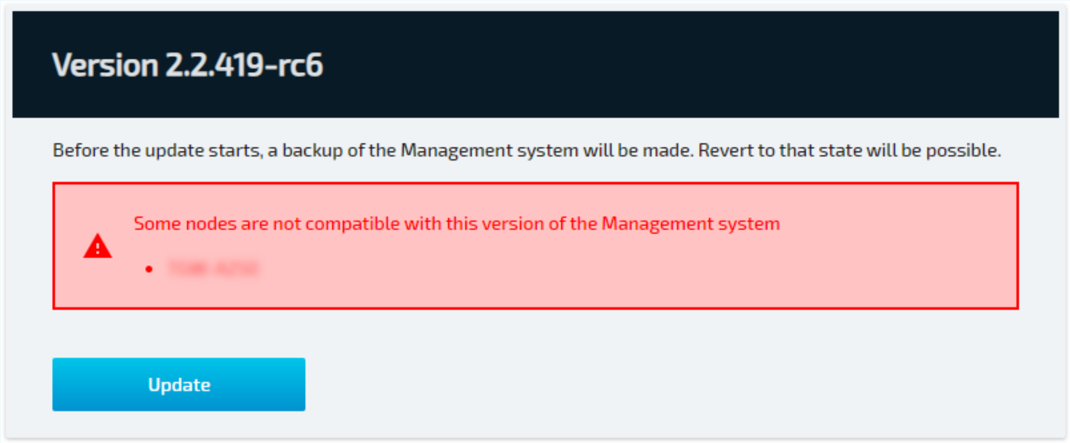
Select the download icon next to the version number in the lower-left corner of the Management System to see if an update to the Management System is available. A backup of the current Management System is made when an update is performed and all workloads are stopped. Reverting to the previous version is possible.

1. Select the download icon in the lower-left corner next to the version number.



2. Select the update version of the Management System.

> **NOTE**
>
> Registered nodes that will be incompatible after the update are shown on the right. These nodes cannot be used with the updated Management System if they are not updated to a newer node version. Refer to Updating a Node for more information.



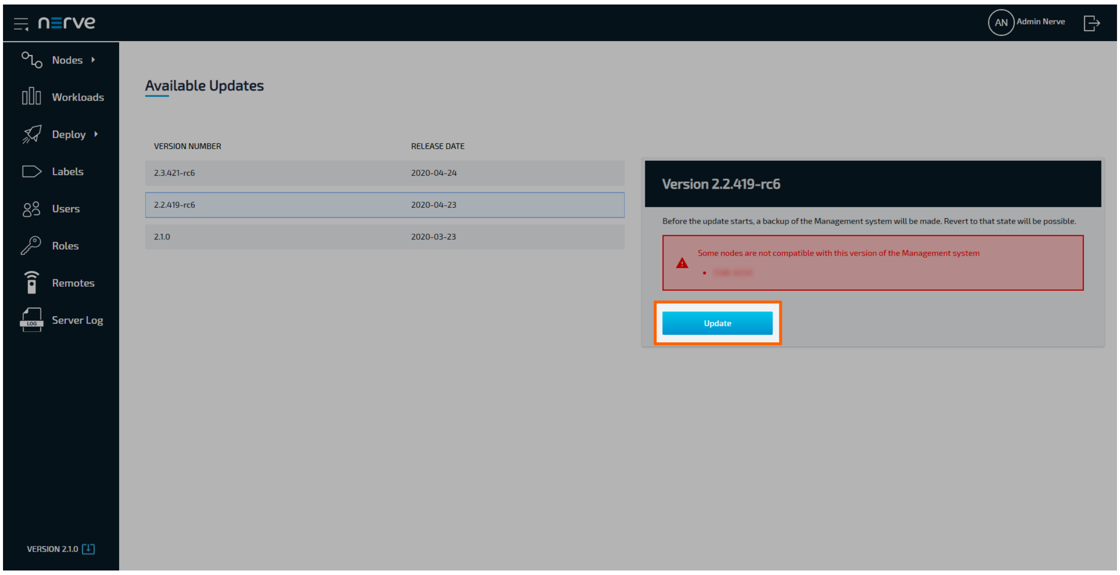
Version 2.2.419-rc6

Before the update starts, a backup of the Management system will be made. Revert to that state will be possible.

Some nodes are not compatible with this version of the Management system

- [redacted]

3. Click **Update** on the right.



Available Updates

VERSION NUMBER	RELEASE DATE
2.3.421-rc6	2020-04-24
2.2.419-rc6	2020-04-23
2.1.0	2020-03-23

Version 2.2.419-rc6

Before the update starts, a backup of the Management system will be made. Revert to that state will be possible.

Some nodes are not compatible with this version of the Management system

- [redacted]

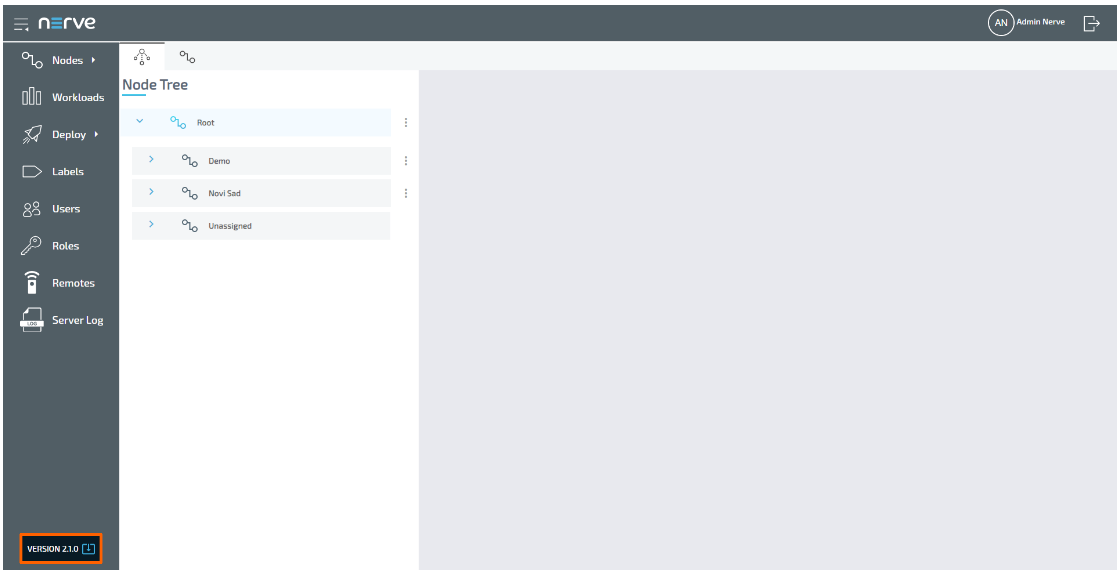
4. Select **YES** in the pop-up window.

Note that updates of the Management System have to be performed in order. Skipping a version is not permitted.

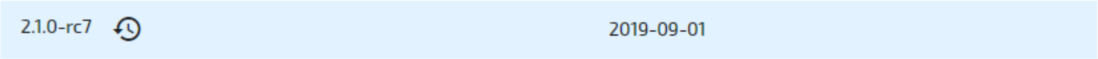
Reverting the Management System to the previous version

The previous version of the Management System will be marked with an icon in the list of available updates. Note that any new data created after the update will be lost when reverting to the previous version of the Management System. Downgrading the Management System to an older version is not possible.

1. Select the download icon in the lower-left corner next to the version number.



2. Select the previous version of the Management System that is marked with an icon.

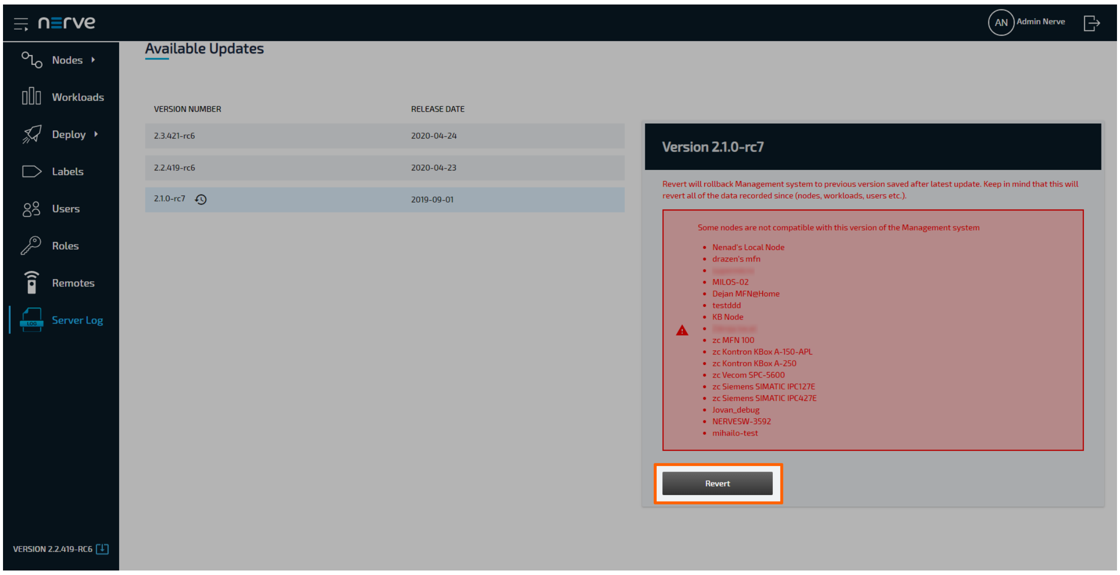


2.1.0-rc7	2019-09-01

NOTE

Registered nodes that will be incompatible after the update are shown on the right. These nodes cannot be used with the reverted Management System.

3. Click **Revert** on the right.



Available Updates

VERSION NUMBER	RELEASE DATE
2.3.421-rc6	2020-04-24
2.2.419-rc6	2020-04-23
2.1.0-rc7	2019-09-01

Version 2.1.0-rc7

Revert will rollback Management system to previous version saved after latest update. Keep in mind that this will revert all of the data recorded since (nodes, workloads, users etc.).

Some nodes are not compatible with this version of the Management system

- Nenad's Local Node
- drazen's mfn
- [redacted]
- MILOS-02
- Dejan MFN@Home
- testddd
- KB Node
- [redacted]
- zc MFN 100
- zc Kontron KBox A-150-APL
- zc Kontron KBox A-250
- zc Vecom SPC-5600
- zc Siemens SIMATIC IPC127E
- zc Siemens SIMATIC IPC427E
- Jovan_debug
- NERVESW-3592
- mihailo-test

4. Select **YES** in the pop-up window.

Nodes

The nodes menu has two sub-menus in the navigation on the left and two tabs in the default view.

The node tree tab is the default view of the Management System. It displays all registered nodes in a node tree. It mainly serves an organizational purpose and does not impact the functionality of the nodes. Nodes and workloads can be managed in the node details view, which is reached through the node tree.

Selecting the node list tab displays a list of all available nodes that have been registered in the Management System. Add, remove, and edit nodes here.

Available updates for nodes can be found in the **Updates** sub-menu. Performed node updates can be viewed in the **Update Log**. When an update is performed, the update log also shows the current progress of the update.

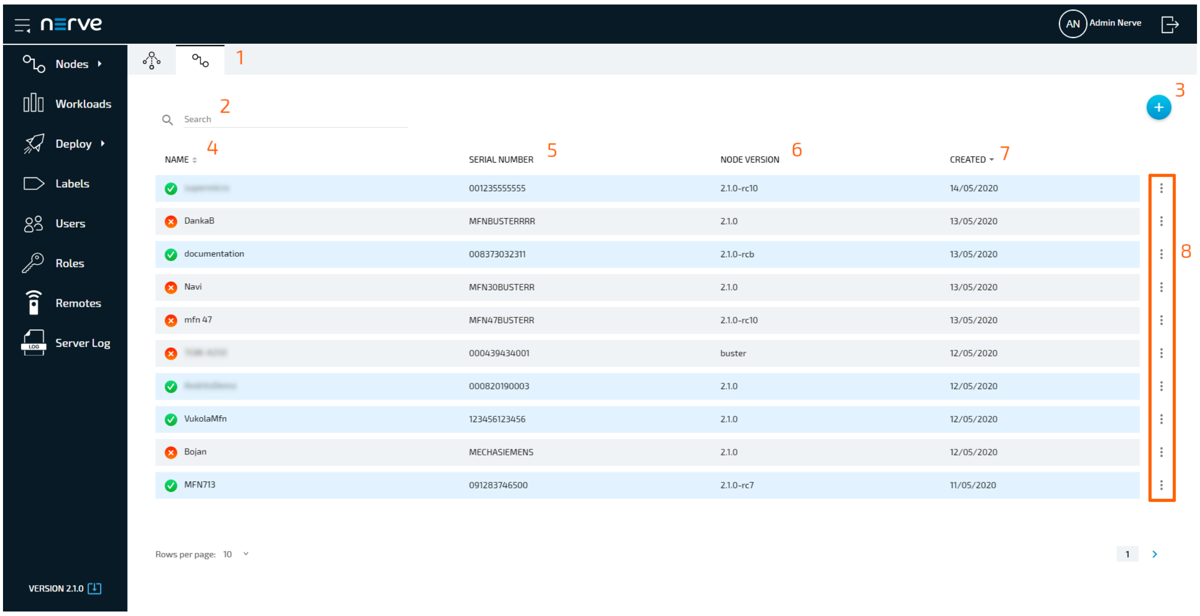
Node list

The topics covered in this chapter are mainly means of organization and have no impact on the functionality of the nodes and their workloads. Select **Nodes** in the navigation on the left. Then select the nodes tab

on the right to display the list of registered nodes.

NOTE

Operational functions of the nodes are located in the node details view in the node tree.



NAME	SERIAL NUMBER	NODE VERSION	CREATED
✓ [redacted]	001235555555	2.1.0-rc10	14/05/2020
✗ DankaB	MFNBUSTERRRR	2.1.0	13/05/2020
✓ documentation	008373032311	2.1.0-rcb	13/05/2020
✗ Navi	MFN30BUSTERR	2.1.0	13/05/2020
✗ mfn 47	MFN47BUSTERR	2.1.0-rc10	13/05/2020
✗ [redacted]	000439434001	buster	12/05/2020
✓ [redacted]	000820190003	2.1.0	12/05/2020
✓ VukolaMfn	123456123456	2.1.0	12/05/2020
✗ Bojan	MECHASIEMENS	2.1.0	12/05/2020
✓ MFN713	091283746500	2.1.0-rc7	11/05/2020

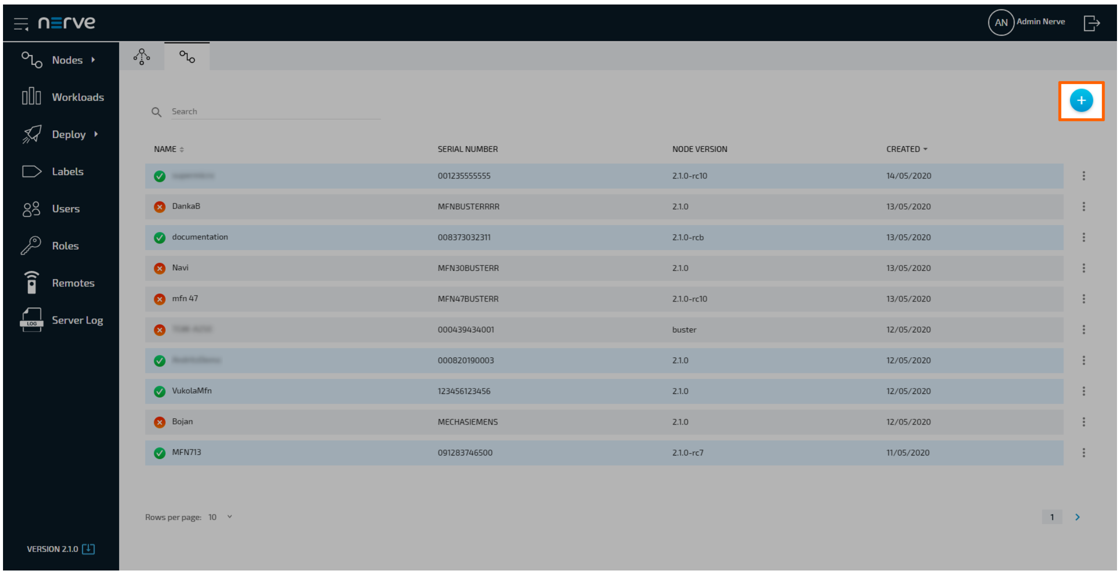
Item	Description
Tab selection (1)	Switch between the node tree and the node list by selecting the appropriate tab.
Search bar (2)	Use the search bar to filter nodes by name.
Add new node (3)	Click here to manually add a new node.
NAME (4)	This is the name of the node. If a node is online or offline can be seen to the left of the name. A green check mark indicates an online node while a red cross shows an offline node. The sorting of the list can also be switched from alphabetical to reverse alphabetical by clicking **NAME**, as well as being sorted by creation date by clicking **CREATED**.
SERIAL NUMBER (5)	This is the serial number of the node that was defined during node configuration.
NODE VERSION (6)	This is the version of the node reflecting the version of the Nerve Blue product.
CREATED (7)	This is the date the node was registered in the Management System in the format DD/MM/YYYY. The list can be sorted by creation date when clicking **CREATED**.
Ellipsis menu (8)	Clicking here opens an overlay that allows deleting nodes.

Adding a node

Nodes can only be added to the Management System if they have been configured in the Local UI, as the serial number and secure ID of the node are required. Refer to Node configuration before continuing.

1. Select **Nodes** in the menu on the left side.
2. Select the nodes tab

   on the right to display the list of registered nodes.
3. Select the plus symbol (**Add new node**) in the upper-right corner.

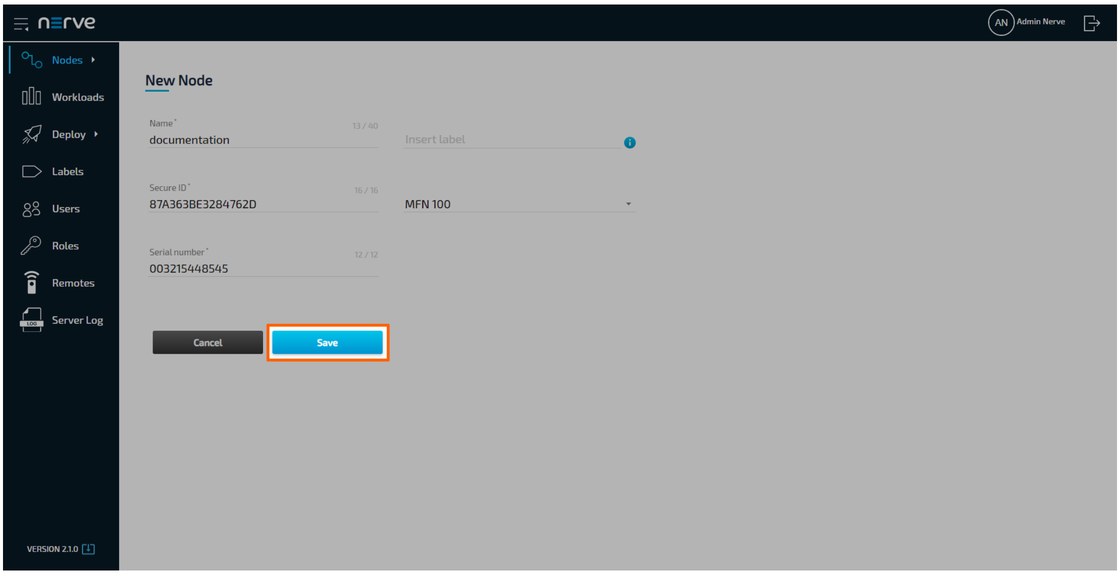


NAME	SERIAL NUMBER	NODE VERSION	CREATED
[redacted]	001235555555	2.1.0-rc10	14/05/2020
DankaB	MFNBUSTERRRR	2.1.0	13/05/2020
documentation	008373032311	2.1.0-rcb	13/05/2020
Navi	MFN30BUSTERR	2.1.0	13/05/2020
mfn 47	MFN47BUSTERR	2.1.0-rc10	13/05/2020
[redacted]	000439434001	buster	12/05/2020
[redacted]	000820190003	2.1.0	12/05/2020
VukolaMfn	123456123456	2.1.0	12/05/2020
Bojan	MECHASIEMENS	2.1.0	12/05/2020
MFN713	091283746500	2.1.0-rc7	11/05/2020

4. Enter the following information:

Item	Description
Name	Enter the name for the node that will make it easy to identify.
Secure ID	Enter the secure ID that is generated when the node is configured in the Local UI. Refer to Node configuration for more information.
Serial no.	Enter the serial number of the Nerve Device that was defined during node configuration.
Insert Label	This field is optional. Add labels to the node for easier identification and workload deployment. Note that labels have to be created first before they can be assigned to nodes. Refer to Labels for more information.
Nerve Device name	Select the Nerve Device, on which the node is hosted from the drop-down menu. The available selections are: ◦ **mfn-100** ◦ **kbox a-150-apl** ◦ **kbox a-250** ◦ **spc-5600-i5-8500** ◦ **simatic ipc127e** ◦ **simatic ipc427e**. The name is set to **mfn-100** by default. A picture of the selected Nerve Device will be displayed in the node details. Refer to the device guide for more information on qualified Nerve Devices.

5. Click **Save** to save the changes.



New Node

Name* documentation 13 / 40

Insert label

Secure ID* 87A363BE3284762D 16 / 16

MFN 100

Serial number* 003215448545 12 / 12

Cancel Save

The node now appears in the node list and in **Root > Unassigned** in the node tree and can be worked with. Refer to the user guide for information on how to continue.

Editing the details of nodes

After registering a node, edit its details in the **Nodes** menu. Use the search bar at the top to search for nodes if a large number of nodes is registered.

1. Select **Nodes** in the left-hand menu.
2. Select the nodes tab

   on the right to display the list of registered nodes.
3. Click a node from the list.

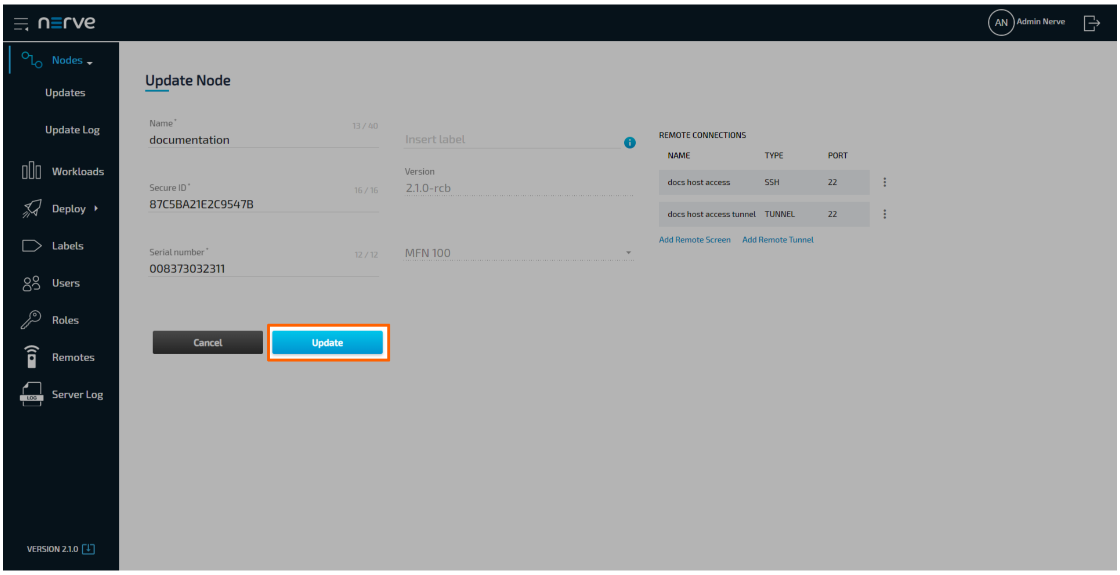
4. Edit the details of the node:

Item	Description
Name	Enter the name of the node.
Secure ID	This is the ID generated during the node registration process. Do not edit this ID without a valid reason.
Serial number	This is the serial number of the node. Do not edit the serial number without a valid reason.
Insert label	This field is optional. Add labels to the node for easier identification and workload deployment. Refer to Labels for more information.
Version	This is the node version. This field cannot be edited.
Device type	This is the device the node is hosted on. This field cannot be edited.
REMOTE CONNECTIONS	This is a list of remote connections to the node stating the **NAME**, **TYPE**, and **PORT** of available remote connections. Add or delete remote connections here. Refer to Remote Connections for more information.

NOTE

Changing the **Secure ID** or **Serial no.** of a node will break the connection between the node and the Management System.

5. Click **Update** to save the changes.



Update Node

Name* 13 / 40
documentation

Insert label

Secure ID* 16 / 16
87C5BA21E2C9547B

Version
2.1.0-rcb

Serial number* 12 / 12
008373032311

MFN 100

REMOTE CONNECTIONS

NAME	TYPE	PORT
docs host access	SSH	22
docs host access tunnel	TUNNEL	22

Updating a node to a new version

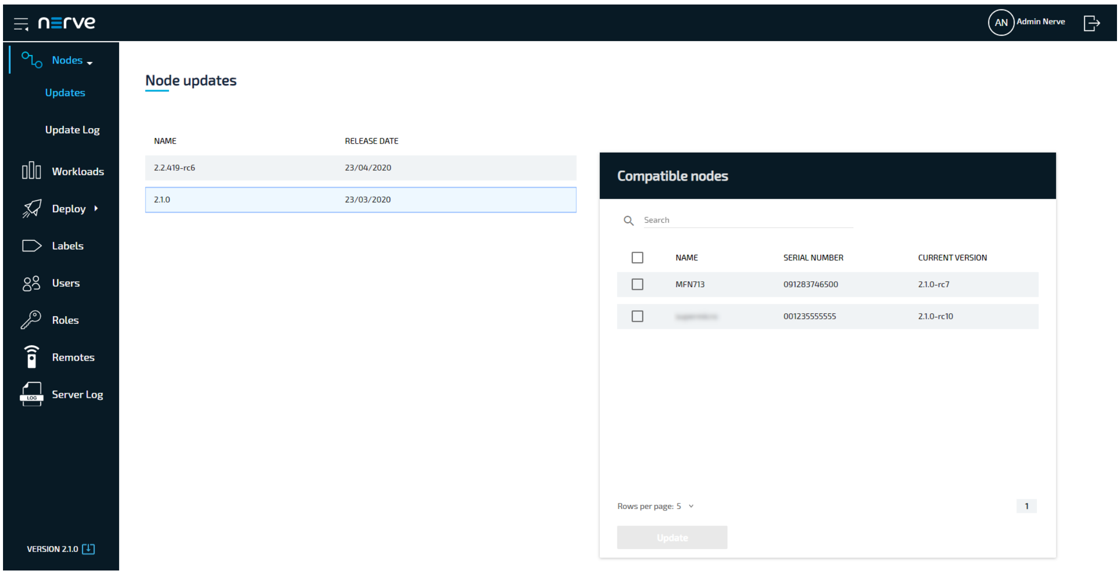
Nodes are updated through the Management System where currently available updates are listed. Note that nodes can only be updated in order, meaning that node versions cannot be skipped. When an update is started, a snapshot of the user data is made. Reverting to the previous version is possible.

NOTE

Note that the update process described here is valid for nodes from version 2.1.0 and later. Nodes below version 2.1.0 need to be updated manually. Contact TTTech Industrial customer support at support@tttech-industrial.com for more information.

1. Expand **Nodes > Updates** in the navigation on the left.

2. Select a node version from the list. Available nodes for the update to the selected version will appear on the right.



Node updates

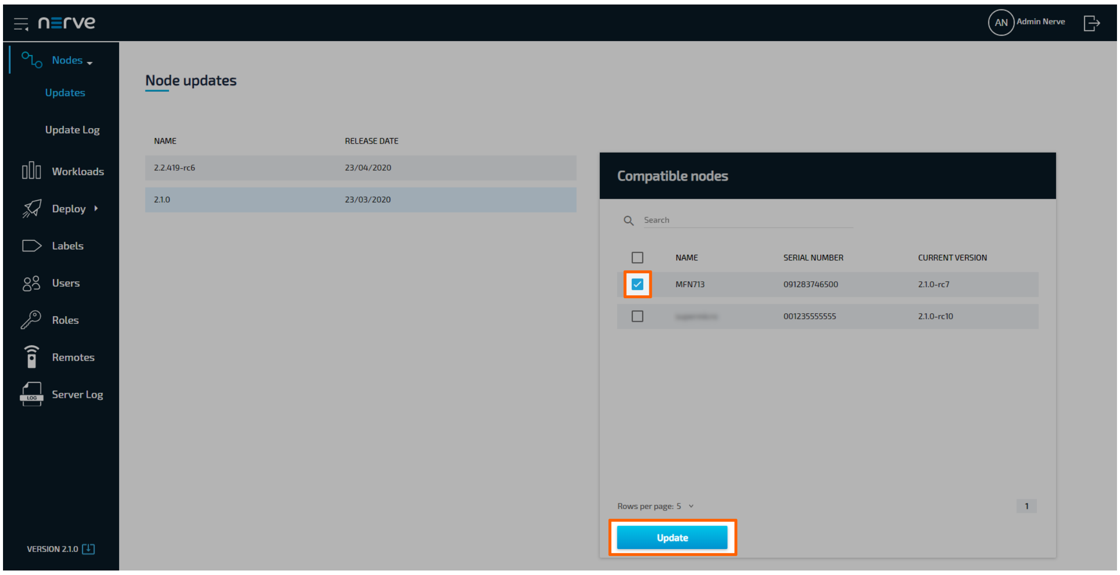
NAME	RELEASE DATE
2.2.419-rc6	23/04/2020
2.1.0	23/03/2020

Compatible nodes

☐	NAME	SERIAL NUMBER	CURRENT VERSION
☐	MFN713	091283746500	2.1.0-rc7
☐	[redacted]	001235555555	2.1.0-rc10

3. Tick the checkboxes next to the nodes that will be updated.

4. Select **Update**.



Node updates

NAME	RELEASE DATE
2.2.419-rc6	23/04/2020
2.1.0	23/03/2020

Compatible nodes

☐	NAME	SERIAL NUMBER	CURRENT VERSION
☑	MFN713	091283746500	2.1.0-rc7
☐	[redacted]	001235555555	2.1.0-rc10

5. Select **YES** in the pop-up window.

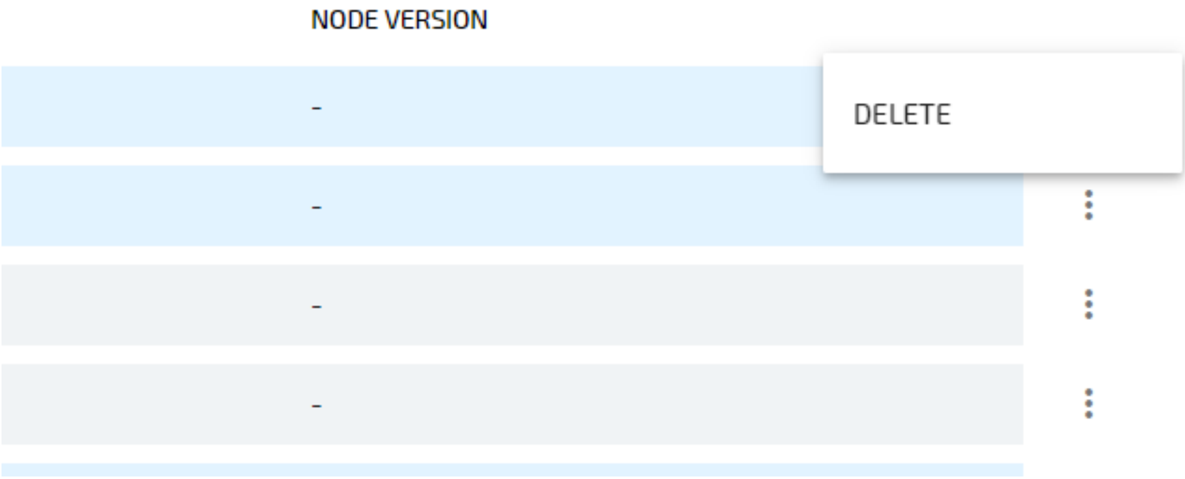
The Management System will display the log screen where the update is shown as in progress. The progress bar here shows the progress of the entire update campaign if more than one node has been selected before. Click the progress bar to see the progress of the update to each node.

NOTE

The update may fail with an **unexpected end of file** error. In that case retry the update and the update will be eventually successful after a certain number of tries.

Removing a node

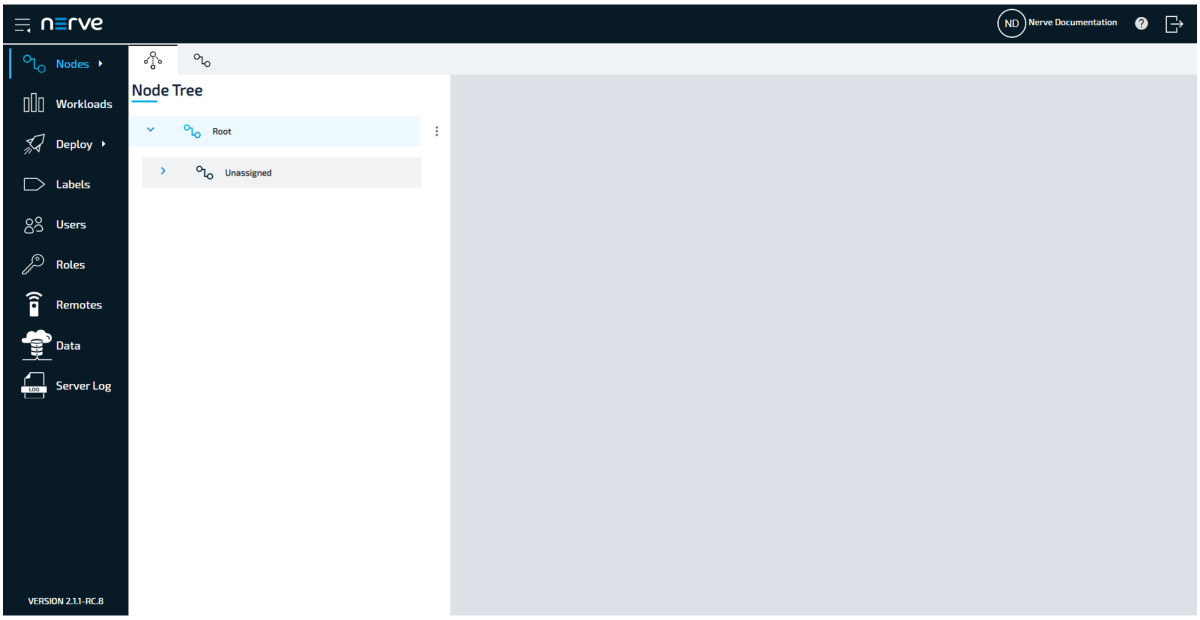
Note that generally there is no need in removing a node. Only remove a node in case of technical difficulties or by request of customer support. To remove a node, select **DELETE** in the ellipsis menu on the right side of a node in the node list.



Node tree

The node tree is the landing page of the Management System. It presents an overview of all nodes that are connected to the Management System, embedded into tree view elements. Being mainly a means of organization, it has no impact on the functionality of the nodes and their workloads. Select **Nodes** in the navigation on the left. Then select the node tree tab

on the right.

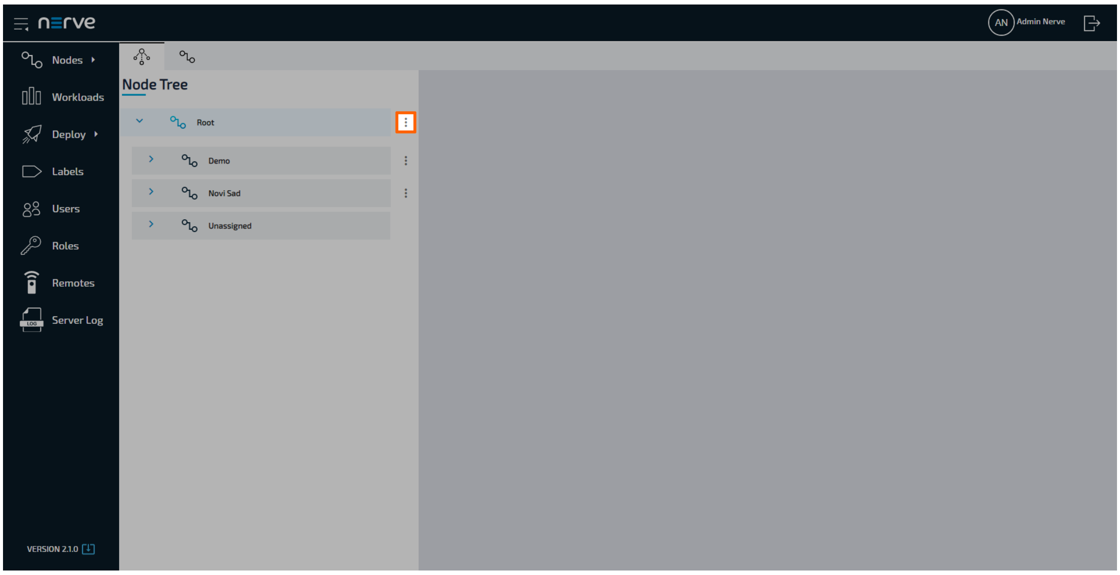


There is only one element under the root after the initial setup: **Root > Unassigned**. All nodes that are registered in the Management System are placed in the **Unassigned** element by default. From there they can be moved to new elements that have to be created first.

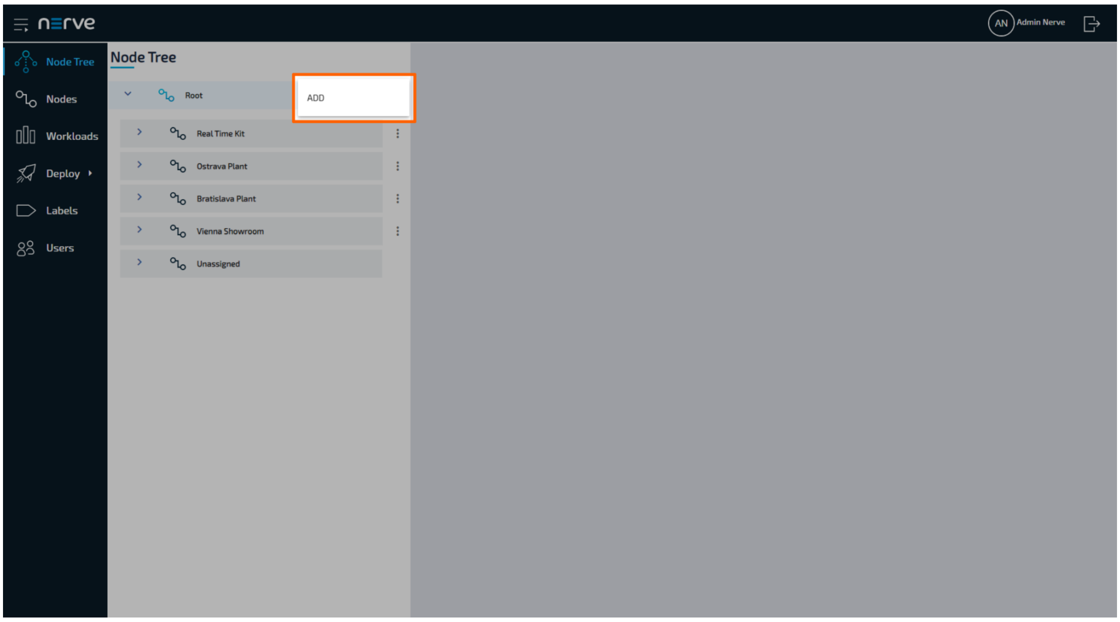
Creating a new element in the node tree

Before moving a node out of the **Unassigned** element, create a new element in the node tree. Elements in the node tree exclusively serve an organizational purpose.

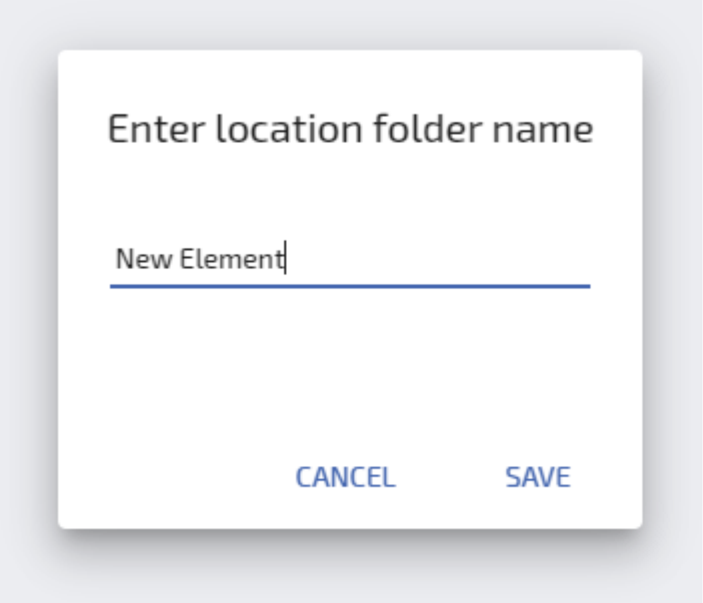
1. Select **Nodes** in the navigation on the left.
2. Select the node tree tab on the right.
3. Select the ellipsis menu to the right of **Root** in the node tree.



4. Click **Add** in the overlay that popped up.



5. Enter the name of the new element under **Location name**.



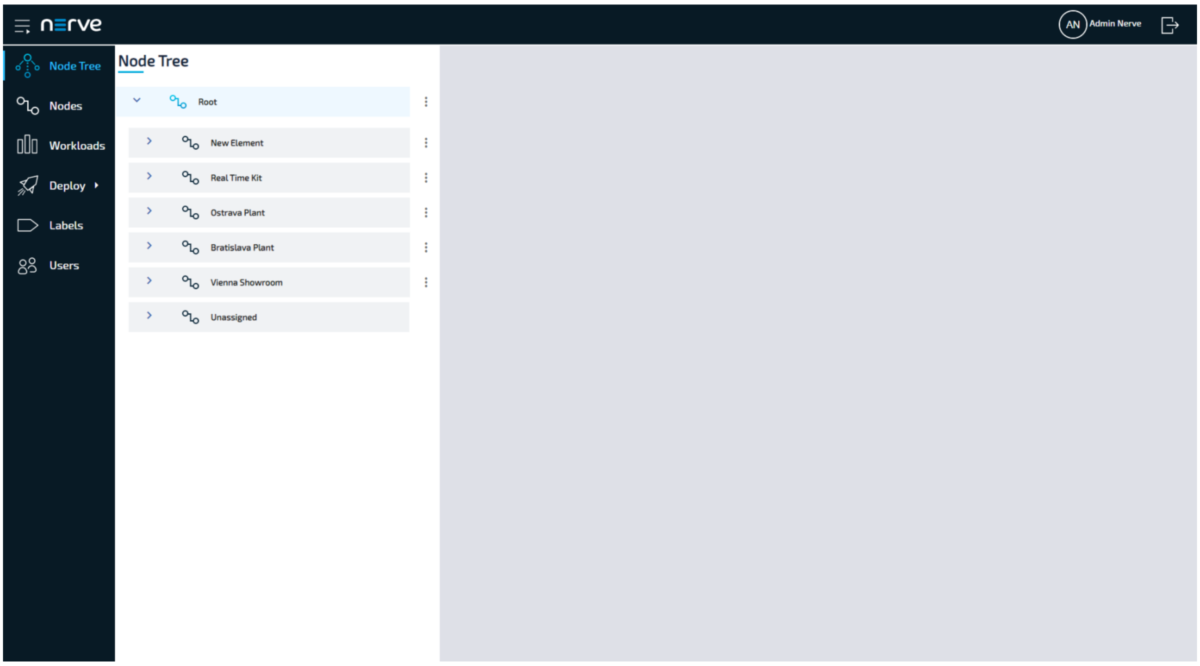
Enter location folder name

New Element

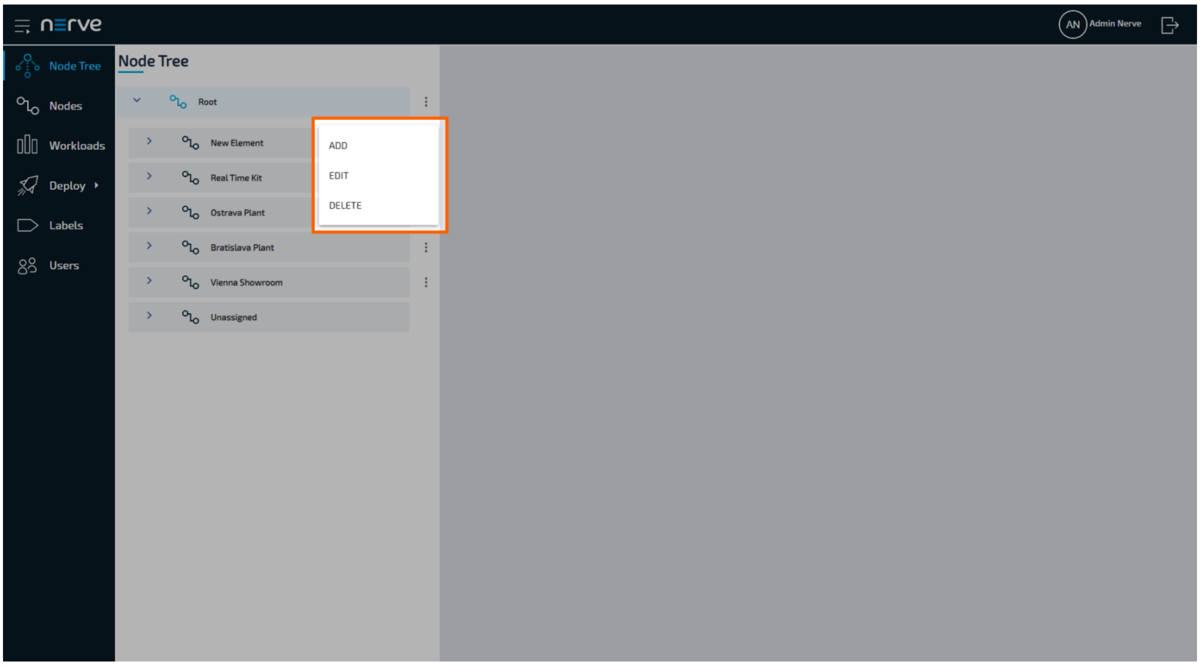
6. Click **Save**.
7. Select **APPLY CHANGES (n)** in the upper-right corner of the node tree.

> **NOTE**
>
> `(n)` is a placeholder for the number of changes made to the node tree. If three changes have been performed, `(3)` will be displayed in the button above the node tree.

The new element now appears under the **Root** element.



Create more elements and freely modify the node tree. To the right of every created element, there is an ellipsis menu that opens up an overlay. Add additional elements, edit the names of elements or delete the elements here.



NOTE

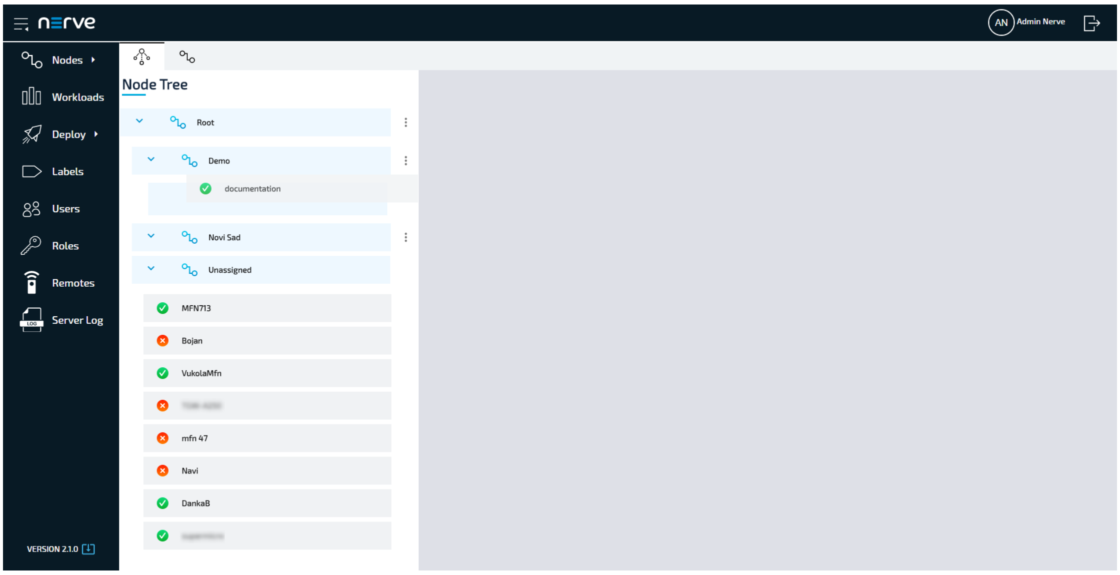
- The order of the tree elements can be modified easily. Drag and drop an element to its new position.
- When a tree element is deleted, all of the nodes inside the element will be moved to **Unassigned**.

Moving a node from one tree element to another

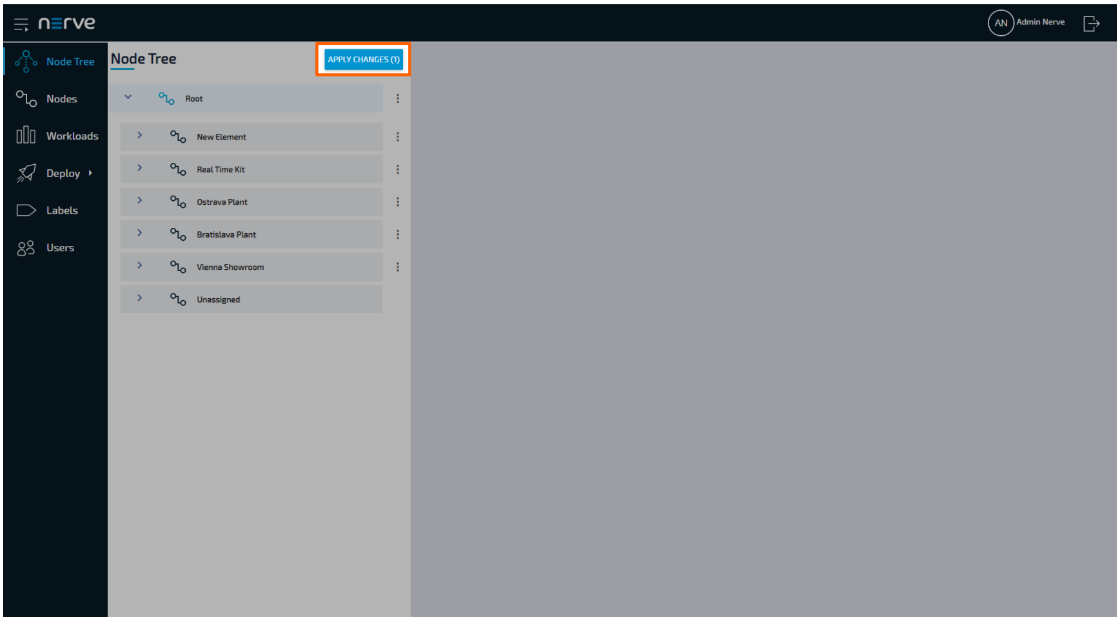
Moving nodes in the node tree is very straightforward and intuitive and possible by drag and drop. Make sure that a new tree element is created before attempting to move a node.

1. Select **Nodes** in the navigation on the left.
2. Select the node tree tab on the right.
3. Expand the tree element of the node that will be moved. The default element is **Root > Unassigned**.
4. Choose the node to move.

5. Drag and drop the node to the newly created element. Elements expand automatically once as they are hovered over.



6. Select **APPLY CHANGES (n)** in the upper-right corner of the node tree.



NOTE

`(n)` is a placeholder for the number of changes made to the node tree. If three changes have been performed, `(3)` will be displayed in the button above the node tree.

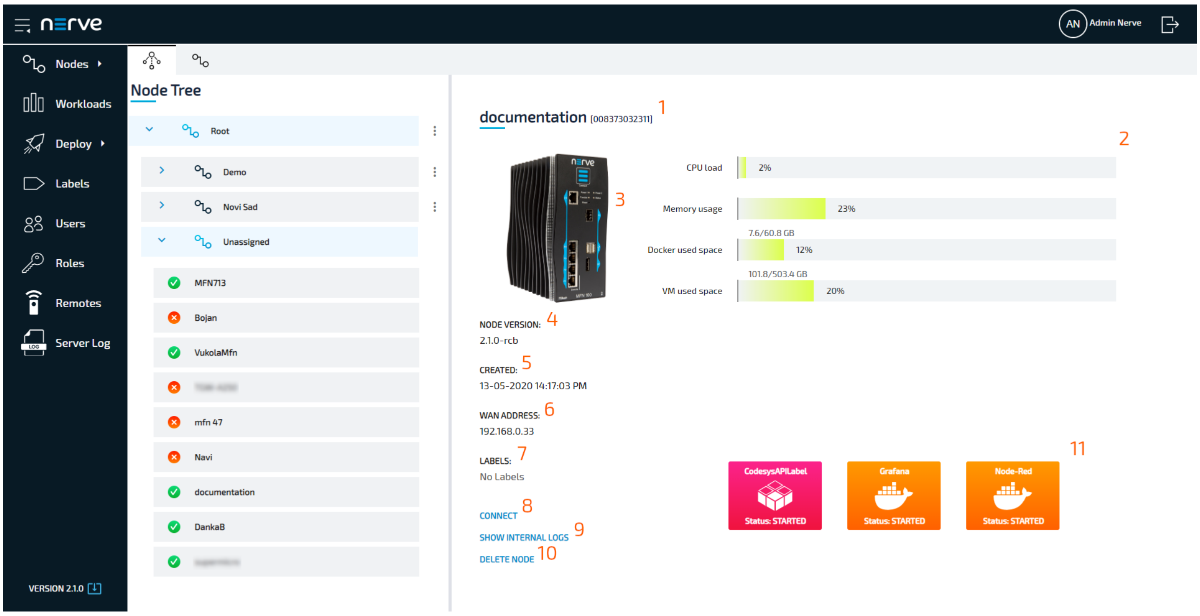
The node has now been moved to the new element.

NOTE

A node cannot be moved back manually to **Unassigned** once it has been moved to another element.

Managing nodes in the node tree

Once nodes are registered in the Management System, view their details next to the node tree. To view the details of a node, select the node name or symbol.



documentation [008373032311]

NODE VERSION:
2.1.0-rcb

CREATED:
13-05-2020 14:17:03 PM

WAN ADDRESS:
192.168.0.33

LABELS:
No Labels

Item	Description
Node name and serial (1)	Here the name and serial number of the node are shown. The serial number is next to the name in brackets.
System metrics (2)	The system metrics that are available in the Local UI dashboard are also visible here: • **CPU** The percentage here shows how much processing power is being used in total at the moment. This includes CPUs that have been assigned to VMs and Docker containers as well. • **Memory** Similar to CPU usage, the percentage of memory used in total at the moment is shown here. This includes memory that has been assigned to VMs or Docker containers. • **Docker free space** Two things are shown in this graph: The percentage shows how much of the available space for Docker containers is already used. The value shows the amount of space that is free. Docker containers have their dedicated virtual partition. • **LVM free space** Similar to Docker free space, two things are shown in this graph as well: The percentage shows how much of the available space for virtual machines is already used. The value shows the amount of free space. Note that deployed Virtual Machine workloads share a logical volume group with the Nerve Blue Base System. Therefore the percentages and values displayed in this graph are in relation to the total amount of space available of the logical volume group.
Device image (3)	An image of the hardware model is displayed here according to the device type that was selected when the node was added.

Item	Description
NODE VERSION (4)	The version of the node depending on the product version of Nerve Blue.
CREATED (5)	This is the date when the node was added to the Management System.
WAN ADDRESS (6)	This is the network address of the node that has been assigned by the DHCP server.
LABELS (7)	Labels that are assigned to this node are listed here. Labels can be set in the Management System. Refer to Labels for more information.
CONNECT (8)	Clicking here opens an overlay, through which a remote connection to the node can be established. Refer to Remote Connections for more information.
SHOW INTERNAL LOGS (9)	Clicking here will open a new window and show the system logs of the node. The internal node logs are aimed at Nerve service technicians in case of error and failure. Data is stored with Elasticsearch and the logs are visualized with the Kibana application. The amount of logs can be modified through the log level settings by Nerve service technicians. Contact customer support for more information.
DELETE NODE (10)	Clicking here removes the node from the Management System. The node needs to be registered again after it has been removed.
Workloads overview (11)	Find workloads that have been deployed to the Nerve Device displayed in tiles here. Selecting these tiles leads to a control area in which the workload can be controlled. If there are no tiles, no workloads have been deployed to the Nerve Device.

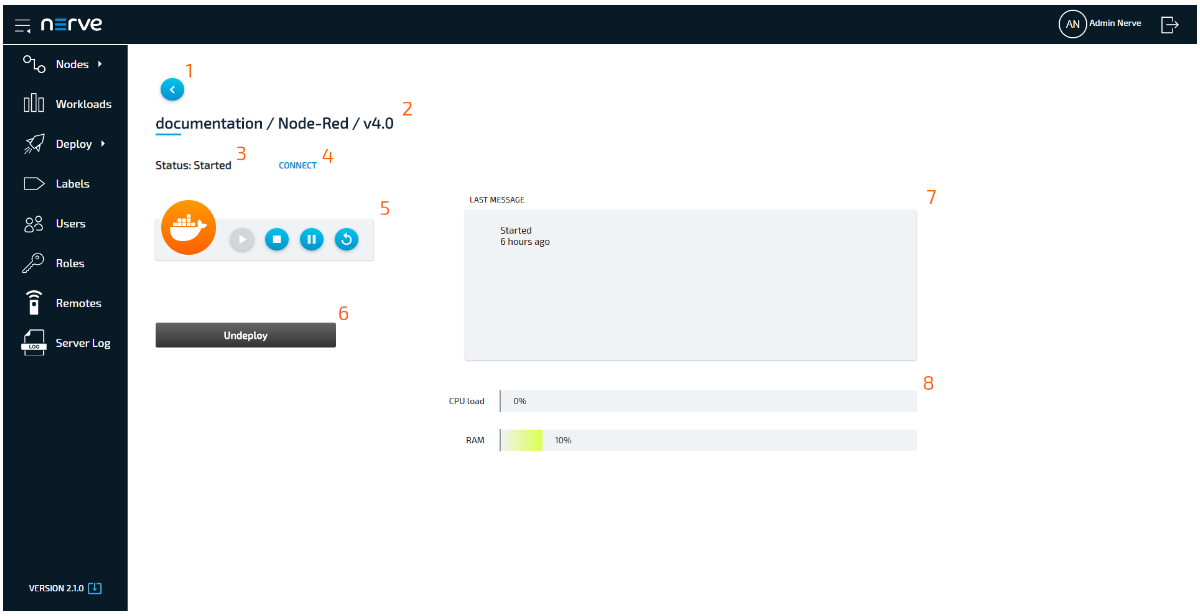
Workload control

All workloads that have been deployed to the node are displayed in tiles below the node details in the node tree. Clicking these tiles allows to control the respective workload.

NOTE

CODESYS workloads can only be controlled from the Local UI.

The workload control screen offers the same options for both Virtual Machine workloads and Docker workloads.



documentation / Node-Red / v4.0

Status: Started

Started
6 hours ago

CPU load 0%

RAM 10%

Item	Description
Back button (1)	Click here to return to the node tree.
Device and workload name (2)	The names of the device, the workload and the release name are displayed here as **<devicename> / <workloadname> / <releaseename>**. The name of the workload version is not displayed.
Workload status (3)	The current status of the workload is displayed here. The possible statuses are the following: • **Idle** This is the initial state of the workload before it is started. • **Creating** This is a transitional state of the workload when it is being created on the node. • **Starting** This is a transitional state when the workload is being started. • **Restarting** This is a transitional state when the workload is being restarted. • **Started** The workload is running and operating. • **Suspending** This is a transitional state when the workload is being suspended. • **Suspended** The workload has been paused. • **Resuming** This is a transitional state when the workload is being resumed from the suspended state. • **Stopping** This is a transitional state when the workload is being stopped. • **Stopped** The workload has been stopped. • **Removing** This is a transitional state when the workload is being undeployed. • **Error** An unknown error has occured.

Item	Description
CONNECT (4)	Clicking here opens an overlay, through which a remote connection to the workload can be established. Refer to Remote Connections for more information.
Control panel (5)	The following control options are available: • **Play** If the workload is in a stopped state, clicking **Play** will start the workload. • **Stop** If the workload is running, clicking **Stop** will stop the workload. • **Suspend** Clicking **Suspend** will pause the workload. It can be continued by clicking **Play**. • **Restart** This will restart the workload.
Undeploy (6)	Selecting this removes the workload from the node. The tile in the node detail screen disappears. The workload can be deployed again from the deployment menu.

Item	Description
Message window (7)	The message window displays the latest message the workload has sent out including a time stamp. The type of message that is displayed here depends on the workload. **Messages for VMs and Docker containers:** • "Domain creating." • "ERROR during creating! <errormessage>" • "Domain starting." • "ERROR during starting! <errormessage>" • "Domain <domainname> started." • "Domain stopping." • "ERROR during stopping! <errormessage>" • "Domain <domainname> stopped." • "Domain suspending." • "ERROR during suspending! <errormessage>" • "Domain <domainname> suspended." • "Domain resuming." • "ERROR during resuming! <errormessage>" • "Domain restarting." • "ERROR during restarting." • "Domain removing!!!" • "ERROR during removing." • "ERROR!!! Domain stopping." In the messages above, <domainname> is a placeholder for the name of the VM or Docker. In case of Docker containers, <errormessage> signifies a message that is generated by the Docker container if an error occurs. **Additional set of messages for VMs only:** • "Failed to connect to hypervisor." • "Failed to create domain." • "Domain <domainname> created." • "Cannot start <domainname> domain because it may already be running!" • "Failed to resume <domainname> domain!" <errormessage> • "Failed to start domain <domainname>. " <errormessage> In this case, <errormessage> is a message that is fetched from the libvirt library. **Messages from CODESYS workloads:** • "Preparing files for installation" • "Starting CODESYS application" • "CODESYS application started" • "Stopping CODESYS application" • "CODESYS application stoppped" • "Removing CODESYS application file" • "An unexpected error has occurred. <errormessage>" Here, <errormessage> is a message that is sent between the node and CODESYS.

Item	Description
Usage statistics (8)	Virtual Machine workloads and Docker workloads have their assigned resources they can use. The use of these resources is displayed with bar graphs: • **CPU load (VM and Docker)** The percentage here shows the usage of CPU resources in relation to the assigned CPUs. Example: A VM is assigned one CPU core out of four and the core is at 75 % usage capacity. The graph will be at 75 %. • **RAM (Docker only)** Similar to the CPU usage statistic, the percentage here shows the usage of system memory resources in relation to the assigned memory. If the assigned memory is at a 100 % usage capacity, the graph will be at 100 %. If no memory has been assigned, the graph will show the percentage of used memory in relation to the total available memory of the host.

NOTE

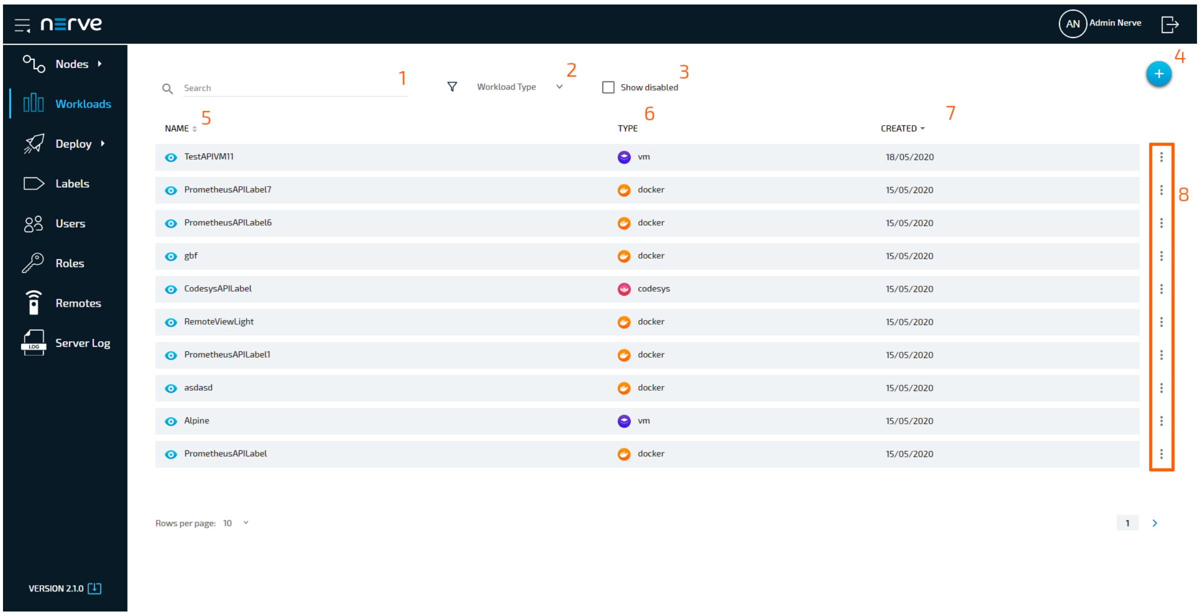
Since CODESYS workloads can only be controlled through the Local UI, the workload control screen does not offer any control options. It offers a message window. the option to undeploy the workload and the **CONNECT** button for establishing remote connections.

Workloads

In order to work with CODESYS applications, virtual machines or Docker containers on nodes, workloads need to be provisioned in the Management System. Here, provisioning is the creation of a workload and its storage in the workload repository of the Management System so that it can be deployed to nodes. This requires configuration of the workload and files that need to be uploaded to the Management System. After that, the workload can be deployed to nodes.

There are three types of workloads that can be provisioned: CODESYS workloads, Virtual Machine workloads and Docker workloads. The process of provisioning each workload is described in their respective chapters.

Select **Workloads** in the left-hand menu to find a list of all workloads that have been provisioned.



NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

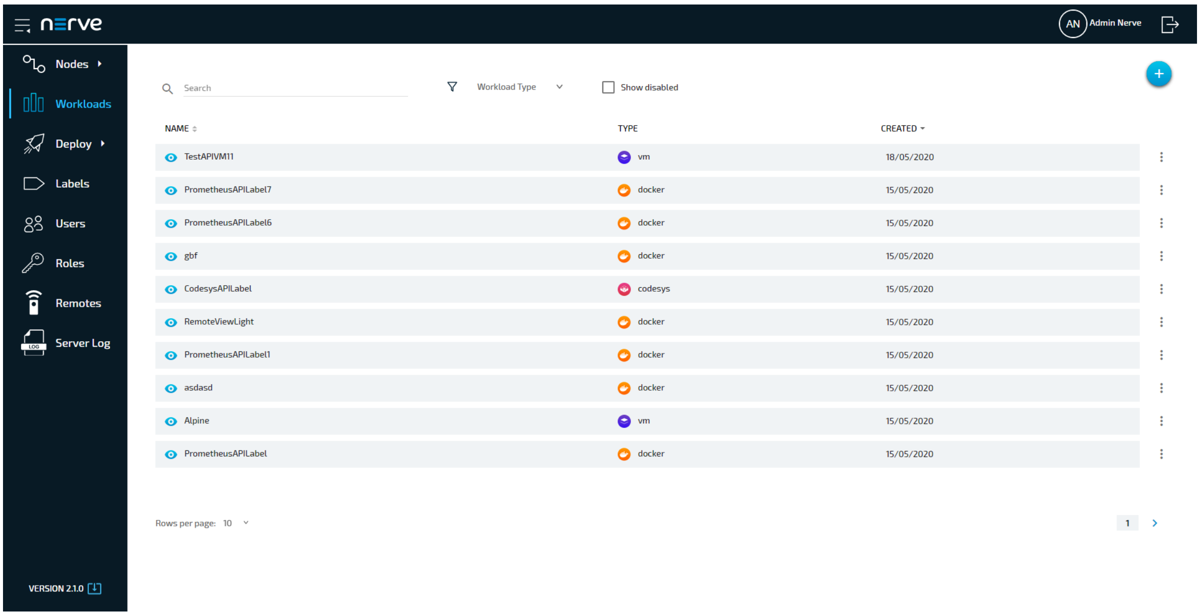
Item	Description
Search bar (1)	Use the search bar to filter workloads in the list by name. Select the cross symbol next to the search bar to reset the search field.
Workload Type (2)	This is a drop-down menu that allows filtering the list below by workload type. The available options are **VM**, **Docker**, **CODESYS** and **All**.
Show disabled (3)	Disabled workloads are not shown in the list of workloads. Ticking this checkbox shows them again. However, note that this does not enable the workloads again.
Add new workload (4)	Select the plus symbol to provision a new workload.
NAME (5)	This is the name of the workload that has been defined in the provisioning process. The list can be sorted in alphabetical and reverse alphabetical order by clicking **NAME**.
TYPE (6)	The type of workload is displayed here: **codesys**, **vm** or **docker**. The list can be sorted by workload type when clicking **TYPE**.
CREATED (7)	This is the date when the workload was provisioned in the format DD/MM/YYYY. The list can be sorted by creation date when clicking **CREATED**.
Ellipsis menu (8)	Clicking here opens an overlay that gives two options: **DELETE** and **DISABLE**.

Provisioning a workload

Provisioning a workload is the creation of a workload in the workload repository of the Management System. Workloads that have been provisioned are ready to be deployed to nodes.
Select the plus symbol in the upper-right corner of the workloads list to start provisioning a workload. The provisioning process of each workload type is covered separately in the following chapters.

- Provisioning a CODESYS workload
- Provisioning a Virtual machine workload
- Provisioning a Docker workload

After provisioning a workload, it will appear in the list of workloads.



NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

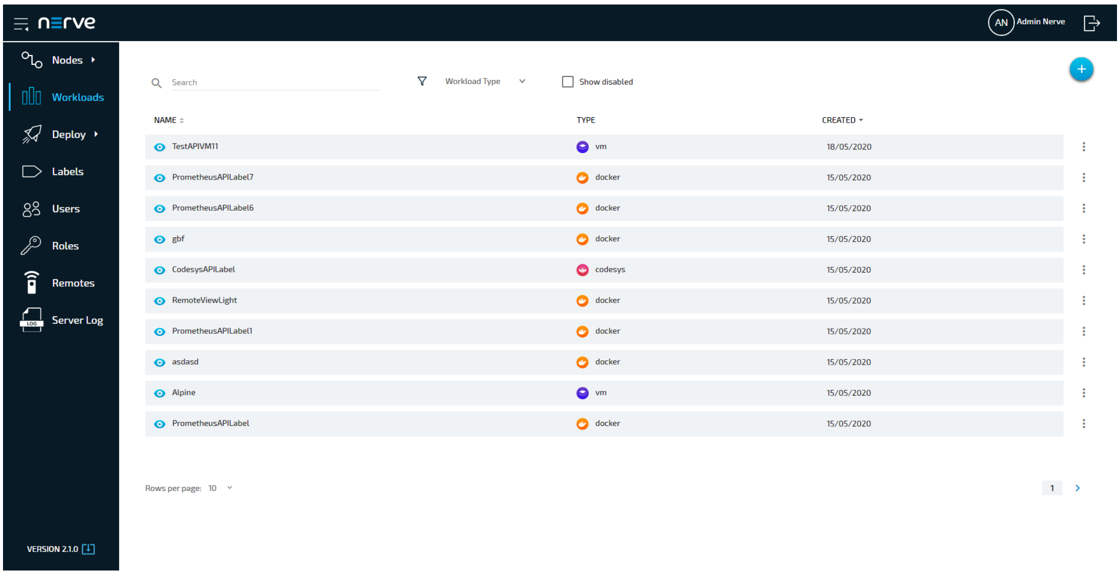
Rows per page: 10

1

Adding a new workload version

Add new versions to a provisioned workload to accommodate different use cases.

1. Select **Workloads** from the menu on the left side.
2. Select the workload to which a new version will be added.

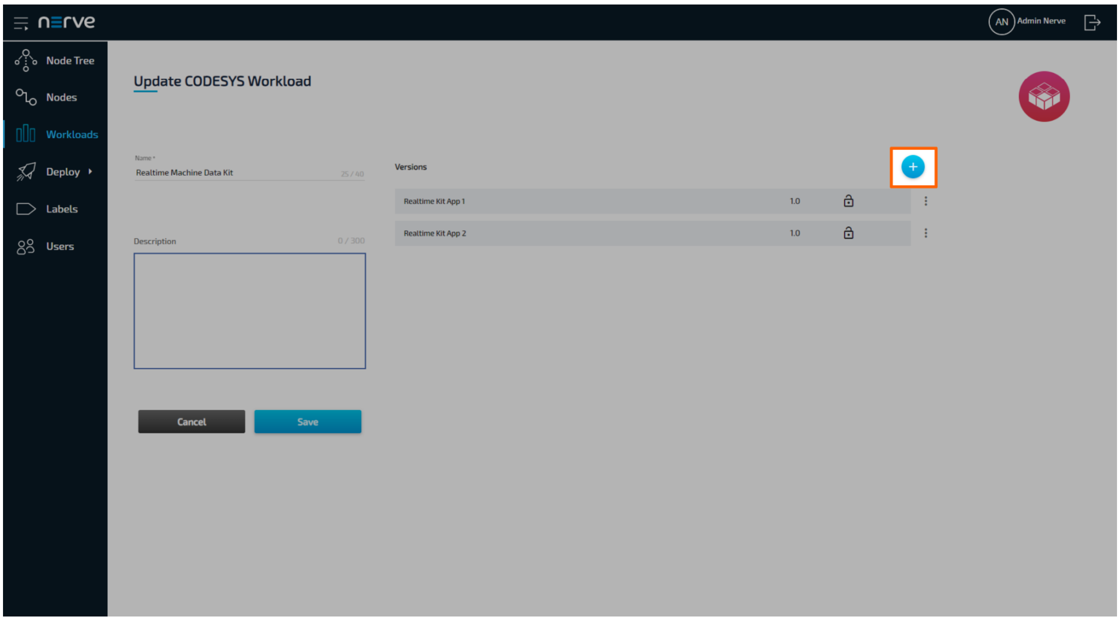


NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

Rows per page: 10

1

3. Click the plus symbol in the upper-right.



Update CODESYS Workload

Name *
Realtime Machine Data Kit 25 / 40

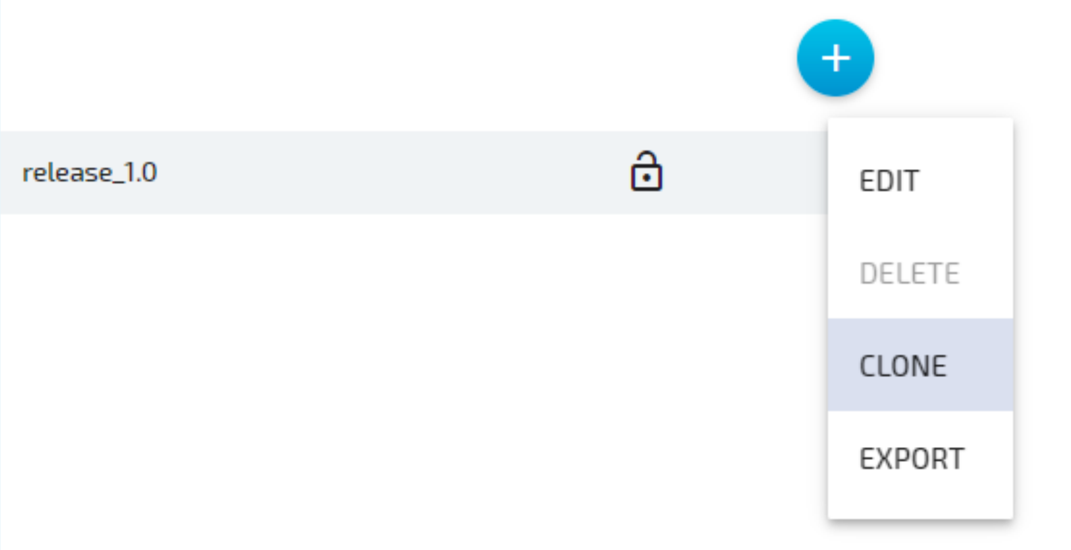
Description 0 / 300

Versions

Realtime Kit App 1	1.0
Realtime Kit App 2	1.0

NOTE

To add a new version that is a slight modification of an existing one, click the ellipsis menu next to a workload and select **CLONE** from the overlay.



4. Configure the new workload version. Refer to the provisioning chapters of each workload type linked above for more information on configuration settings.

NOTE

The fields of the new version will already have information filled in. The system enters the settings of the latest version automatically. If **CLONE** next to a workload version was used, the information filled in will be from that version instead.

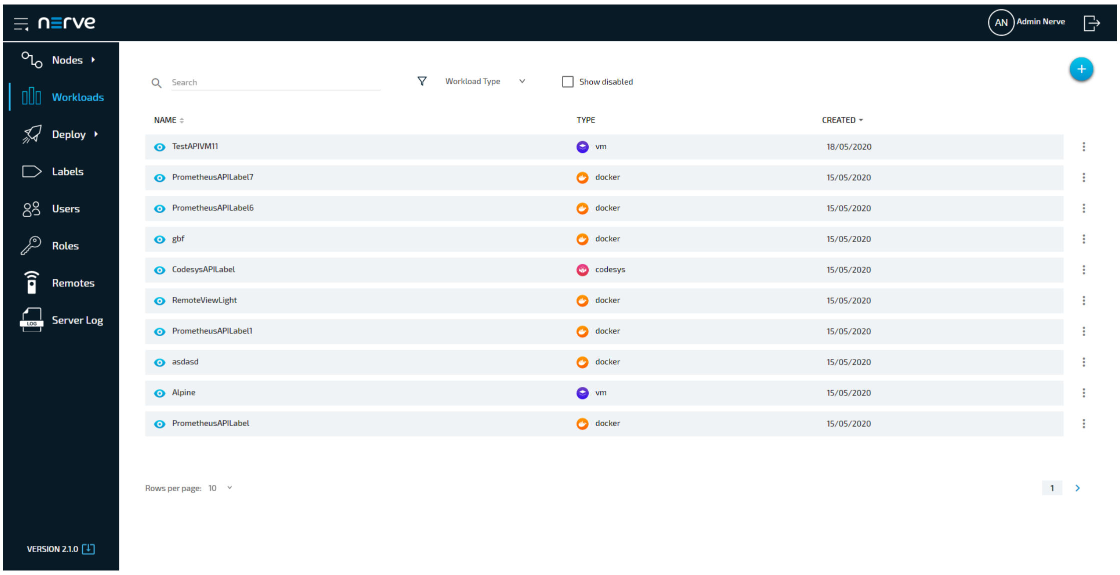
5. Click **Update** to save the new version of the workload.

Editing a workload

General information of a workload and configuration settings of each version can be edited starting from the workload list. General information of a workload is valid for all workload versions.

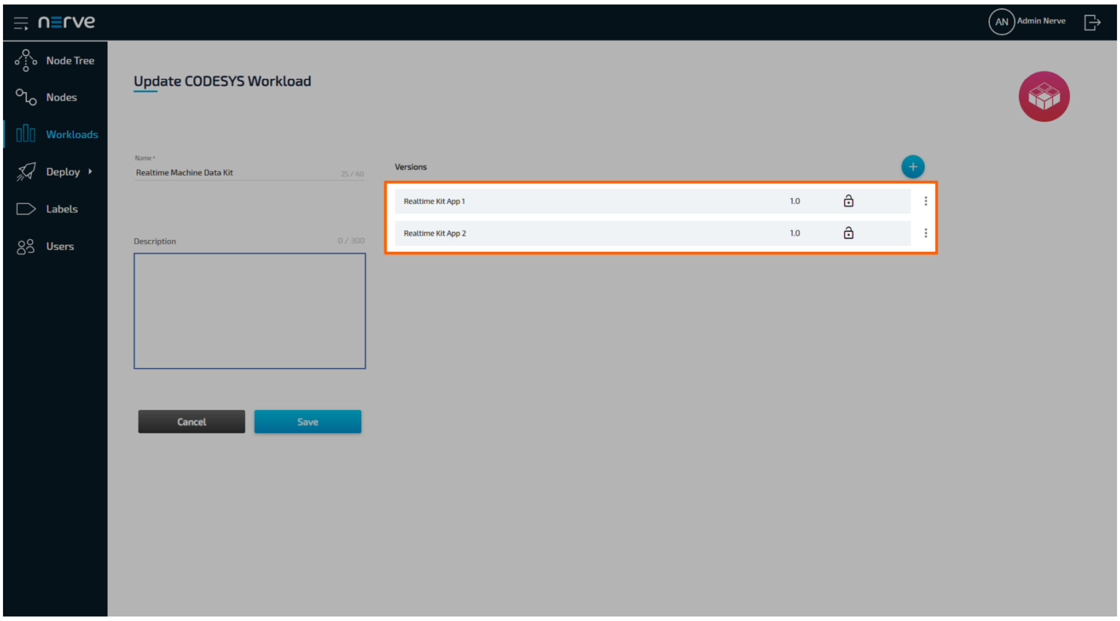
1. Select **Workloads** from the menu on the left side.

2. Select the workload to edit.



NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

3. Select the workload version to edit from the list on the right.



Update CODESYS Workload

Name * Realtime Machine Data Kit 25 / 40

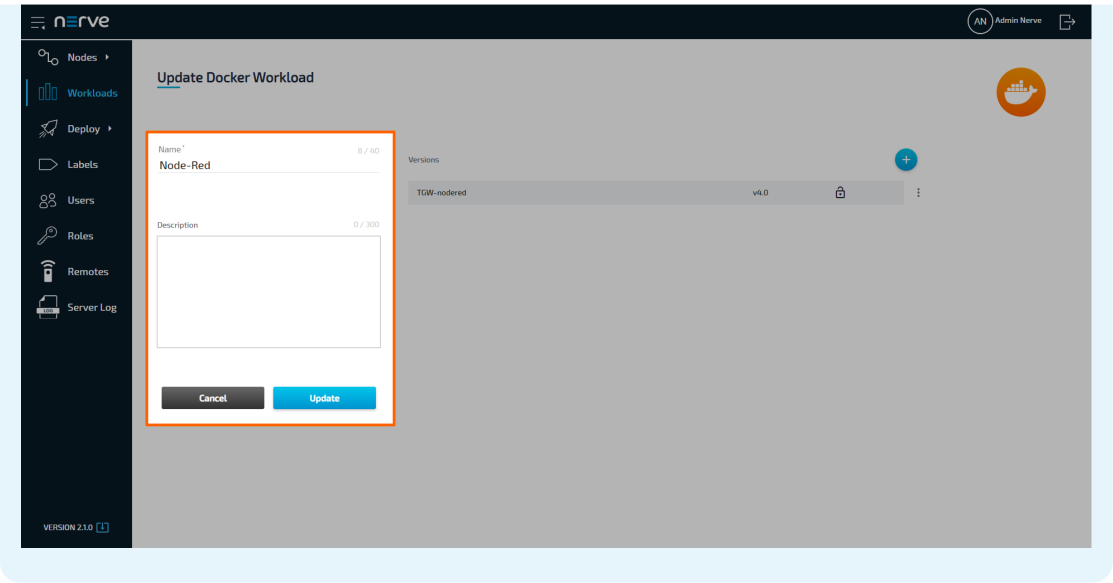
Description 0 / 300

Versions

Realtime Kit App 1	1.0
Realtime Kit App 2	1.0

NOTE

Editing the **Name** and **Description** of the workload can be done on the left side before selecting a version. Perform the desired changes and click **Update**.



Update Docker Workload

Name* 8 / 40
Node-Red

Description 0 / 300

Versions

TGW-nodered v4.0

4. Perform changes to the workload version.

> **NOTE**
>
> The settings of a workload depend on the workload type. See the version settings for each workload in the provisioning chapters: CODESYS workloads, Virtual Machine workloads and Docker workloads.

5. Click **Update** to save the changes.

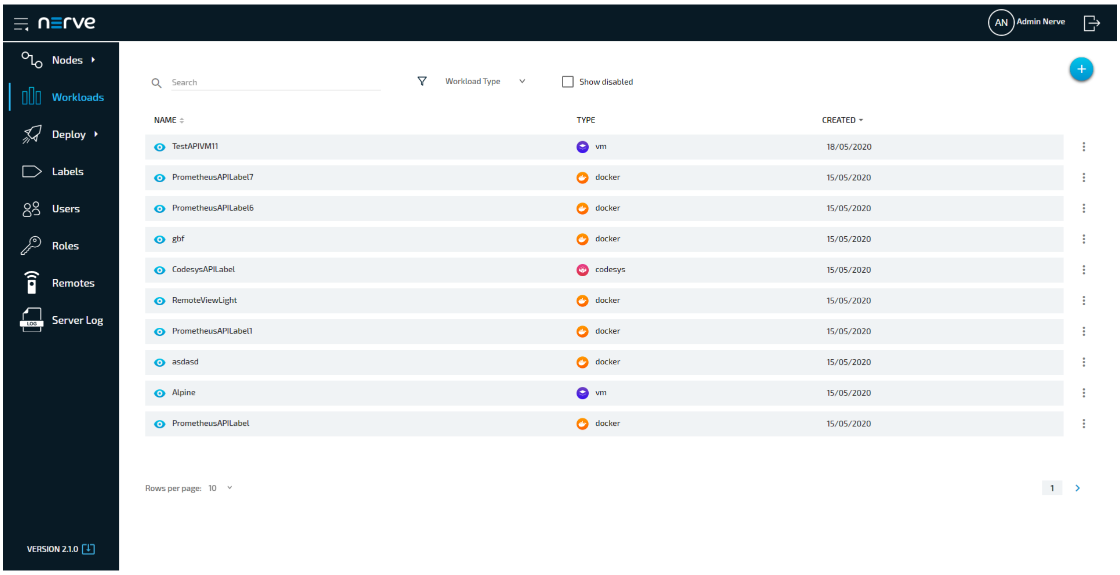
Note that a workload needs to be undeployed first before an updated version can be deployed. However, this does not apply to remote connections. Once a remote connection is configured, it is available immediately for the workload on all nodes that it was deployed to.

Disabling a workload

A workload can be disabled to make it hidden and not selectable. This will hide the workload in the workload list and deployment process but it will not be deleted from the Management System. This also means that the workload cannot be deployed. Workloads that have been deployed to nodes before are not affected.

1. Select **Workloads** from the menu on the left side.

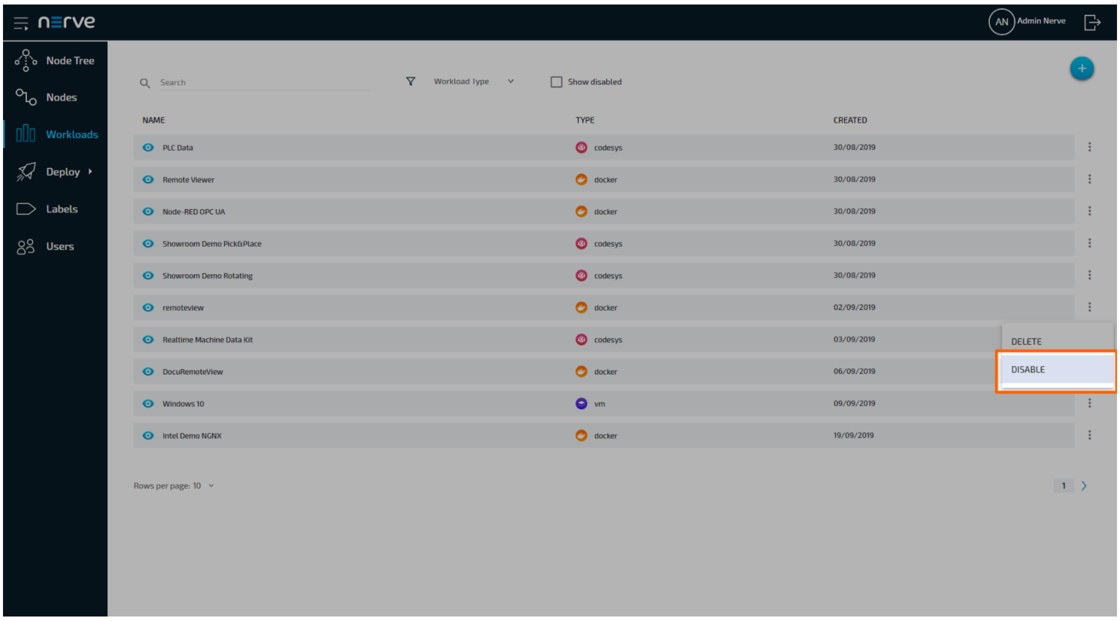
2. Choose the workload to disable.



NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

3. Click the ellipsis menu to the right of the workload.

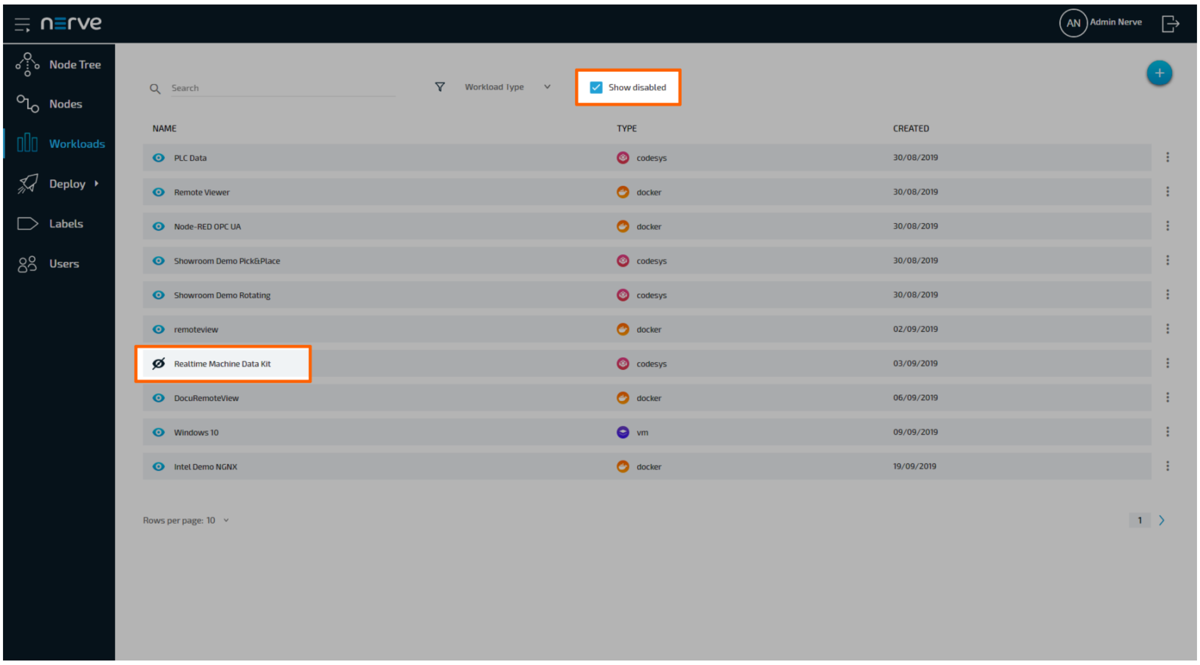
4. Select **DISABLE** from the overlay that appeared.



NAME	TYPE	CREATED
PLC Data	codesys	30/08/2019
Remote Viewer	docker	30/08/2019
Node-RED OPC UA	docker	30/08/2019
Showroom Demo Pick&Place	codesys	30/08/2019
Showroom Demo Rotating	codesys	30/08/2019
remoteview	docker	02/09/2019
Realtime Machine Data Kit	codesys	03/09/2019
DocuRemoteView	docker	06/09/2019
Windows 10	vm	09/09/2019
Intel Demo NGNX	docker	19/09/2019

5. Click **OK** in the new window.

The workload is now disabled and hidden in the list. To show disabled workloads again, tick the checkbox next to **Show Disabled** in the list of workloads. All disabled workloads are marked by an icon resembling a crossed out eye.



☑ Show disabled

NAME	TYPE	CREATED
PLC Data	codesys	30/08/2019
Remote Viewer	docker	30/08/2019
Node-RED OPC UA	docker	30/08/2019
Showroom Demo Pick&Place	codesys	30/08/2019
Showroom Demo Rotating	codesys	30/08/2019
remoteview	docker	02/09/2019
Realtime Machine Data Kit	codesys	03/09/2019
DocuRemoteView	docker	06/09/2019
Windows 10	vm	09/09/2019
Intel Demo NGNX	docker	19/09/2019

Rows per page: 10

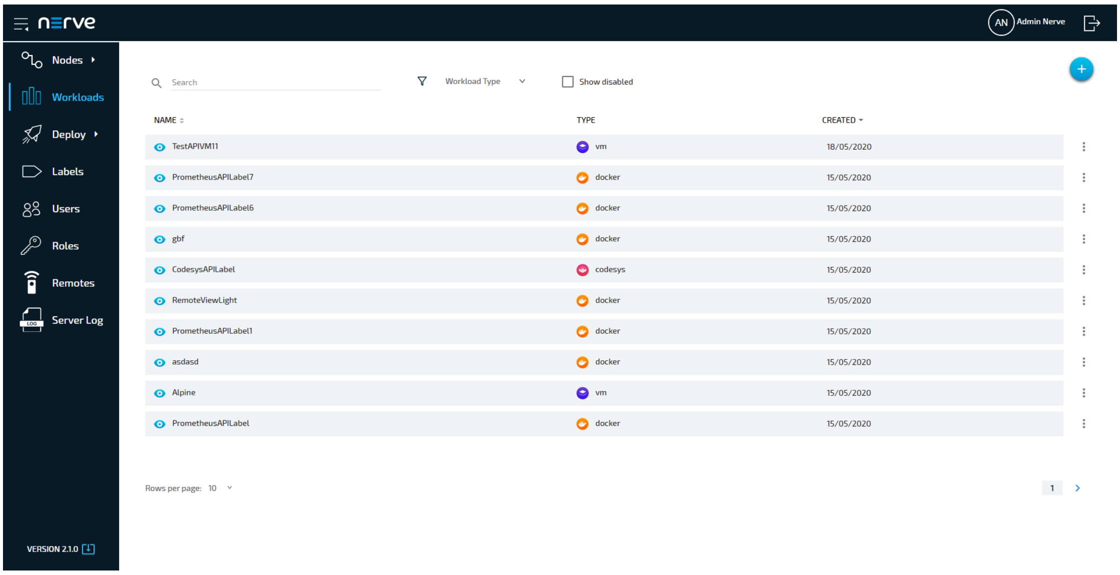
Follow the steps above to enable the workload again. However, select **ENABLE** in the overlay.

Deleting a workload

The instructions below cover the deletion of a workload from the repository in the Management System. Note that deleting a workload from the Management System will not automatically remove the workload from nodes. To remove a workload from a node, undeploy the workload in the node tree.

1. Select **Workloads** from the menu on the left side.

2. Choose the workload to delete.



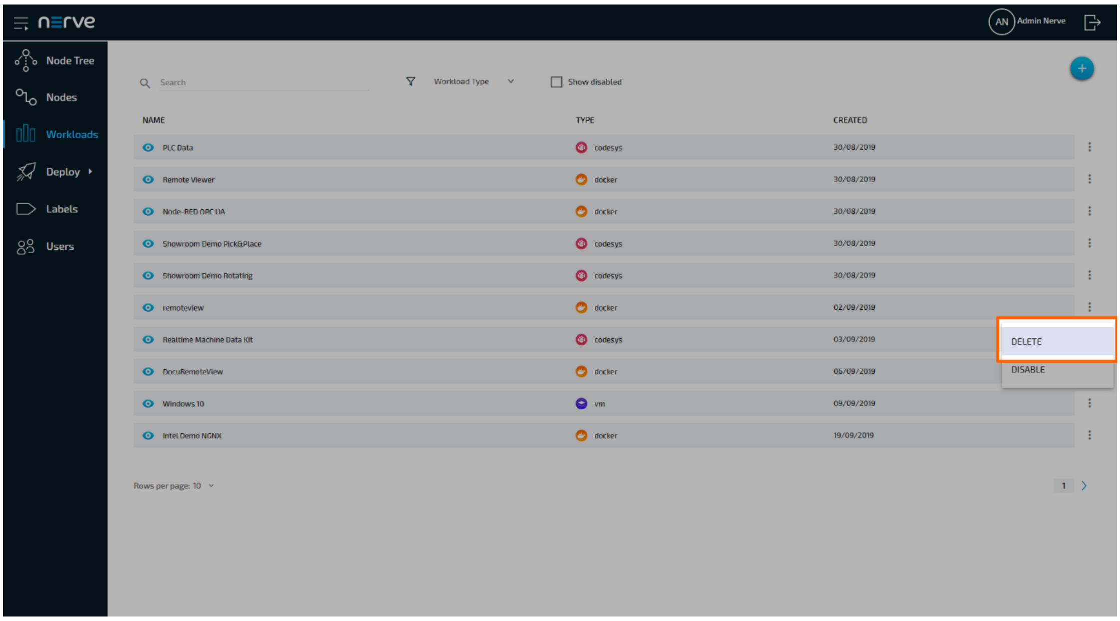
□ Show disabled

NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

Rows per page: 10

3. Click the ellipsis menu to the right of the workload.

4. Select **DELETE** in the overlay that appeared.



NAME	TYPE	CREATED
PLC Data	codesys	30/08/2019
Remote Viewer	docker	30/08/2019
Node-RED OPC UA	docker	30/08/2019
Showroom Demo Pick&Place	codesys	30/08/2019
Showroom Demo Rotating	codesys	30/08/2019
remoteview	docker	02/09/2019
Realtime Machine Data Kit	codesys	03/09/2019
DocuRemoteView	docker	06/09/2019
Windows 10	vm	09/09/2019
Intel Demo NGNX	docker	19/09/2019

5. Click **OK** in the new window to confirm the deletion.

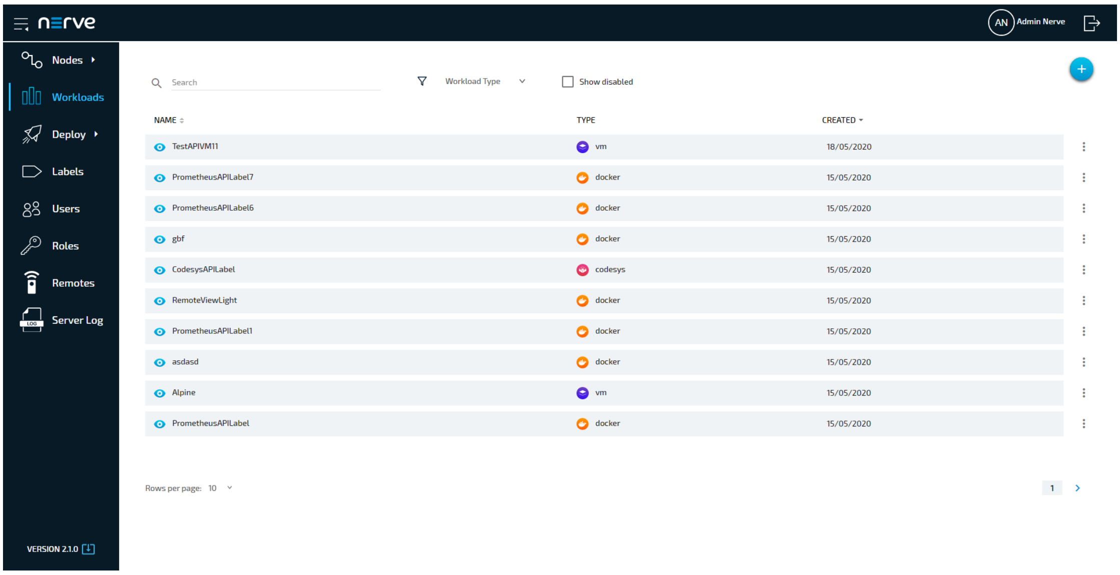
> **NOTE**
>
> Deleting a workload will automatically delete all versions of the workload as well.

Deleting a workload version

Workload versions can be deleted separately in the workload details. Note that deleting a workload version is not possible if only one version of the workload exists. Delete the workload to delete the only workload version.

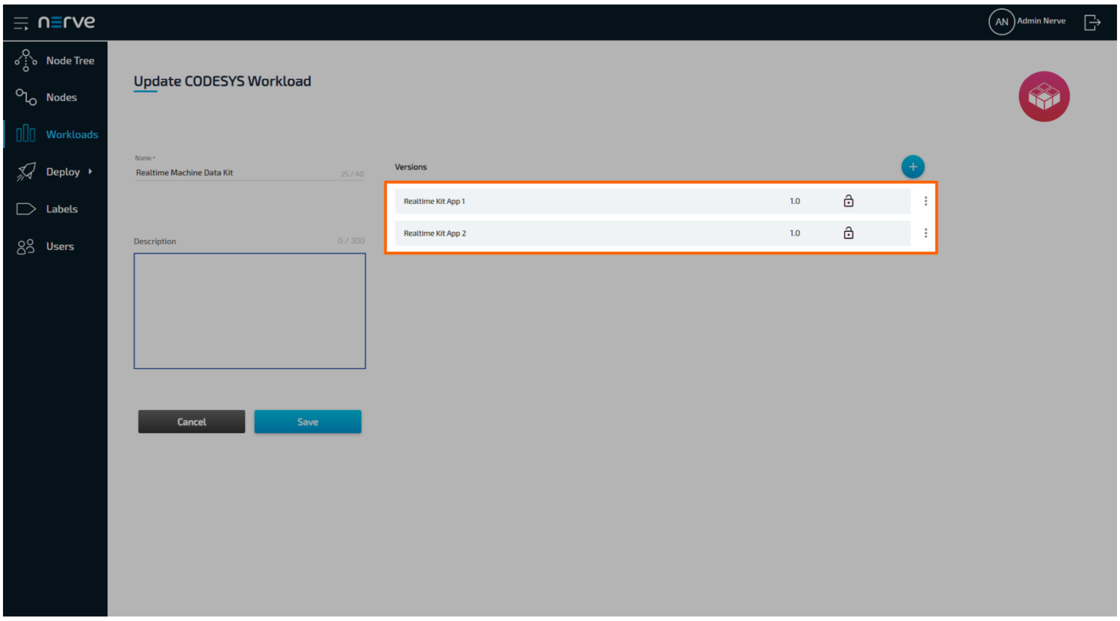
1. Select **Workloads** from the menu on the left side.

2. Select the workload of which a version will be deleted.



NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

3. Choose the workload version to delete.



Update CODESYS Workload

Name* Realtime Machine Data Kit 25 / 40

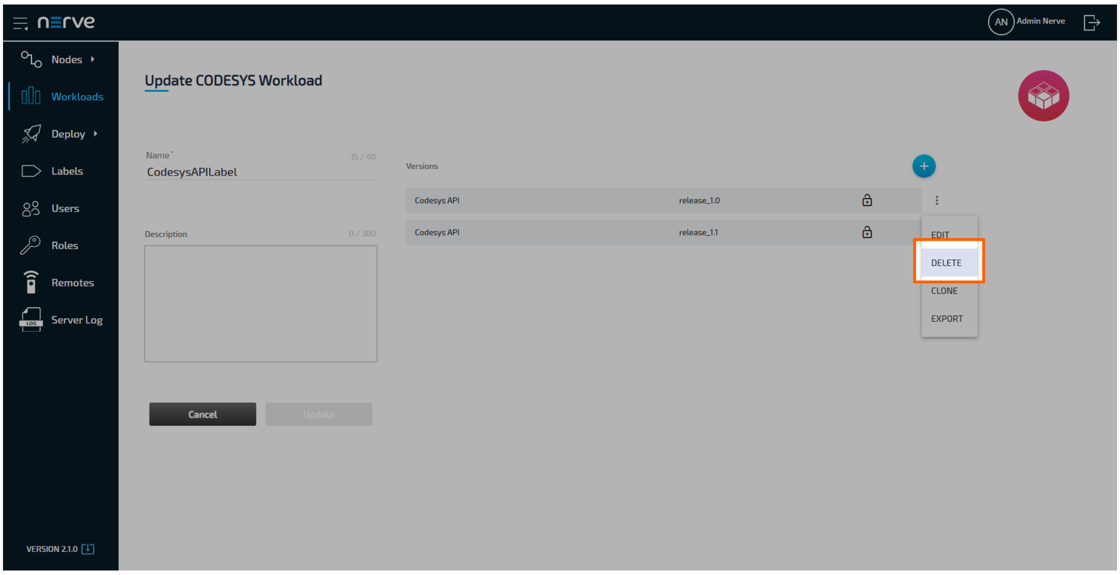
Description 0 / 300

Versions

Realtime Kit App 1	1.0	
Realtime Kit App 2	1.0	

4. Click the ellipsis menu to the right of the workload version.

5. Select **DELETE** in the overlay that appeared.



Update CODESYS Workload

Name* CodesysAPILabel 15 / 40

Description 0 / 300

Versions

Codesys API	release_1.0	
Codesys API	release_1.1	

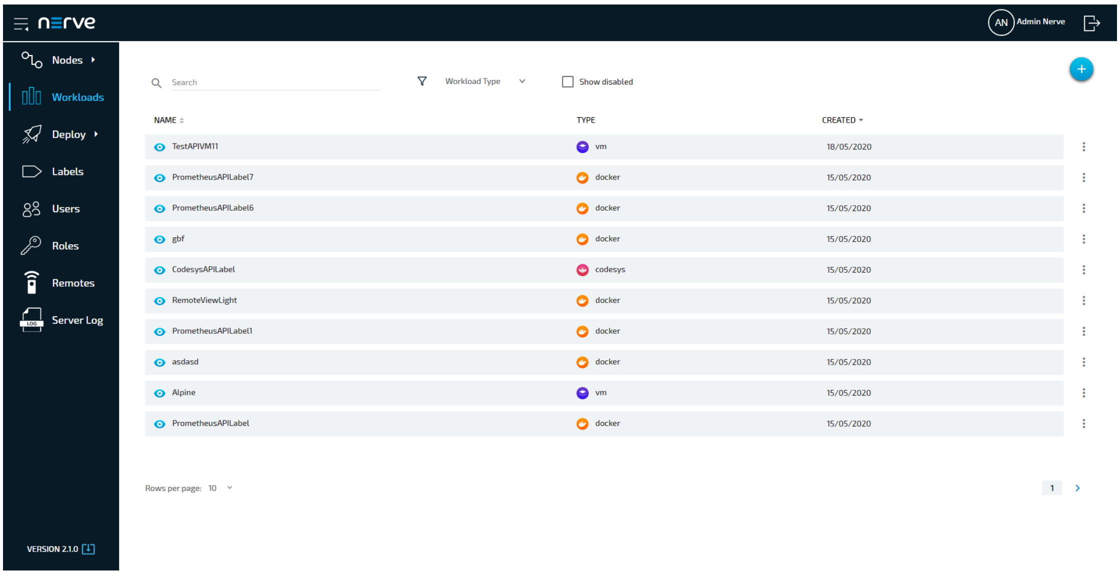
6. Click **OK** in the new window to confirm the deletion.

Exporting a workload

Workloads can be manually deployed to nodes. To do that, a workload version must be exported. Refer to Local Workload Deployment for more information.

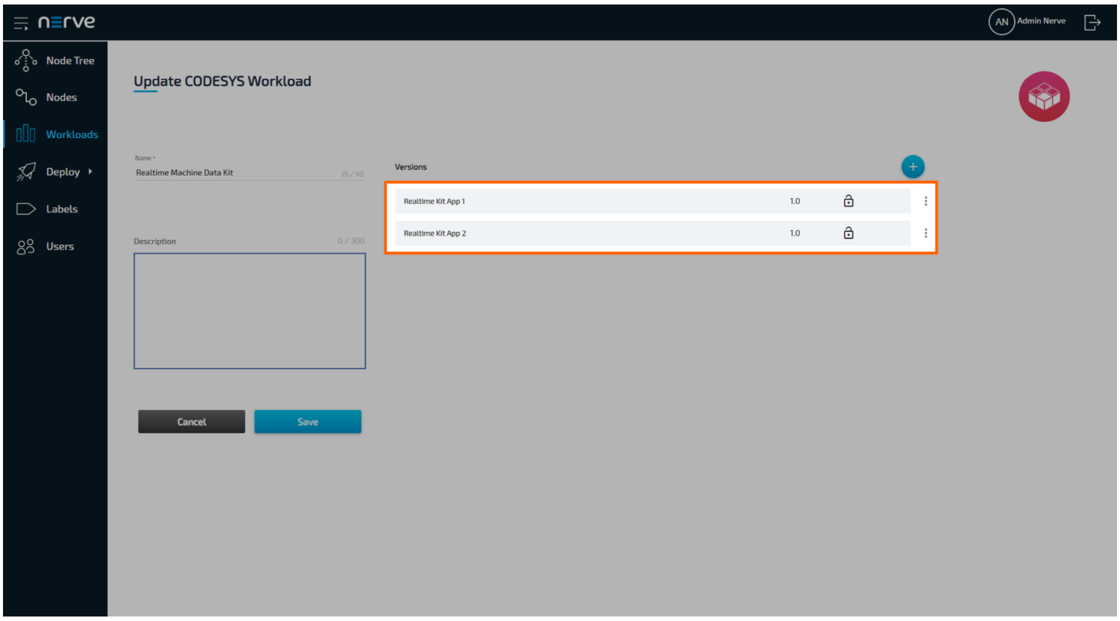
1. Select **Workloads** from the menu on the left side.

2. Select the workload of which a version will be exported.



NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

3. Choose the workload version to export.



Update CODESYS Workload

Name *
Realtime Machine Data Kit 25 / 40

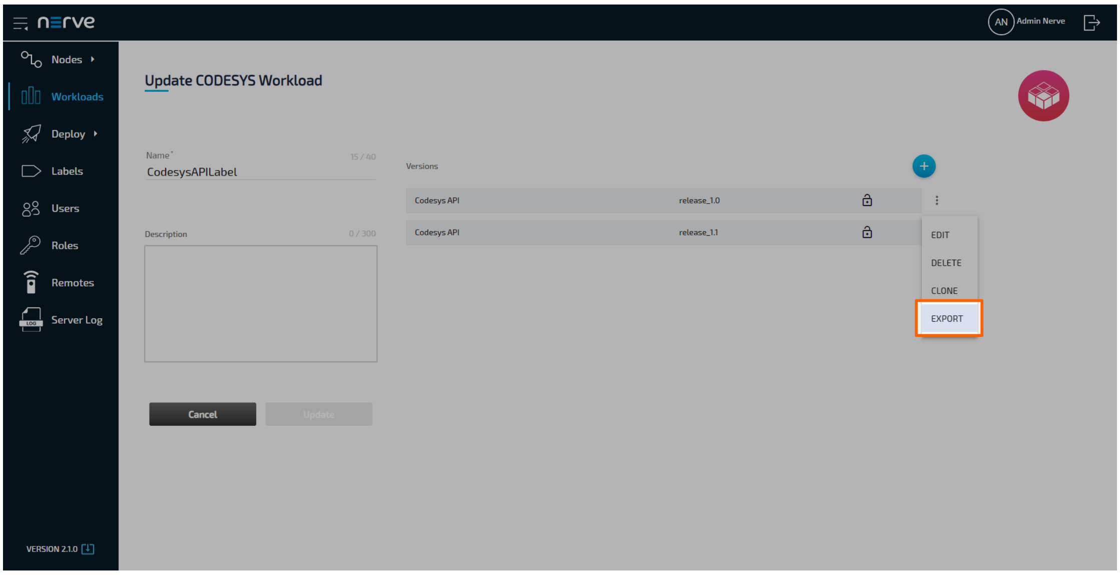
Description 0 / 300

Versions

Realtime Kit App 1	1.0	🔒
Realtime Kit App 2	1.0	🔒

4. Click the ellipsis menu to the right of the workload version.

5. Select **EXPORT** in the overlay that appeared.



Update CODESYS Workload

Name* CodesysAPILabel 15 / 40

Description 0 / 300

Versions

Codesys API	release_1.0	
Codesys API	release_1.1	

The download of the exported workload version is started automatically. The workload version is compressed into a TAR file.

Controlling a workload

Workloads can be controlled from the node details view in the node tree. Refer to the node tree chapter for more information.

Provisioning a CODESYS workload

Before a CODESYS workload can be provisioned, a CODESYS application has to be loaded into the CODESYS runtime first. Refer to First steps with CODESYS first before continuing.

Once a CODESYS application has been loaded into the CODESYS runtime, the following steps have to be taken before the workload can be provisioned:

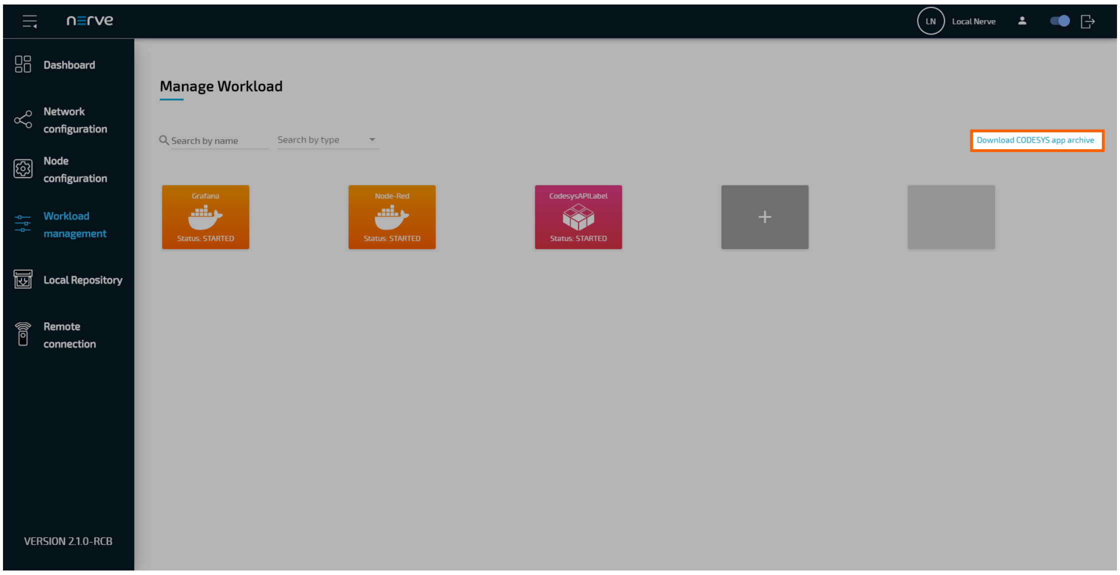
1. Creating the ZIP file of the CODESYS application
2. Transferring the ZIP file to a local workstation

Also the workstation needs to be connected to the physical port of the Nerve Device associated with host access, and the network adapter IP address of the workstation needs to be configured in the correct range. This information is device specific. Refer to the device guide for information on the Nerve Device.

Creating the ZIP file on the Nerve Device

First, the CODESYS project needs to be zipped on the Nerve Device before it can be copied from the CODESYS runtime. This is done through the Local UI.

1. Follow the link to the Local UI of the used Nerve Device. Refer to the device guide for more information.
2. Select **Workload management** in the navigation on the left.
3. Click **Download CODESYS app archive**.



4. Select **YES** in the pop-up. Note that the CODESYS application will be stopped.

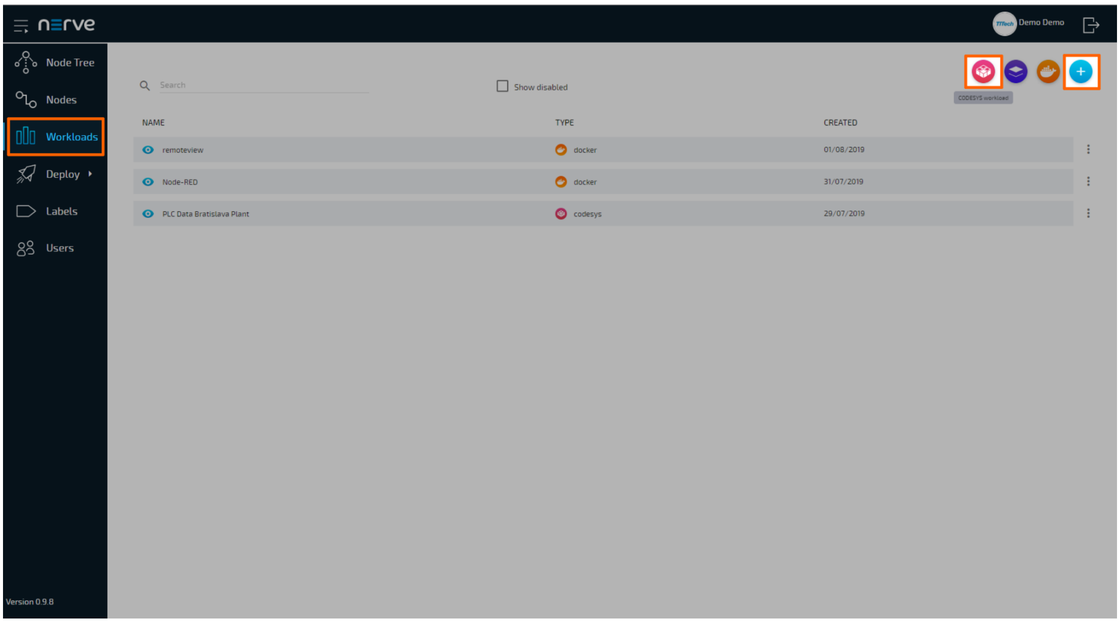
The ZIP file is automatically downloaded to the workstation and a CODESYS workload can now be provisioned in the Management System.

Provisioning a CODESYS workload

The following instructions cover the basic requirements for provisioning a CODESYS workload. Optional settings will be left out. Extended options are addressed in the last section of this chapter. There are two further types of workloads that can be provisioned: Virtual Machine workloads and Docker workloads. The process for each workload is highlighted in its respective chapter.

1. Log in to the Management System.
2. Select **Workloads** in the left-hand menu.
3. Select the plus symbol in the upper-right corner.

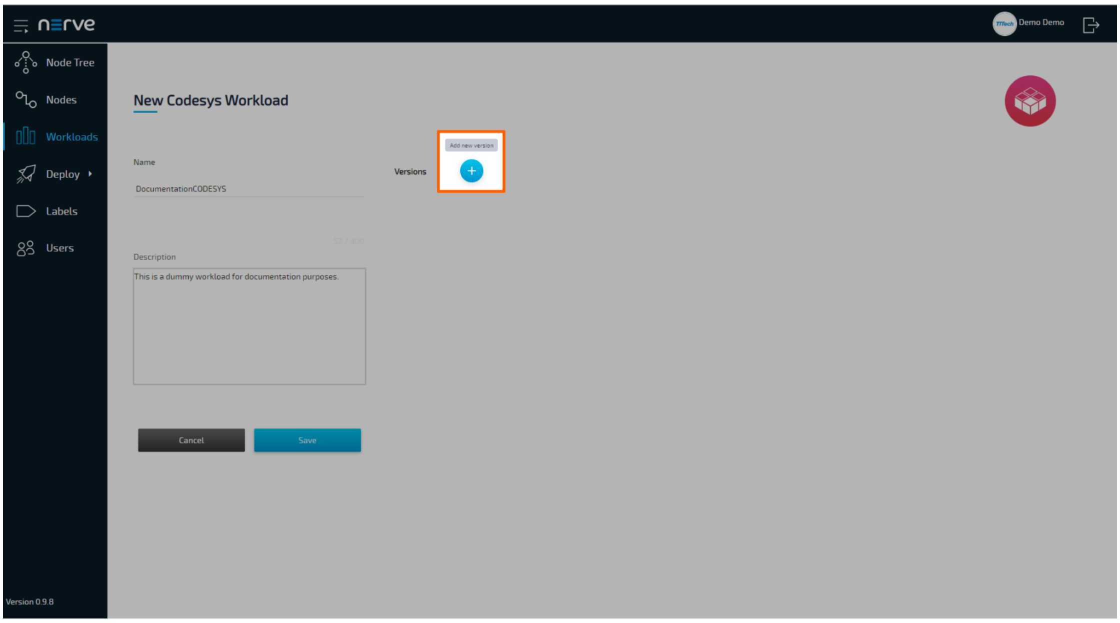
4. Select the CODESYS symbol (**CODESYS workload**) on the left of the three symbols that expanded.



NAME	TYPE	CREATED
remoteview	docker	01/08/2019
Node-RED	docker	31/07/2019
PLC Data Bratislava Plant	codesys	29/07/2019

5. In the new window, enter a name for the workload.

6. Select the plus symbol next to **Versions** to add a new version of the workload.



New Codesys Workload

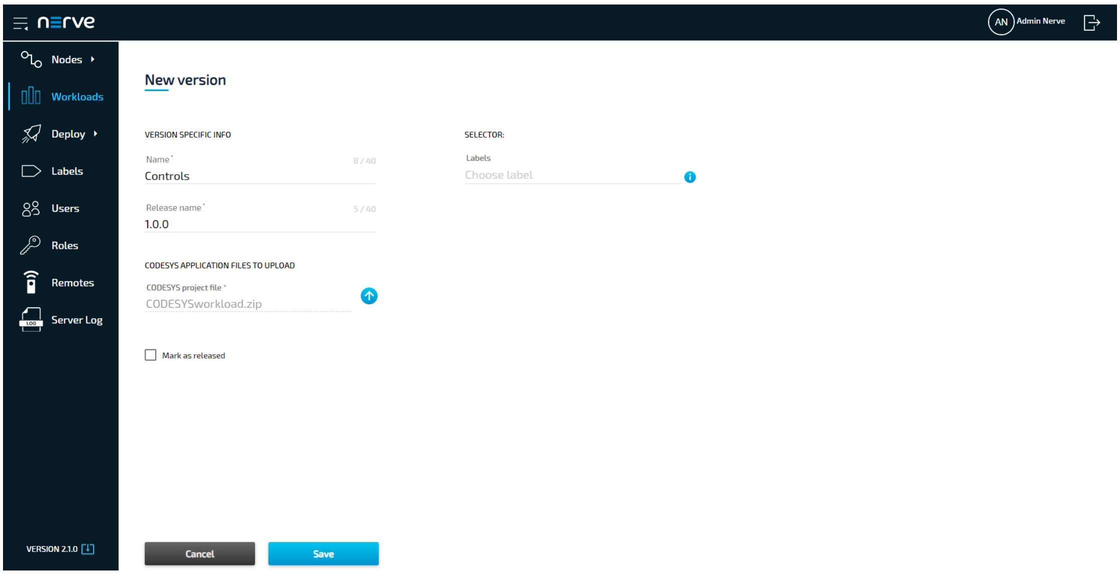
Name
DocumentationCODESYS

Versions

Description
This is a dummy workload for documentation purposes.

7. In the new window, enter the following information:

Item	Description
Name	Enter a **Name** for the version of this workload.
Release name	Enter a **Release name** for the version of this workload.
CODESYS project file	Click the **upward arrow** symbol to add the CODESYS application ZIP file. This is the ZIP file that was created above.



New version

VERSION SPECIFIC INFO

Name * 8 / 40
Controls

Release name * 5 / 40
1.0.0

CODESYS APPLICATION FILES TO UPLOAD

CODESYS project file *
CODESYSworkload.zip

□ Mark as released

SELECTOR:

Labels
Choose label

8. Click **Save**.

The workload has now been provisioned and is ready to be deployed in the **Deploy** menu.

Settings for CODESYS workloads

In the instructions above, optional settings have been left out. Below is an overview of all options with an explanation to each option.

Setting	Description
VERSION SPECIFIC INFO	**Name** A name for the workload version. Choose a precise name to make the workload version unambiguous. **Release name** A release name for the workload version. This could be a version number. Example: 1.0.1
CODESYS APPLICATION FILES TO UPLOAD	Upload the **CODESYS project file** here. This is a ZIP file that has to be generated from a CODESYS project running in the CODESYS runtime. Upload it here by clicking the **upward arrow** symbol to open the file browser.
Mark as released	Tick this checkbox to mark this workload as released. Once marked as released, the workload cannot be edited anymore.
SELECTOR	**Labels** If labels have been defined and assigned to nodes, add them as selectors to the workload. When deploying a workload, the list of nodes will be filtered automatically to the specified label.

Provisioning a Virtual machine workload

Before a Virtual Machine workload can be provisioned, it is required to set up the virtual machine. Virtual machine workloads can be created on any PC but an environment to effectively create a Virtual machine workload is not always readily available. Therefore, it is described how the Nerve Device can be utilized to create a Virtual machine workload. What needs to be done is:

1. Creating a virtual machine on the node
2. Installing an operating system on the virtual machine
3. Obtaining the virtual machine IMG and XML files
4. Provisioning the Virtual Machine workload in the Management System

In this version, Nerve Blue does not provide a GUI based method for installing an OS on a virtual machine and obtaining the virtual machine IMG and XML configuration files. Therefore, this chapter focuses on the manual process. Three tools are required for the instructions below, assuming Windows is used on the workstation:

- an X Server application like Xming
- an SSH client like PuTTY
- a file transfer client like WinSCP

Also the workstation needs to be connected to the physical port of the Nerve Device associated with host access, and the network adapter IP address of the workstation needs to be configured in the correct range. This information is device specific. Refer to the device guide for information on the Nerve Device.

The instructions below are split up into multiple parts to make them easier to follow. The subsections of the instructions are connected and every subheading is a requirement for the next paragraph.

NOTE

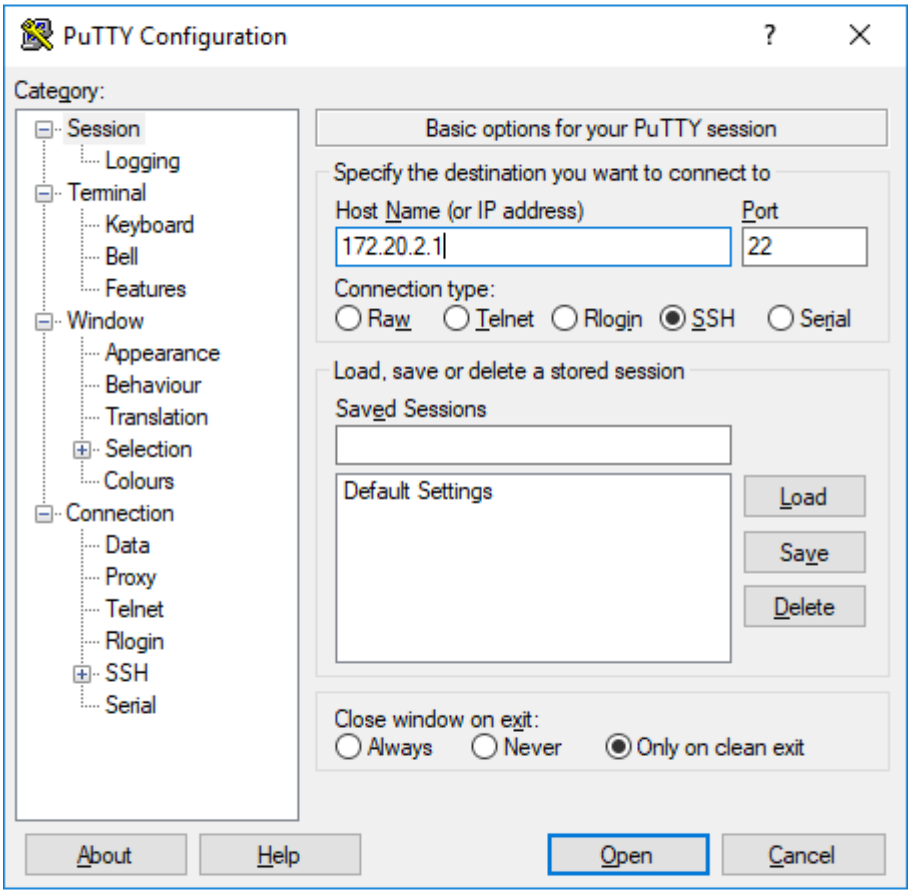
The virtual machine generated in this chapter is a fresh installation and will be generated on the Nerve Device directly.

Creating a logical volume

Logical volumes are created using the `lvcreate` command, which takes a number of command line arguments. Firstly, the `-L` flag is used to specify the size of the volume. Secondly, the `-n` flag is used to specify a name for the logical volume. `<image_size>` and `<volume_name>` are used as placeholders in the instructions below. The volume group is already predefined with `nerve` as its name.

1. Open an SSH client like PuTTY.

2. Enter the IP address for host access to the Nerve Device under **Host Name (or IP address)** to log in to the host of the Nerve Device.



Host Name (or IP address): 172.20.2.1
Port: 22

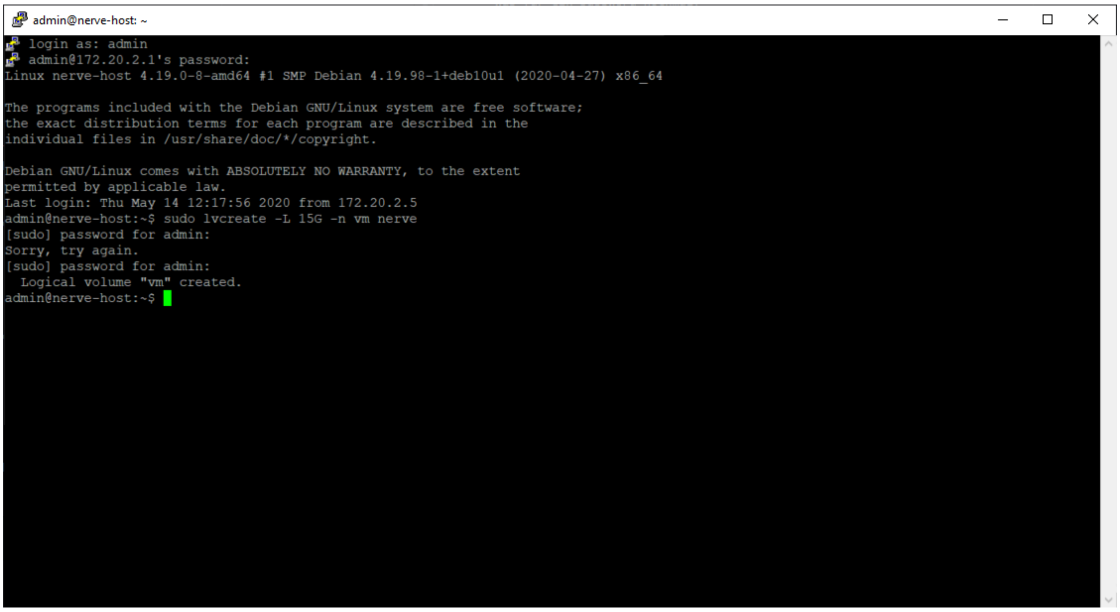
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

3. Log in with the credentials for host access to the Nerve Device.

4. Enter the following command:

```
sudo lvcreate -L <image_size> -n <volume_name> nerve
```



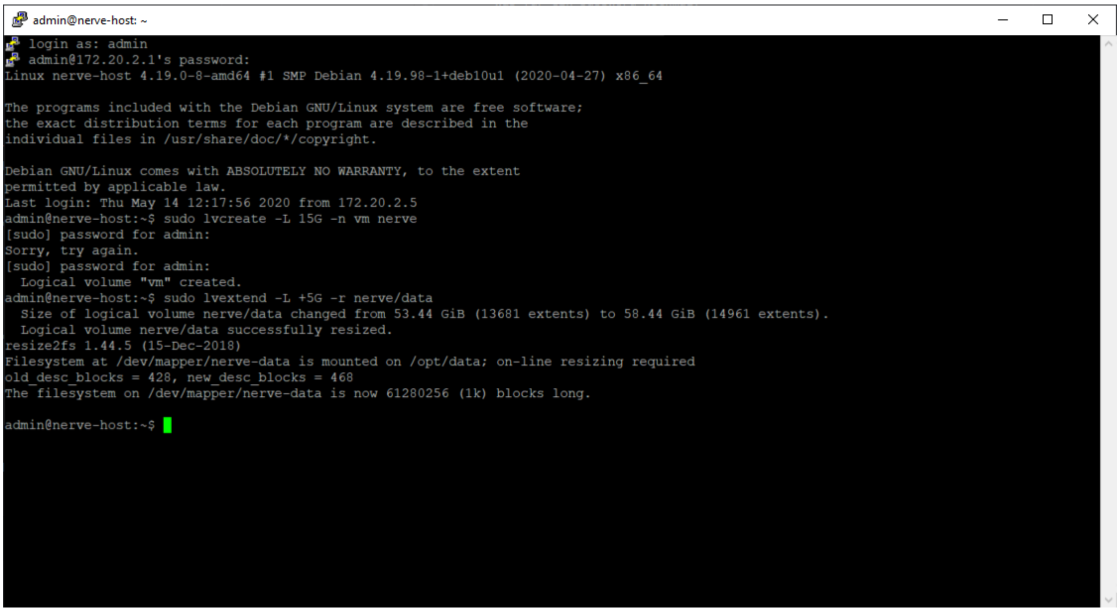
```
login as: admin
admin@172.20.2.1's password:
Linux nerve-host 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1+deb10u1 (2020-04-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 14 12:17:56 2020 from 172.20.2.5
admin@nerve-host:~$ sudo lvcreate -L 15G -n vm nerve
[sudo] password for admin:
Sorry, try again.
[sudo] password for admin:
  Logical volume "vm" created.
admin@nerve-host:~$
```

Increasing storage in /opt/data and creating a filesystem

The ISO image of the OS needs to be copied to the device from which the virtual machine will be installed. In order to do that, the size of `/opt/data` needs to be increased. A path to store the ISO file will also be created. `<size>` is a placeholder for the amount of storage that is added, while `nerve/data` is the name of the volume group and the logical volume.

1. Enter the following command to increase storage:

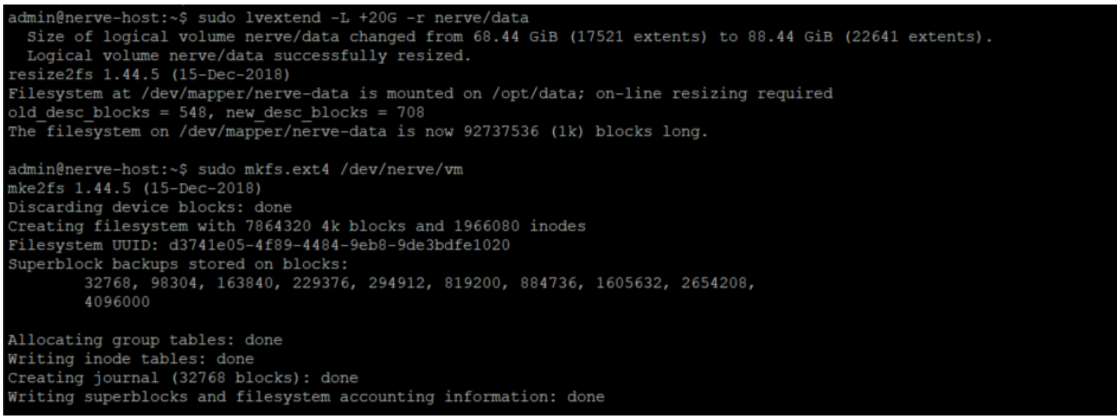
   ```
   sudo lvextend -L +<size> -r nerve/data
   ```



```
login as: admin
admin@172.20.2.1's password:
Linux nerve-host 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1+deb10u1 (2020-04-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 14 12:17:56 2020 from 172.20.2.5
admin@nerve-host:~$ sudo lvcreate -L 15G -n vm nerve
[sudo] password for admin:
Sorry, try again.
[sudo] password for admin:
  Logical volume "vm" created.
admin@nerve-host:~$ sudo lvextend -L +5G -r nerve/data
  Size of logical volume nerve/data changed from 53.44 GiB (13681 extents) to 58.44 GiB (14961 extents).
  Logical volume nerve/data successfully resized.
resize2fs 1.44.5 (15-Dec-2018)
Filesystem at /dev/mapper/nerve-data is mounted on /opt/data; on-line resizing required
old_desc_blocks = 428, new_desc_blocks = 468
The filesystem on /dev/mapper/nerve-data is now 61280256 (1k) blocks long.

admin@nerve-host:~$
```

2. Enter the following command to create a filesystem:

   ```
   sudo mkfs.ext4 /dev/nerve/<volume_name>
   ```



```
admin@nerve-host:~$ sudo lvextend -L +20G -r nerve/data
  Size of logical volume nerve/data changed from 68.44 GiB (17521 extents) to 88.44 GiB (22641 extents).
  Logical volume nerve/data successfully resized.
resize2fs 1.44.5 (15-Dec-2018)
Filesystem at /dev/mapper/nerve-data is mounted on /opt/data; on-line resizing required
old_desc_blocks = 548, new_desc_blocks = 708
The filesystem on /dev/mapper/nerve-data is now 92737536 (1k) blocks long.

admin@nerve-host:~$ sudo mkfs.ext4 /dev/nerve/vm
mke2fs 1.44.5 (15-Dec-2018)
Discarding device blocks: done
Creating filesystem with 7864320 4k blocks and 1966080 inodes
Filesystem UUID: d3741e05-4f89-4484-9eb8-9de3bdfe1020
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Creating a directory for the ISO file and mounting the filesystem

Create a directory in `/opt/data` in order to be able to generate the IMG file and XML file that are required for the provisioning of the Virtual machine in the Management System

1. Enter the following command to create a directory for the ISO file:

   ```
   sudo mkdir /opt/data/<directory>
   ```

2. Enter the following command to mount the filesystem:

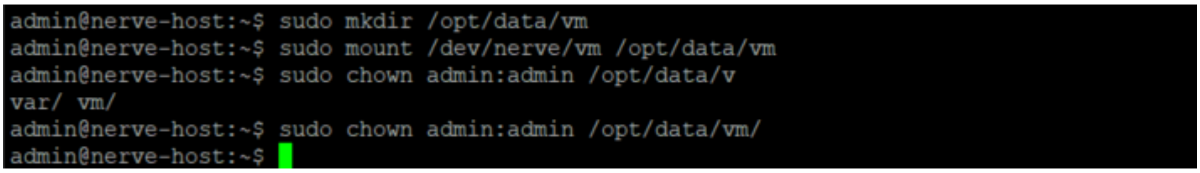
   ```
   sudo mount /dev/nerve/<volume_name> /opt/data/<directory>
   ```

Granting permission for the newly created directory

Take ownership of the new directory in order to be able to work with it, e.g. copying the ISO file into the directory.

Enter the following command to take ownership of the created directory:

```
sudo chown admin:admin /opt/data/<directory>
```



```
admin@nerve-host:~$ sudo mkdir /opt/data/vm
admin@nerve-host:~$ sudo mount /dev/nerve/vm /opt/data/vm
admin@nerve-host:~$ sudo chown admin:admin /opt/data/v
var/ vm/
admin@nerve-host:~$ sudo chown admin:admin /opt/data/vm/
admin@nerve-host:~$ 
```

Copying the ISO file to the Nerve Device

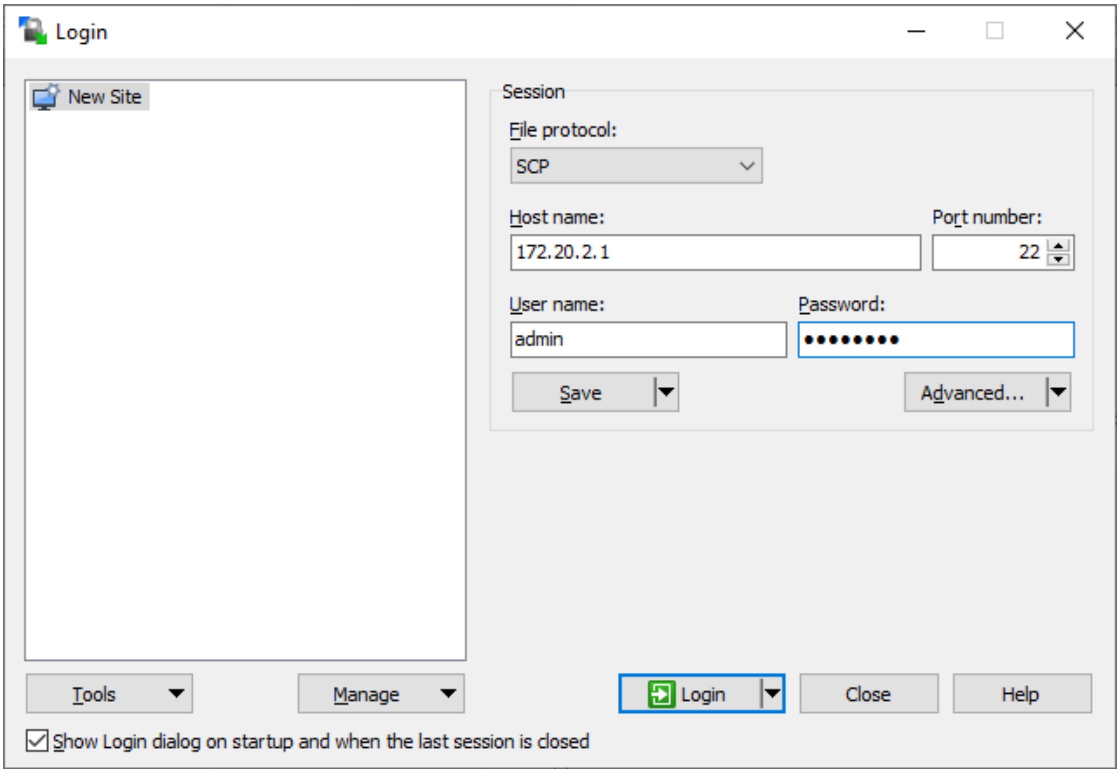
With the directory created and ownership established, copy the ISO file of the OS to the Nerve Device for the installation on the virtual machine.

NOTE

The instructions below are hardware specific. The MFN 100 is used as an example in the screenshots. Refer to the device guide for specific information on the Nerve Device.

1. Open a file transfer client like WinSCP.
2. Enter the IP address for host access to the Nerve Device under **Host Name**.

3. Enter the credentials for host access to the Nerve Device below under **User name** and **Password**.



Login

Session

File protocol: SCP

Host name: 172.20.2.1

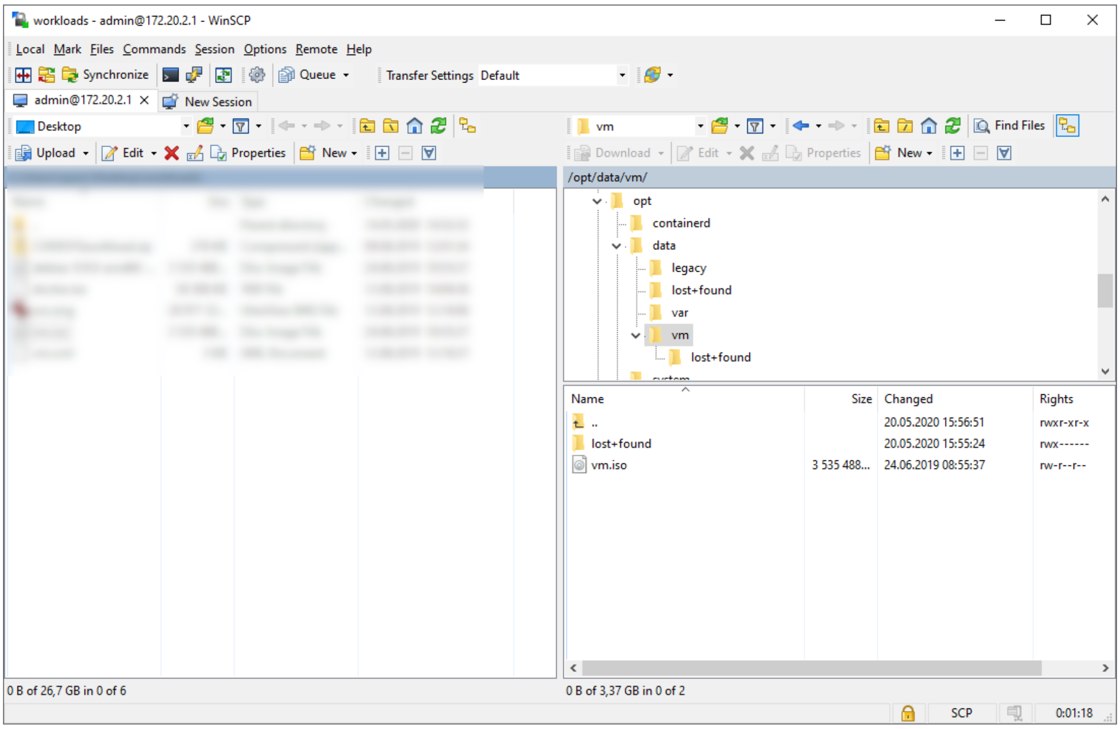
Port number: 22

User name: admin

Password: ••••••••

☑ Show Login dialog on startup and when the last session is closed

4. Navigate to the **/opt/data/** directory on the right side of the WinSCP window. It is located in the **root** directory.



workloads - admin@172.20.2.1 - WinSCP

/opt/data/vm/

Name	Size	Changed	Rights
..		20.05.2020 15:56:51	rwxr-xr-x
lost+found		20.05.2020 15:55:24	rwx------
vm.iso	3 535 488...	24.06.2019 08:55:37	rw-r--r--

0 B of 26,7 GB in 0 of 6

0 B of 3,37 GB in 0 of 2

SCP 0:01:18

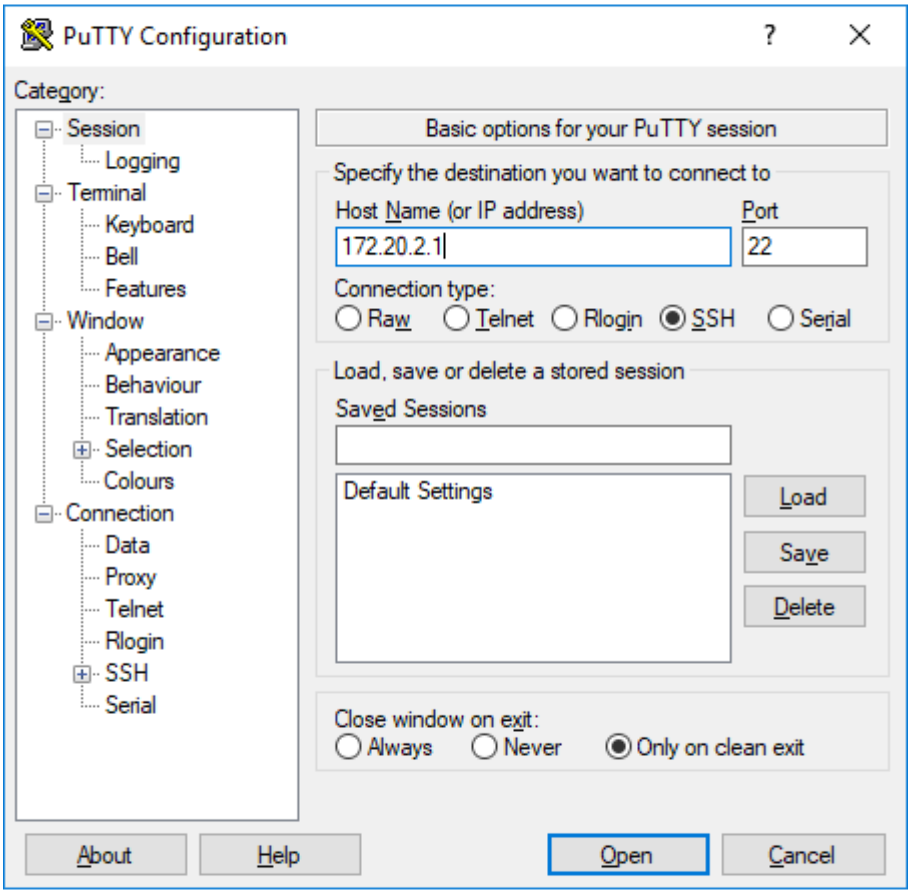
5. Copy the ISO file of the OS that is to be installed on the virtual machine to the directory on the Nerve Device.

Creating a virtual machine on a node

Using the Virtual Machine Manager is recommended to create a virtual machine and install the OS from the ISO file. Note that the virtual machine in this chapter is a fresh installation and will be generated on the Nerve Device directly.

1. Run Xming or an alternative.
2. Open an SSH client like PuTTY.

3. Enter the IP address for host access to the Nerve Device under **Host Name (or IP address)** to log in to the host of the Nerve Device.



Host Name (or IP address): 172.20.2.1

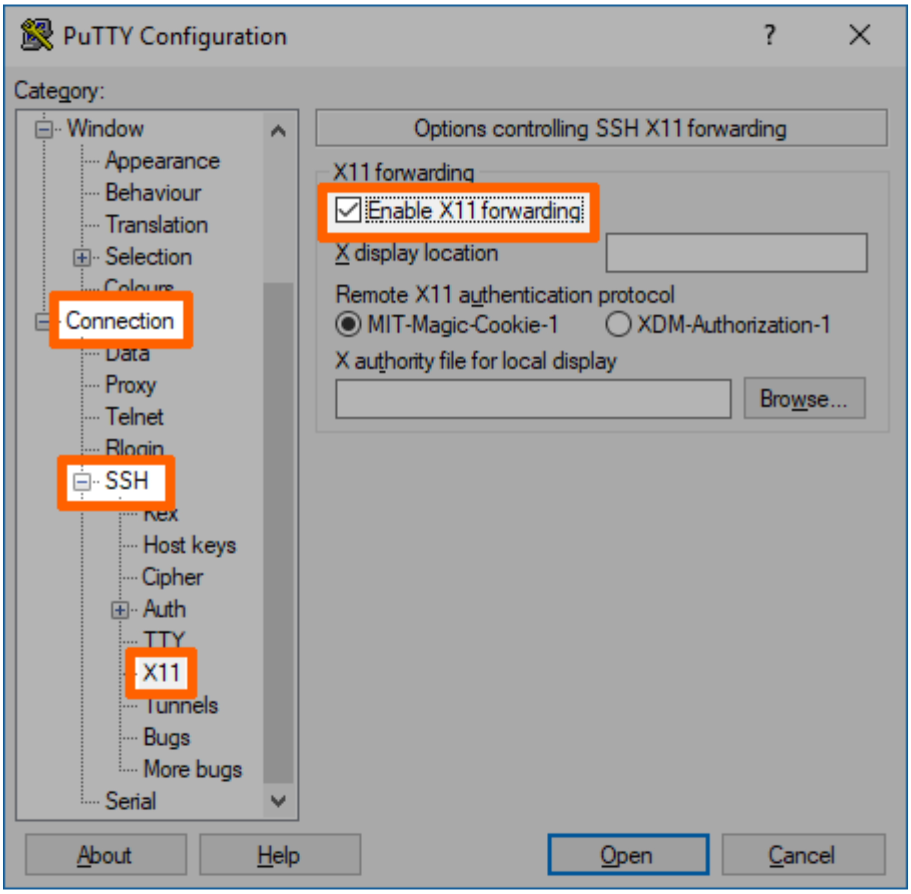
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

4. Expand **Connection > SSH > X11** on the left side.

5. Tick the checkbox next to **Enable X11 forwarding**.



Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

X display location

Remote X11 authentication protocol

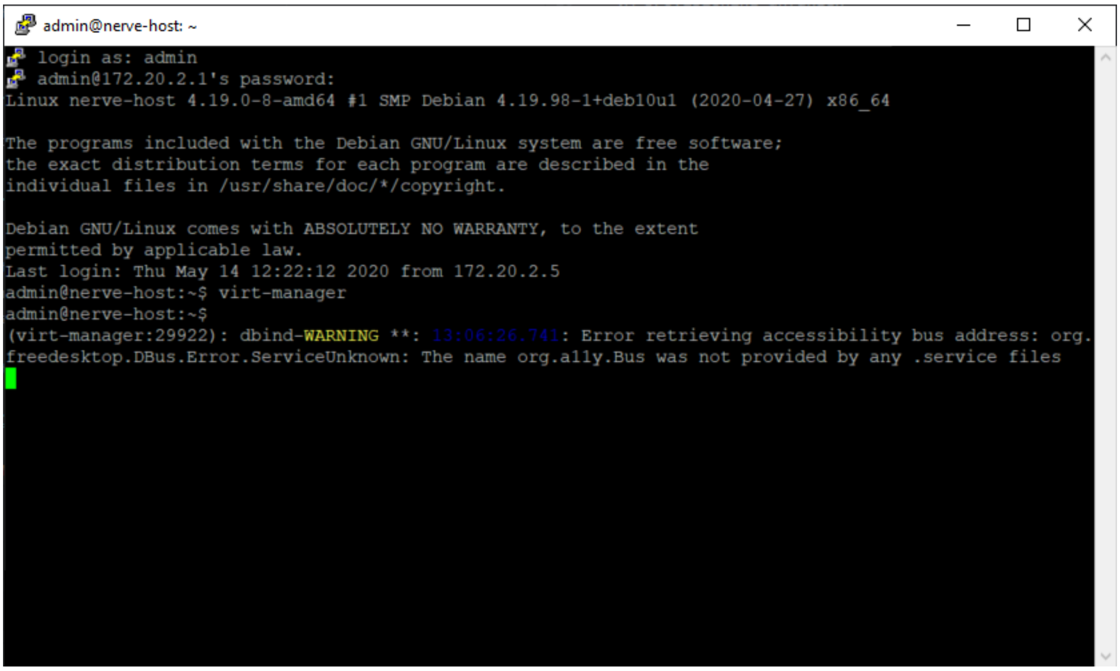
● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display

6. Click **Open**.

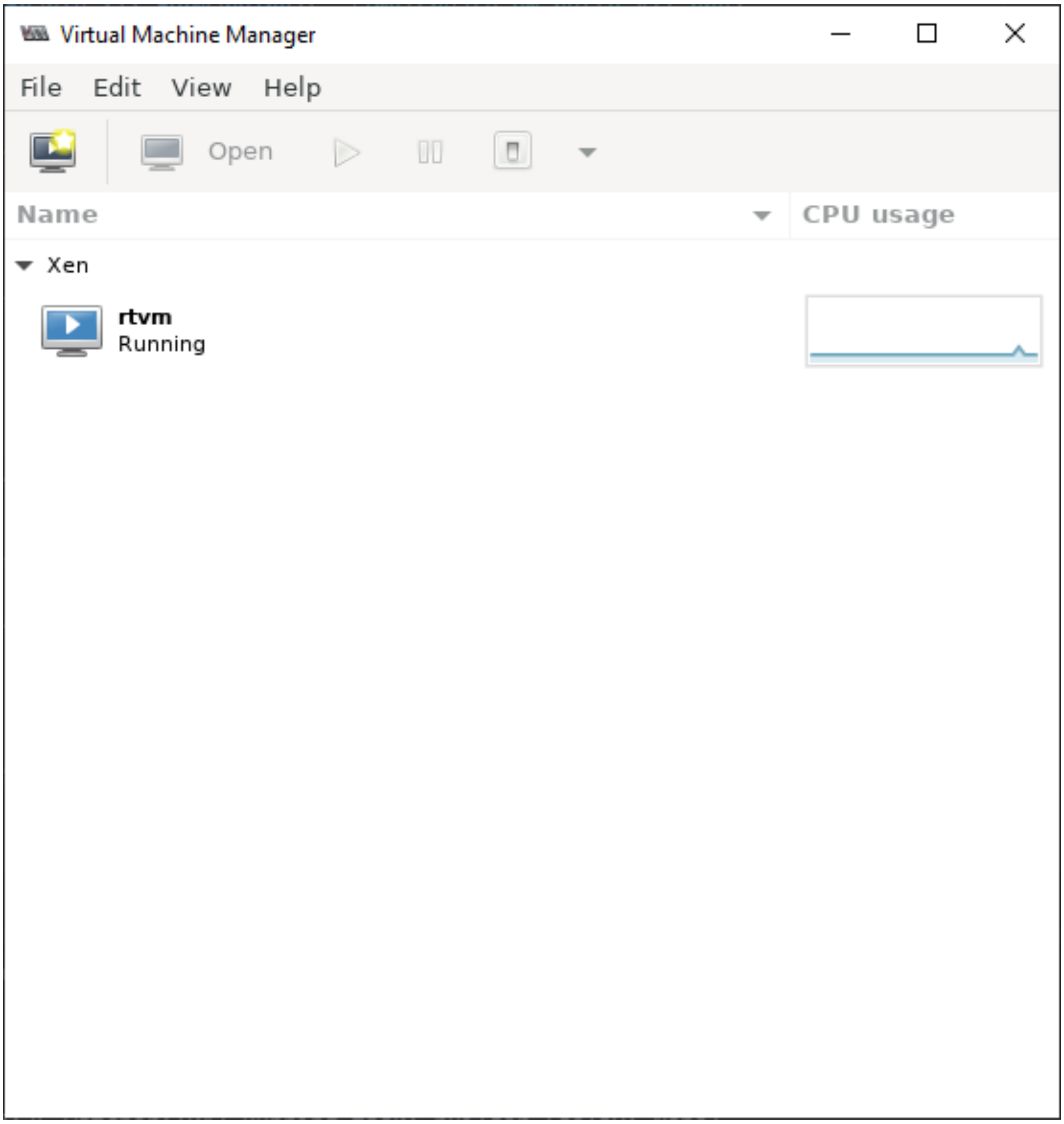
7. Log in with the credentials for host access to the Nerve Device.

8. Enter `virt-manager`.



```
login as: admin
admin@172.20.2.1's password:
Linux nerve-host 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1+deb10u1 (2020-04-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 14 12:22:12 2020 from 172.20.2.5
admin@nerve-host:~$ virt-manager
admin@nerve-host:~$
(virt-manager:29922): dbind-WARNING **: 13:06:26.741: Error retrieving accessibility bus address: org.
freedesktop.DBus.Error.ServiceUnknown: The name org.a11y.Bus was not provided by any .service files
```

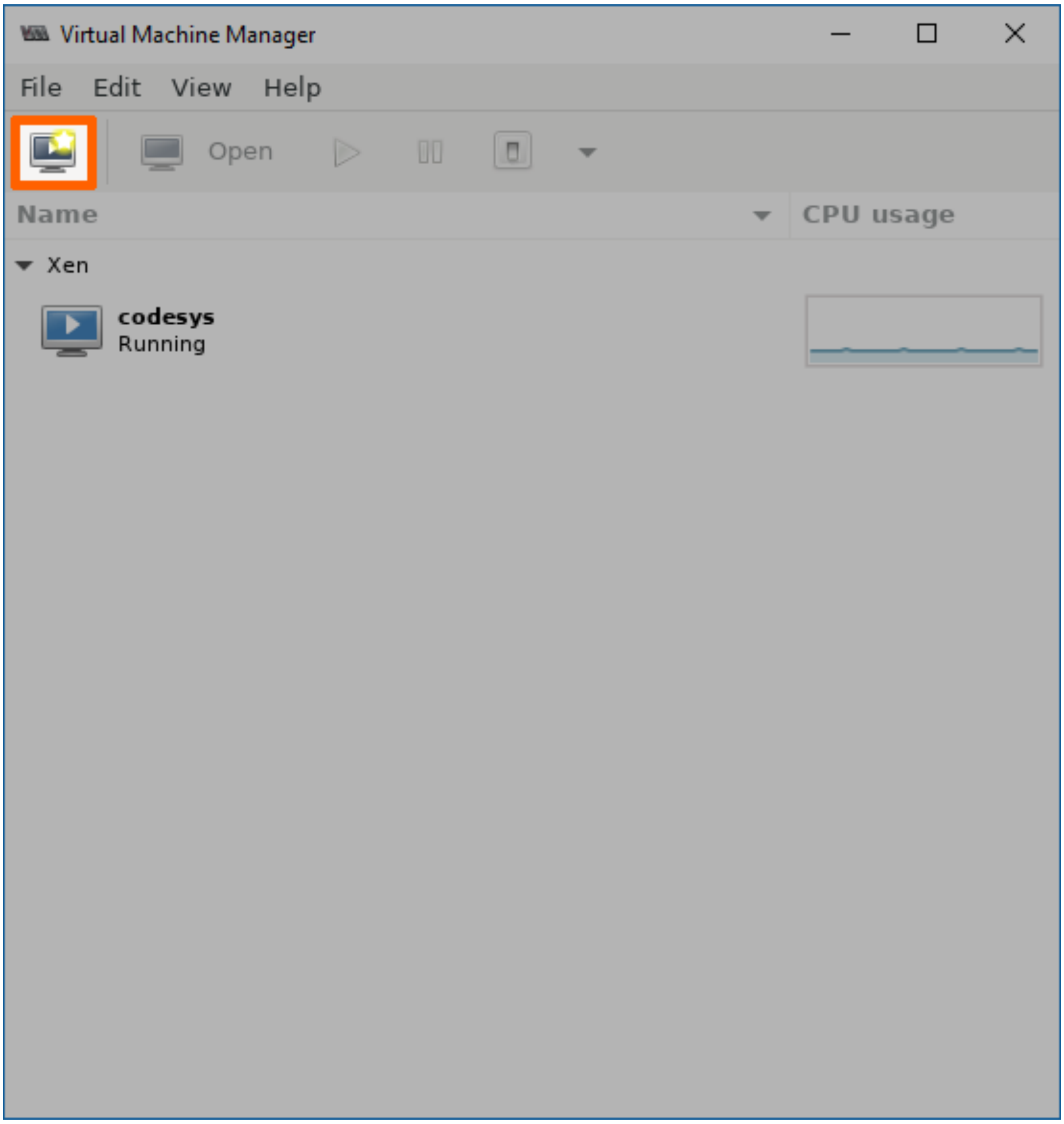
The interface of the Virtual Machine Manager will open.



Inserting the installation file (ISO)

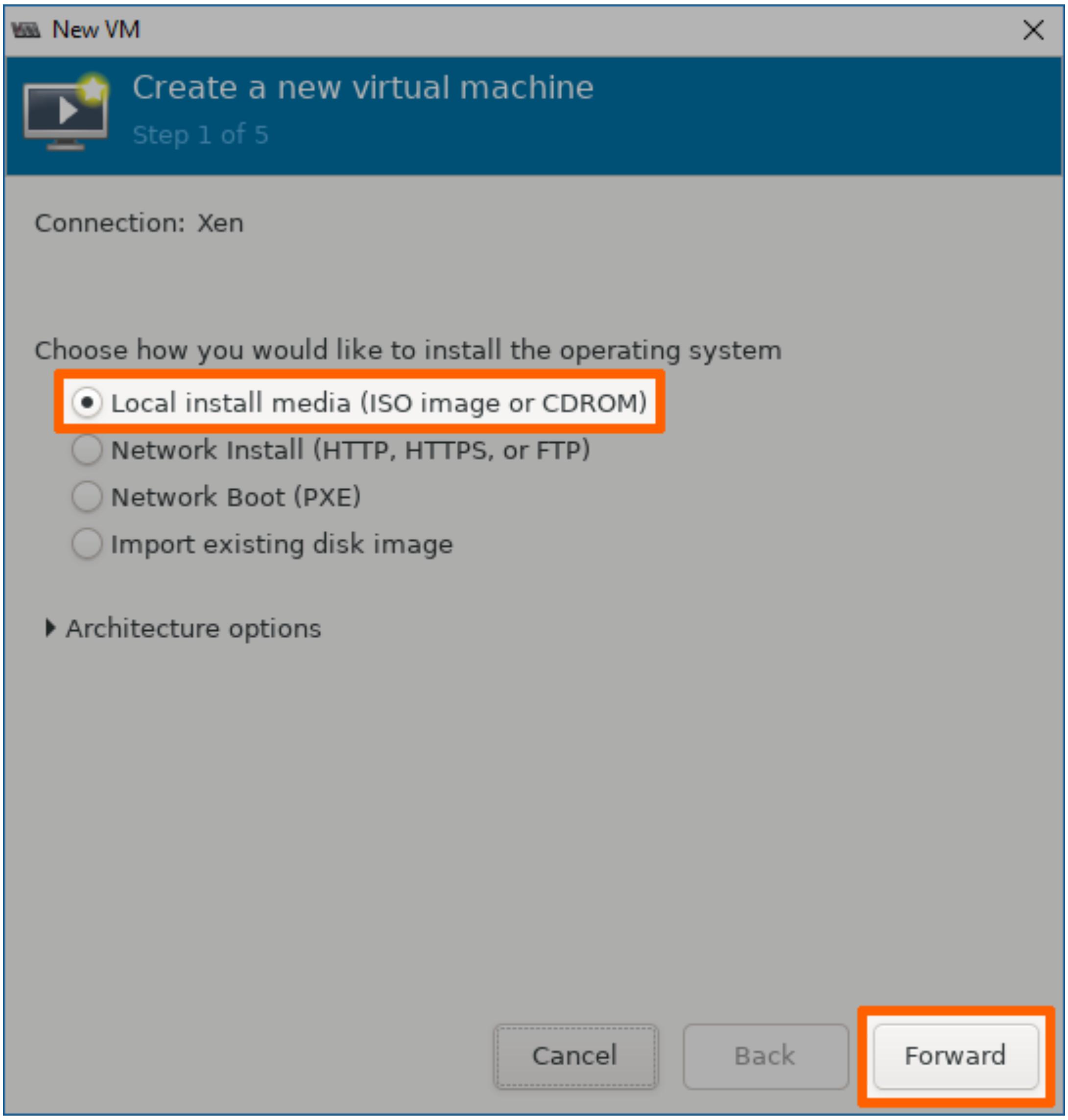
The creation of the virtual machine can now be initiated with the installation of the OS following right after. Note that the Virtual Machine Manager requires the virtual insertion of the ISO file in the beginning while resources for the virtual machine are defined later.

1. Select **File > New Virtual Machine** or click the symbol.



2. Select **Local install media (ISO image or CDROM)**.

3. Click **Forward**.



Connection: Xen

Choose how you would like to install the operating system

● Local install media (ISO image or CDROM)

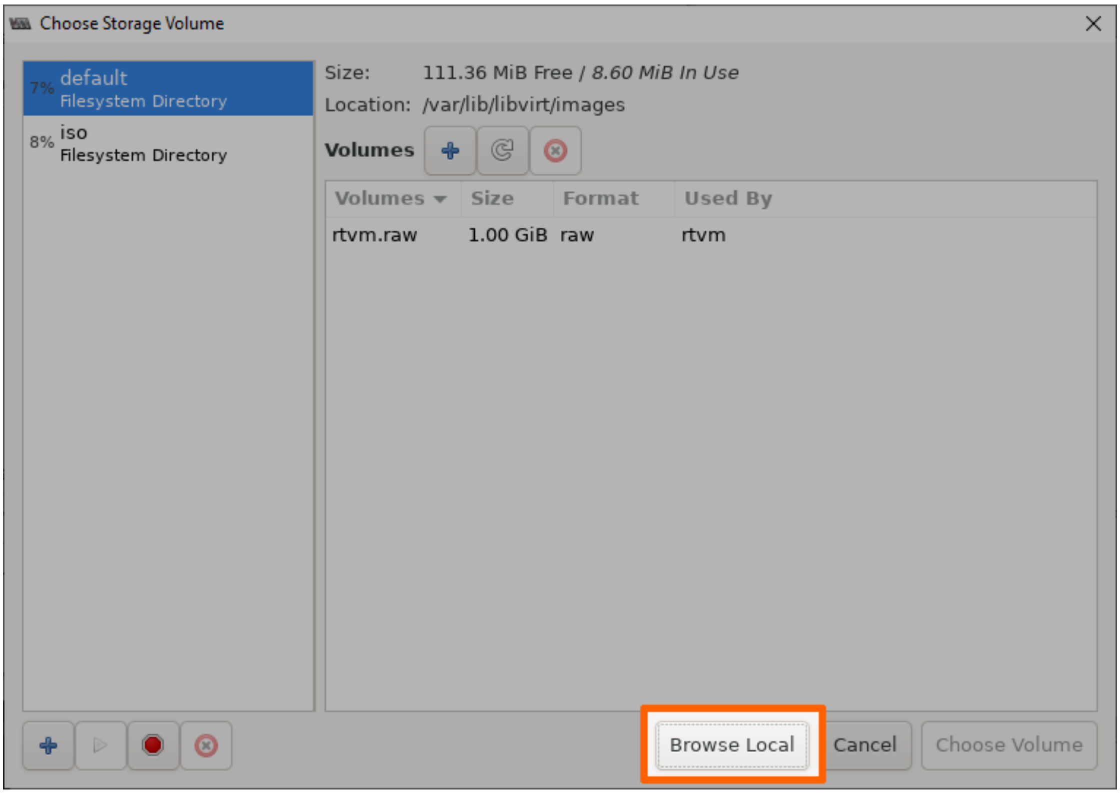
○ Network Install (HTTP, HTTPS, or FTP)

○ Network Boot (PXE)

○ Import existing disk image

4. Click **Browse...** in the next window.

5. In the new window select **Browse Local**.



Size: 111.36 MiB Free / *8.60 MiB In Use*

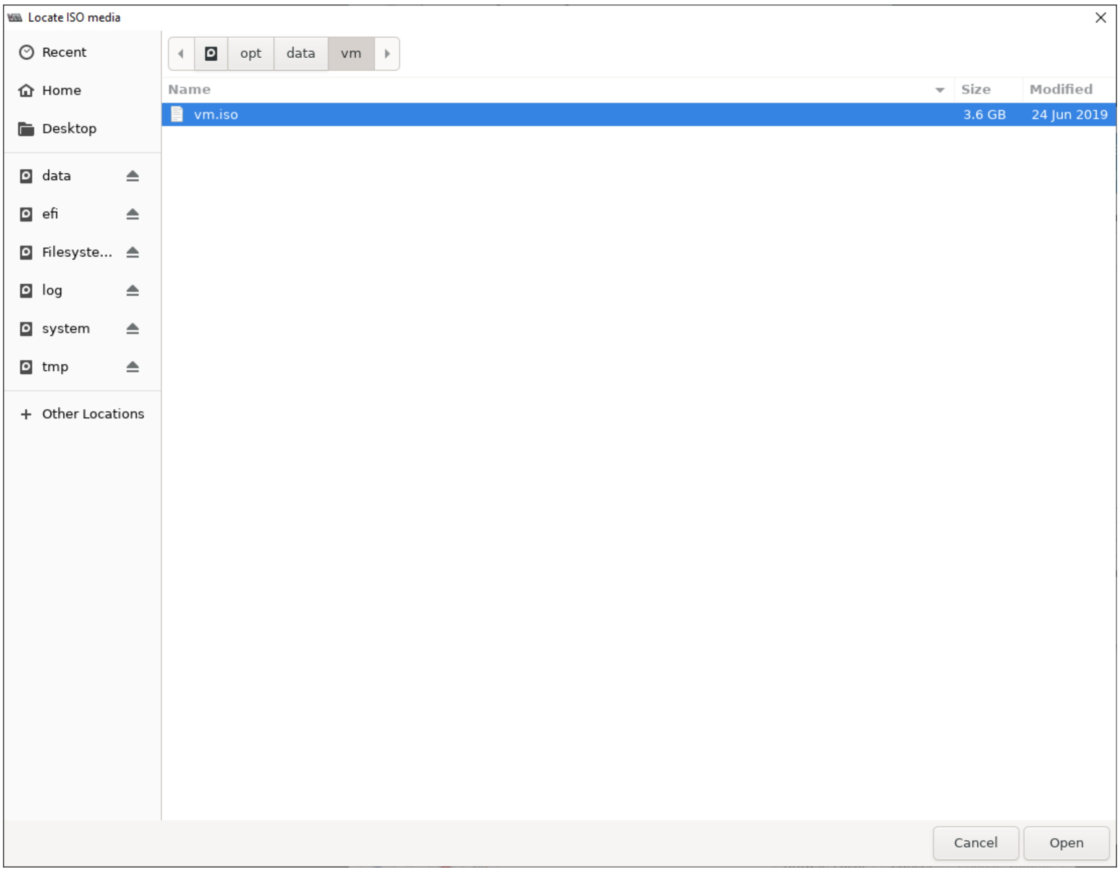
Location: /var/lib/libvirt/images

Volumes	Size	Format	Used By
rtvm.raw	1.00 GiB	raw	rtvm

6. Navigate to **opt/data/** by clicking the left arrow next to **admin**. The **opt** directory is located in the root directory.

7. Double-click the directory containing the ISO file.

8. Select the ISO file of the OS that was copied before.



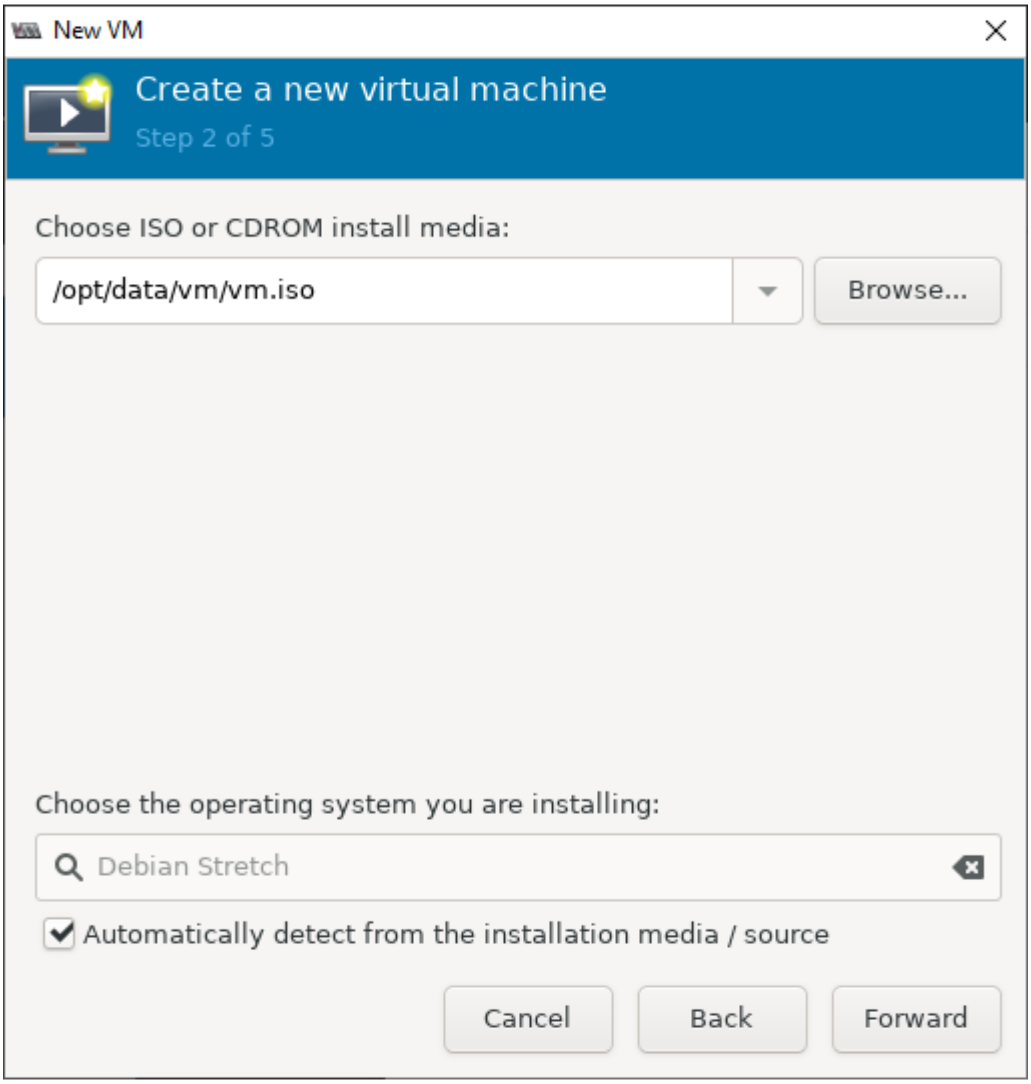
vm.iso 3.6 GB 24 Jun 2019

9. Click **Open**.

Defining resources for the virtual machine

Next, the amount of memory and the number of CPUs need to be defined, and the logical volume assigned to the virtual machine needs to be selected.

1. Click **Forward**.



New VM

Create a new virtual machine

Step 2 of 5

Choose ISO or CDROM install media:

/opt/data/vm/vm.iso

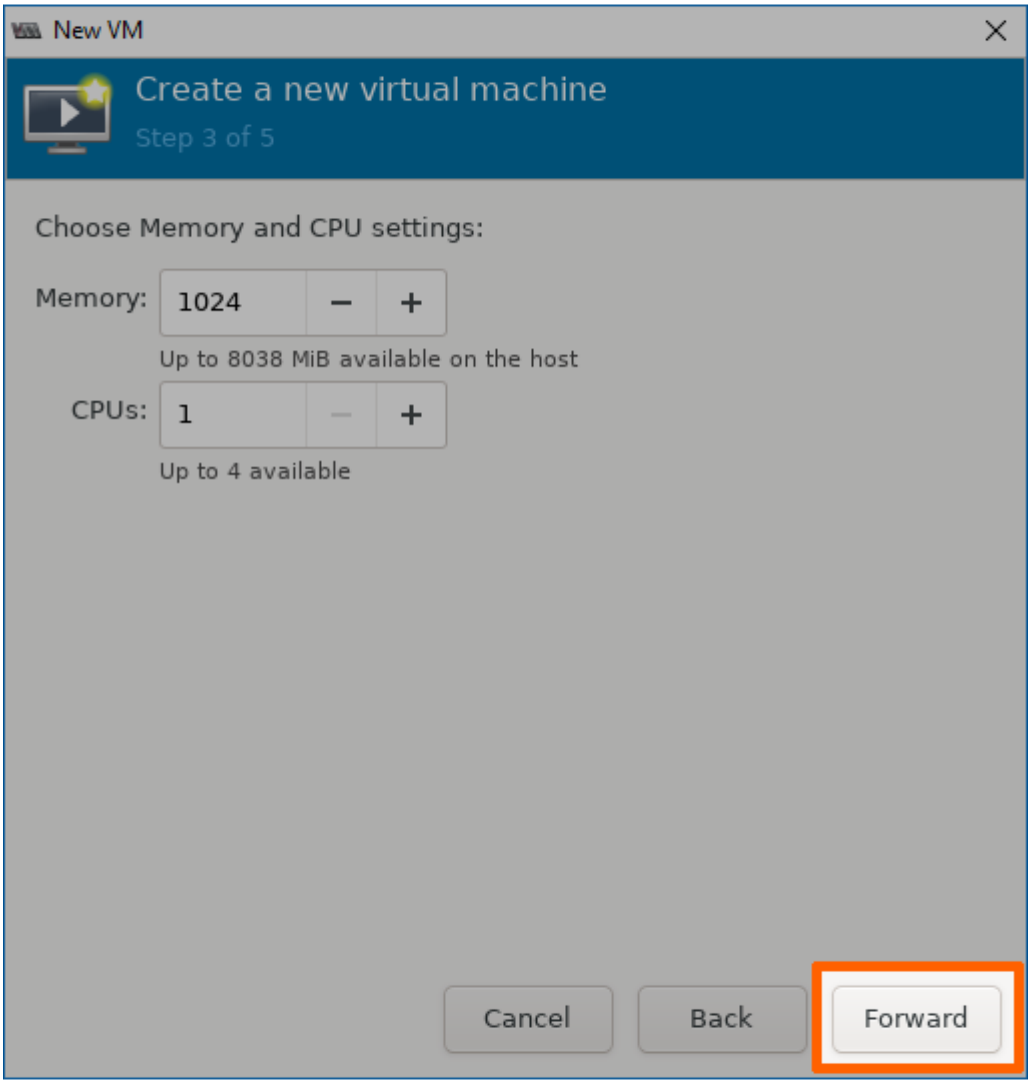
Choose the operating system you are installing:

Debian Stretch

☑ Automatically detect from the installation media / source

2. Define how much memory and how many CPUs to assign to this virtual machine.

3. Select **Forward**.



Choose Memory and CPU settings:

Memory: 1024

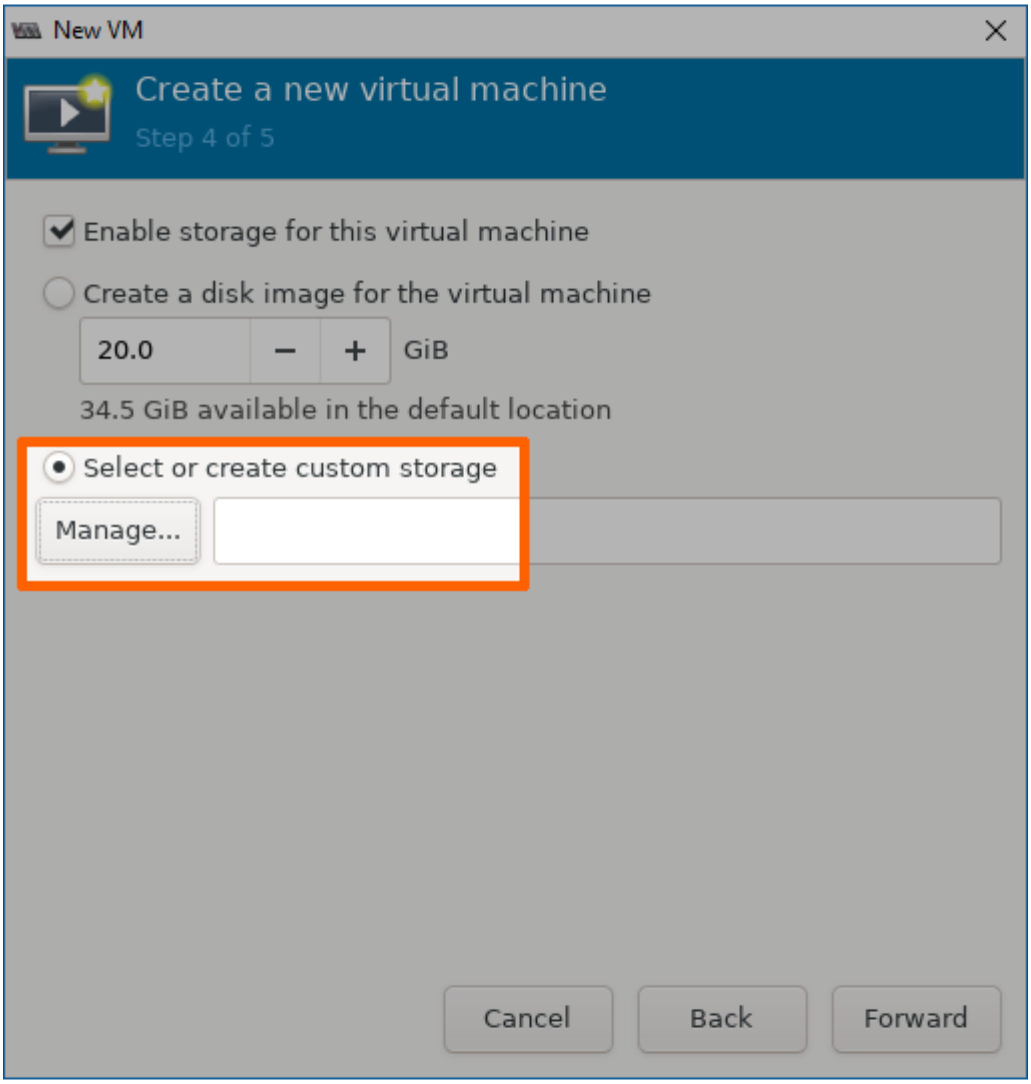
Up to 8038 MiB available on the host

CPUs: 1

Up to 4 available

4. Click **Select or create custom storage**.

5. Select **Manage...**.



New VM

Create a new virtual machine

Step 4 of 5

☑ Enable storage for this virtual machine

○ Create a disk image for the virtual machine

20.0 GiB

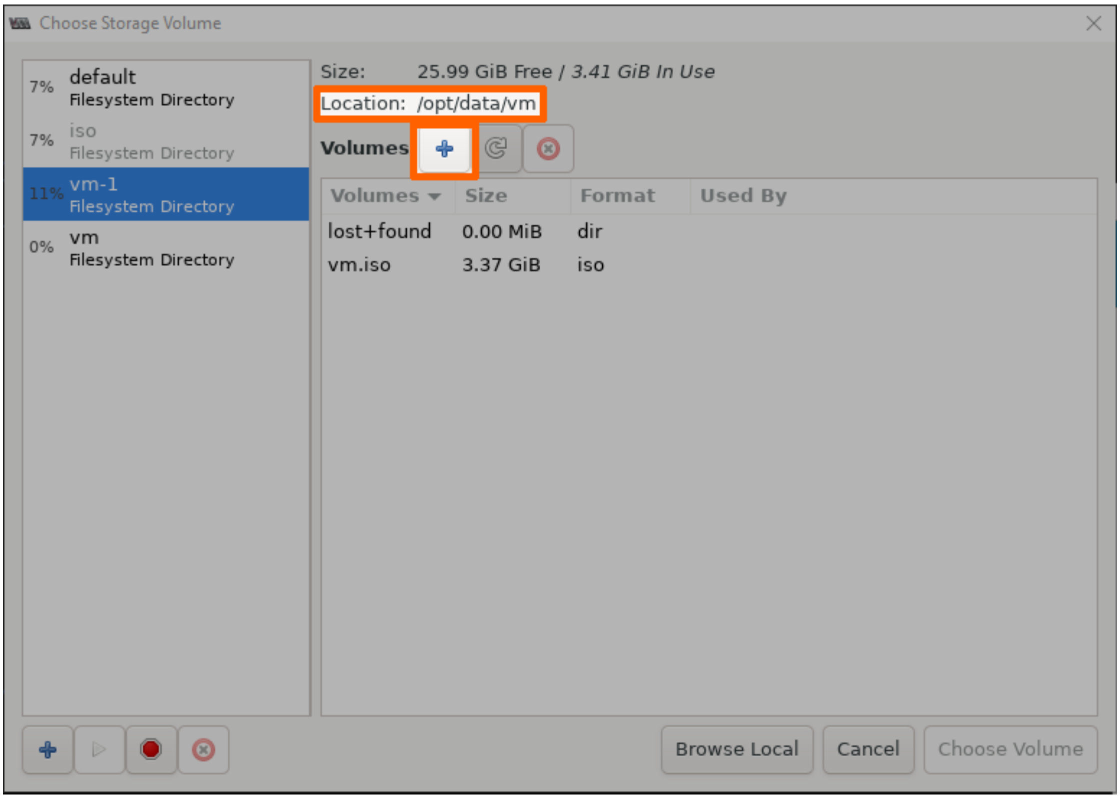
34.5 GiB available in the default location

● Select or create custom storage

Installing the operating system

Now the virtual machine will be initiated and the installation of the OS will be started.

1. Select the pool that was created in the command line on the left side in the **Choose Storage Volume** window. Make sure it is the volume with the location **opt/data/**.
2. Select the plus symbol in the middle of the screen.

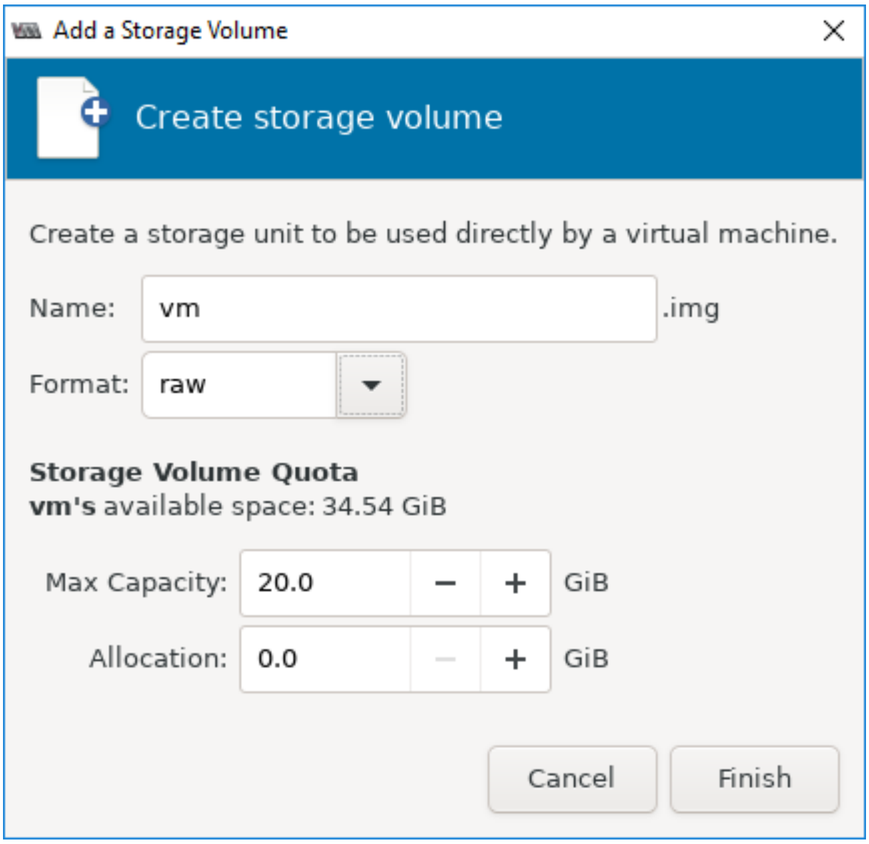


Size: 25.99 GiB Free / *3.41 GiB In Use*

Location: /opt/data/vm

Volumes	Size	Format	Used By
lost+found	0.00 MiB	dir	
vm.iso	3.37 GiB	iso	

3. Enter a name for the IMG file.



Create a storage unit to be used directly by a virtual machine.

Name: vm .img

Format: raw

Storage Volume Quota

vm's available space: 34.54 GiB

Max Capacity: 20.0 GiB

Allocation: 0.0 GiB

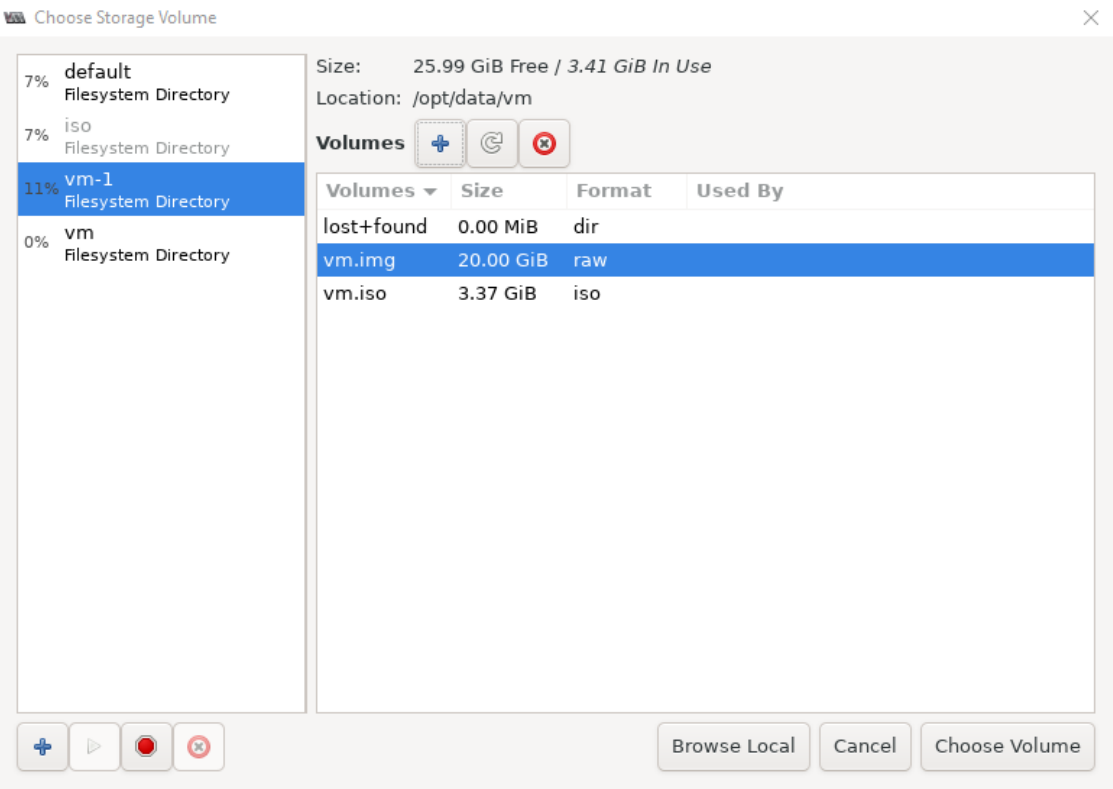
NOTE

Make sure that the value entered for **Max Capacity** is a factor of 512 Bytes. Any value that is not a factor of 512 B will cause an error when the virtual machine workload is deployed. In other words, do not enter any decimal value except `.0` or `.5`.

4. Select **Finish**.

5. Select the IMG file from the list in the middle.

6. Click **Choose Volume**.



Choose Storage Volume

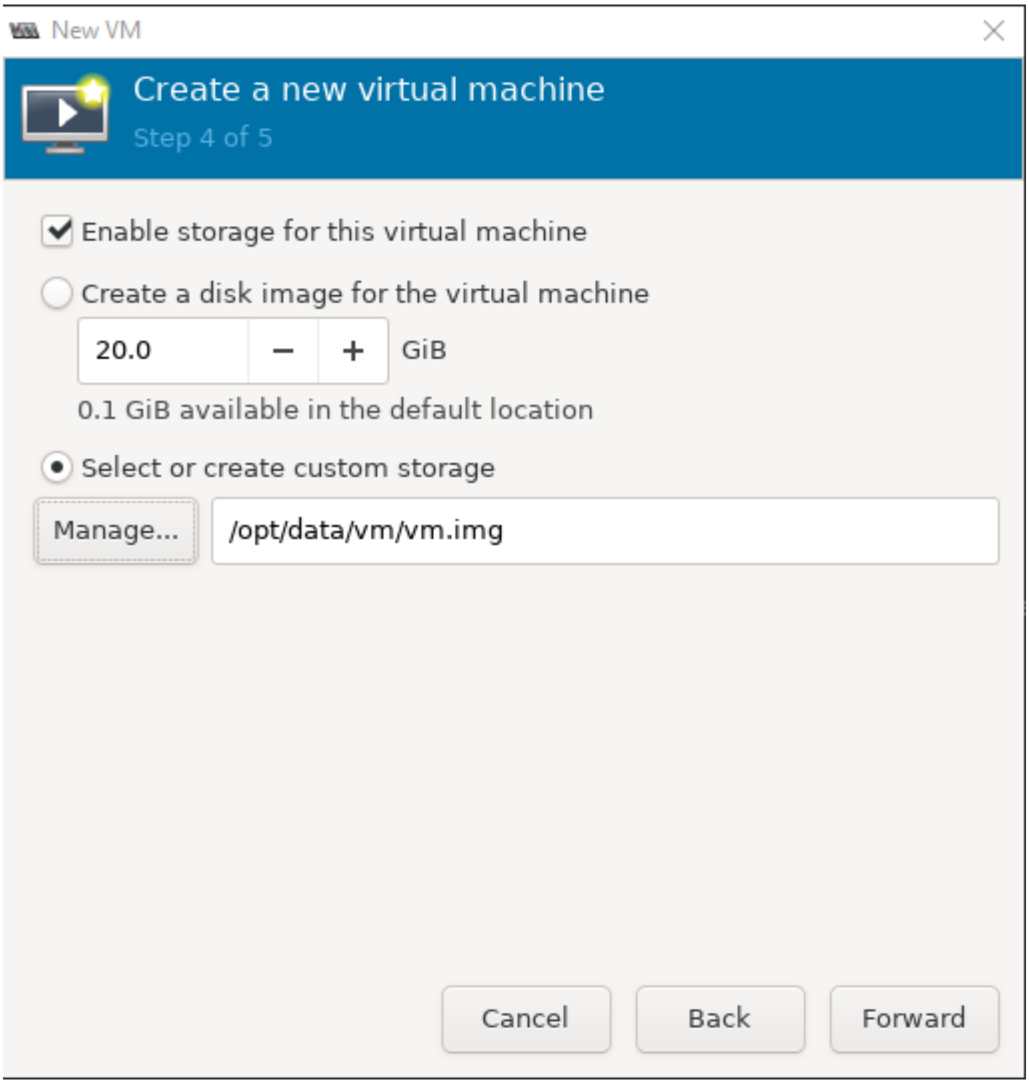
7%	default Filesystem Directory
7%	iso Filesystem Directory
11%	vm-1 Filesystem Directory
0%	vm Filesystem Directory

Size: 25.99 GiB Free / *3.41 GiB In Use*

Location: /opt/data/vm

Volumes	Size	Format	Used By
lost+found	0.00 MiB	dir	
vm.img	20.00 GiB	raw	
vm.iso	3.37 GiB	iso	

7. Click **Forward** to initialize the installation of the OS on the virtual machine.



☑ Enable storage for this virtual machine

○ Create a disk image for the virtual machine

20.0 GiB

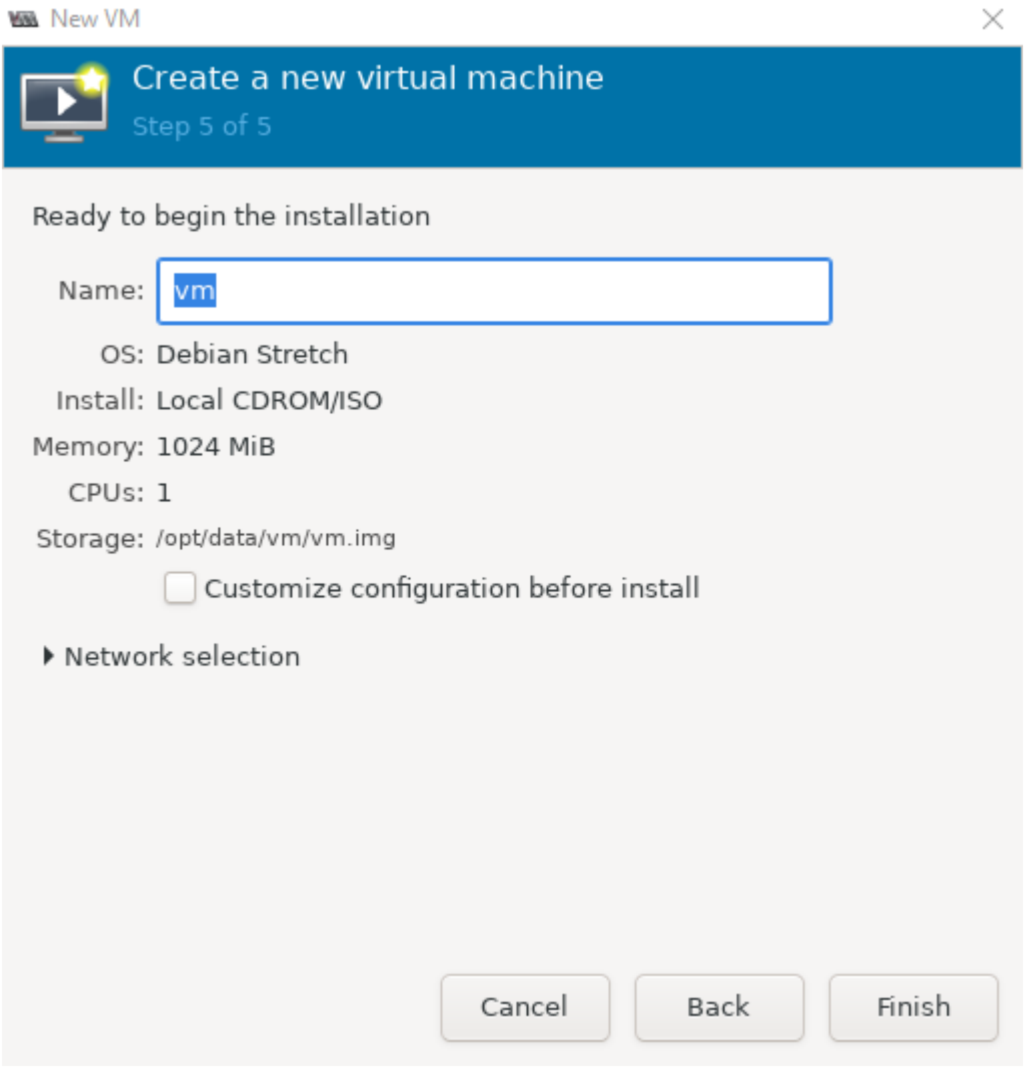
0.1 GiB available in the default location

● Select or create custom storage

Manage... /opt/data/vm/vm.img

8. Enter a name for the virtual machine.

9. Click **Finish**. The virtual machine will be initiated and the installation of the OS will be started.



Ready to begin the installation

Name: vm

OS: Debian Stretch

Install: Local CDROM/ISO

Memory: 1024 MiB

CPUs: 1

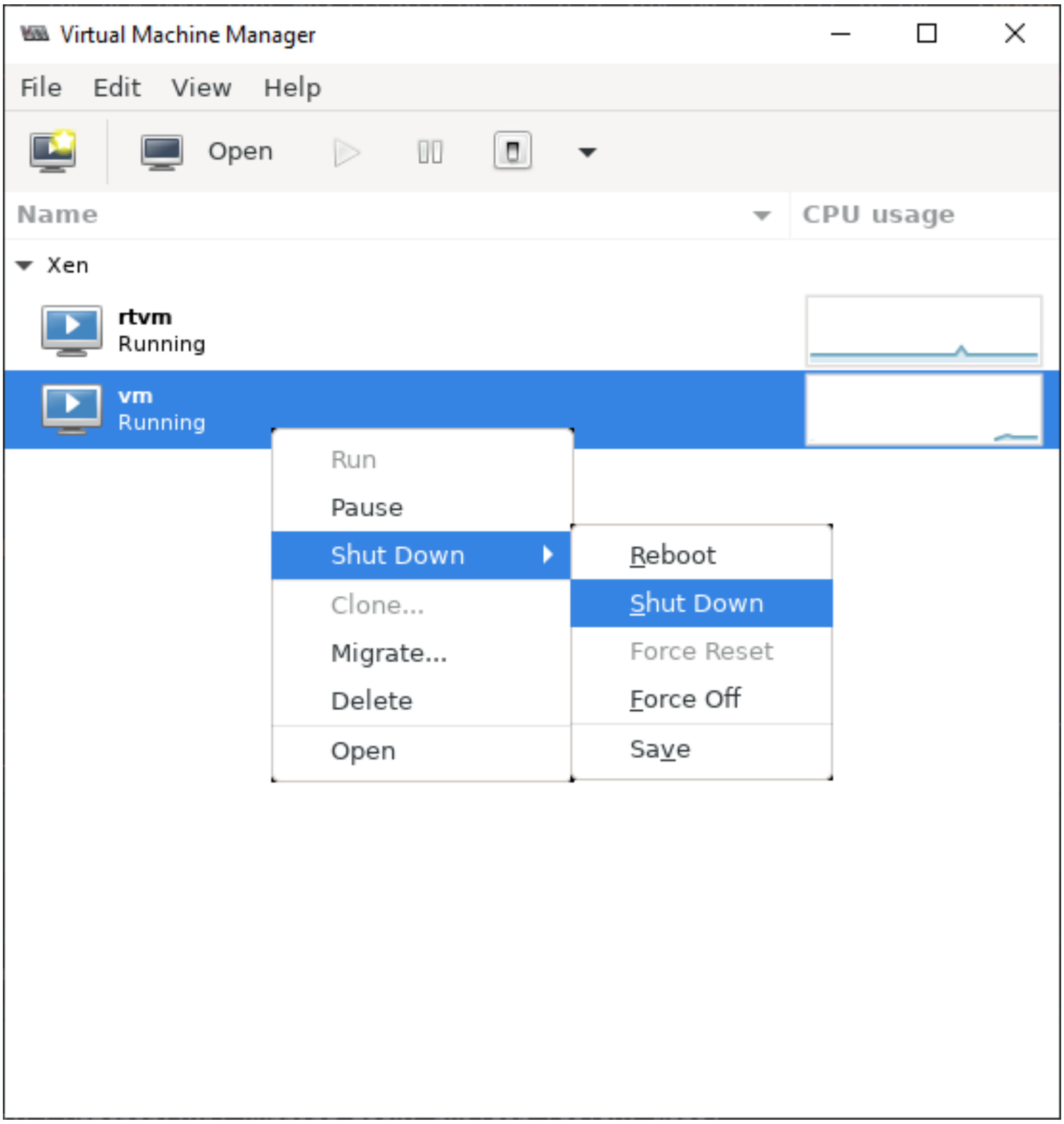
Storage: /opt/data/vm/vm.img

□ Customize configuration before install

10. Complete the installation of the OS. Follow the instructions provided by the vendor.

11. After the installation is completed, right-click the virtual machine in the main Virtual Machine Manager window.

12. Select **Shut Down > Shut Down** to shut down the VM.



Copying the IMG file to a local workstation

With the generation of the VM, the IMG file of the VM has also been generated on the Nerve Device. Copy the IMG file to the local workstation.

1. Switch to the SSH client window.

2. Enter the following command:

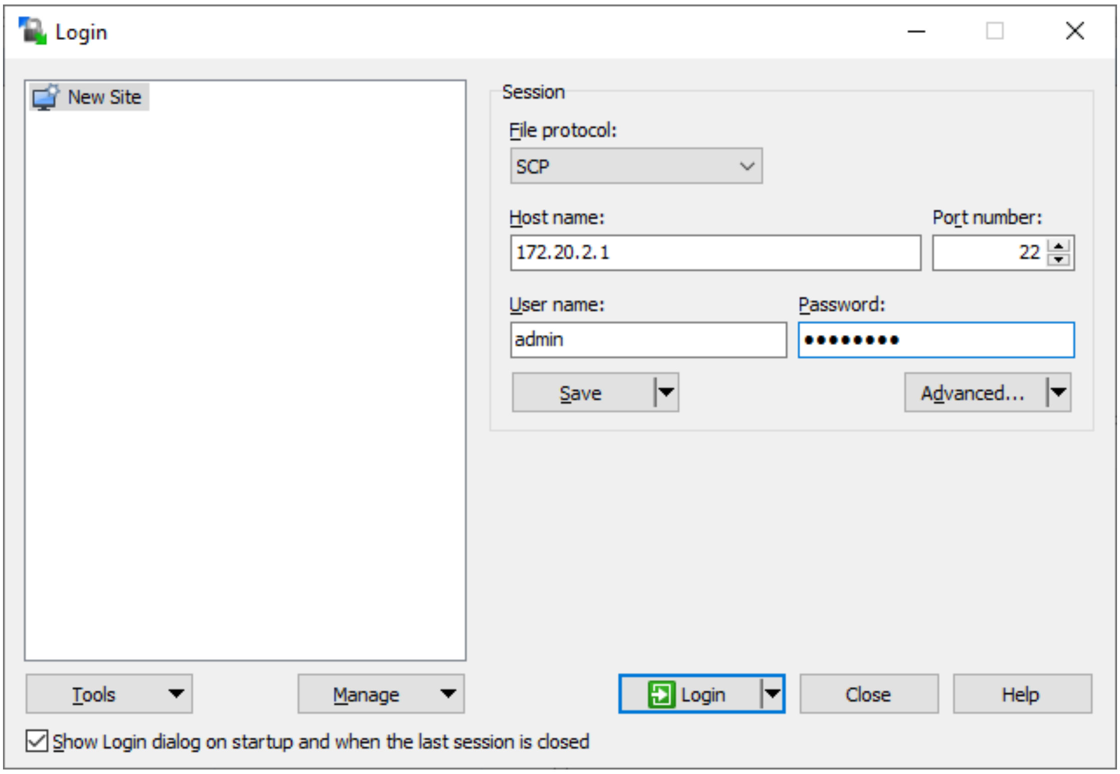
```
sudo chmod o+r /opt/data/<directory>/<vmname>.img
```

> **NOTE**
>
> The IMG file is called `<vmname>.img` in the command example above. Replace the placeholder name of the image in the command with the actual name of the IMG file.

3. Open a file transfer client like WinSCP.

4. Enter the IP address for host access to the Nerve Device under **Host Name**.

5. Enter the credentials for host access to the Nerve Device below under **User name** and **Password**.



Login

Session

File protocol: SCP

Host name: 172.20.2.1

Port number: 22

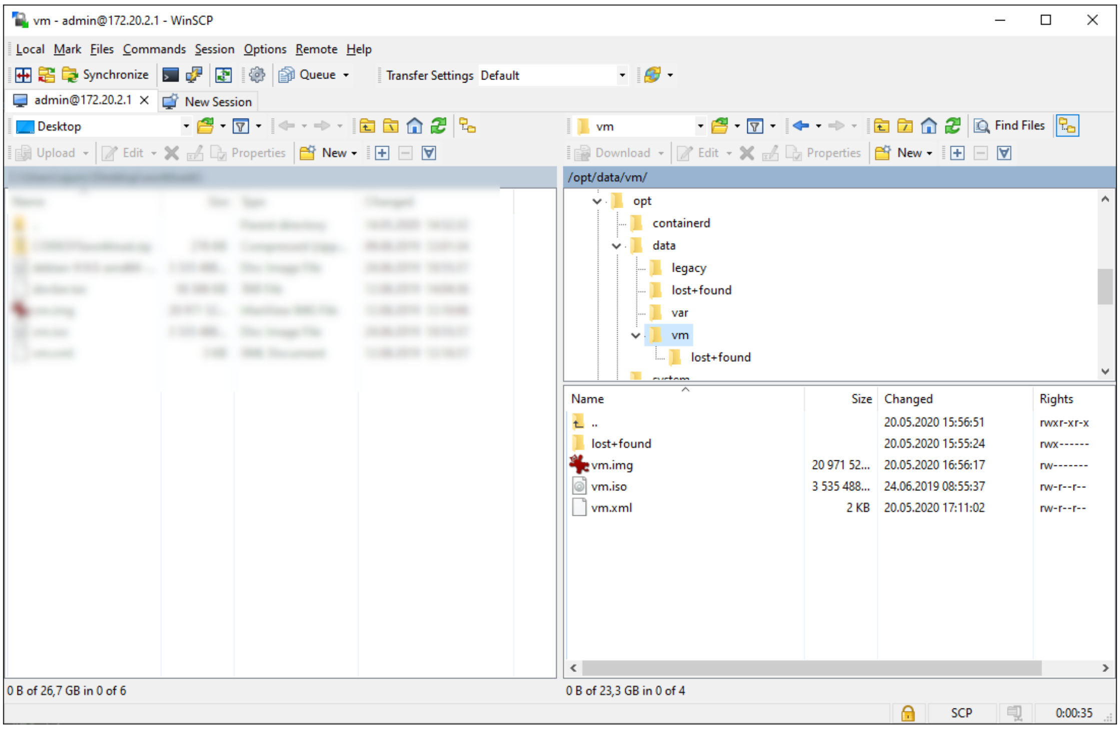
User name: admin

Password: ••••••••

☑ Show Login dialog on startup and when the last session is closed

6. Navigate to `/opt/data/<directory>` on the right side of the window. The **opt** directory is located in the **root** directory.

7. Copy the `<vmname>.img` file to the local workstation.



vm - admin@172.20.2.1 - WinSCP

/opt/data/vm/

Name	Size	Changed	Rights
..		20.05.2020 15:56:51	rwxr-xr-x
lost+found		20.05.2020 15:55:24	rwx------
vm.img	20 971 52...	20.05.2020 16:56:17	rw-------
vm.iso	3 535 488...	24.06.2019 08:55:37	rw-r--r--
vm.xml	2 KB	20.05.2020 17:11:02	rw-r--r--

0 B of 26,7 GB in 0 of 6

0 B of 23,3 GB in 0 of 4

The virtual machine has now been generated on the Nerve Device and the IMG file of the virtual machine is now on the local workstation.

NOTE

Do not deploy the virtual machine from the process above to the same Nerve Device through the Management System. The virtual machine will be present twice. The deployment of the virtual machine from the process above should be done to different nodes.

Obtaining the XML file

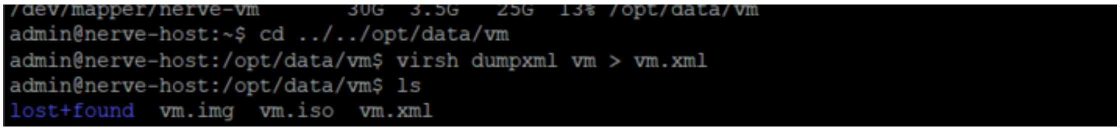
When the IMG file was generated on the Nerve Device, an XML file for the IMG file was generated as well. It also has to be obtained manually. However, before it can be copied to the workstation. It has to be transferred out of the **etc** directory.

1. Switch to the SSH client window.
2. Navigate to `/opt/data/<directory>` by entering the following command:

   ```
   cd ../../opt/data/<directory>
   ```

3. Enter the following command to dump the XML file into the directory:

   ```
   virsh dumpxml <volume_name> > <filename>.xml
   ```

   ```
   /dev/mapper/nerve-vm       30G  3.5G   25G  13% /opt/data/vm
   admin@nerve-host:~$ cd ../../opt/data/vm
   admin@nerve-host:/opt/data/vm$ virsh dumpxml vm > vm.xml
   admin@nerve-host:/opt/data/vm$ ls
   lost+found  vm.img  vm.iso  vm.xml
   ```

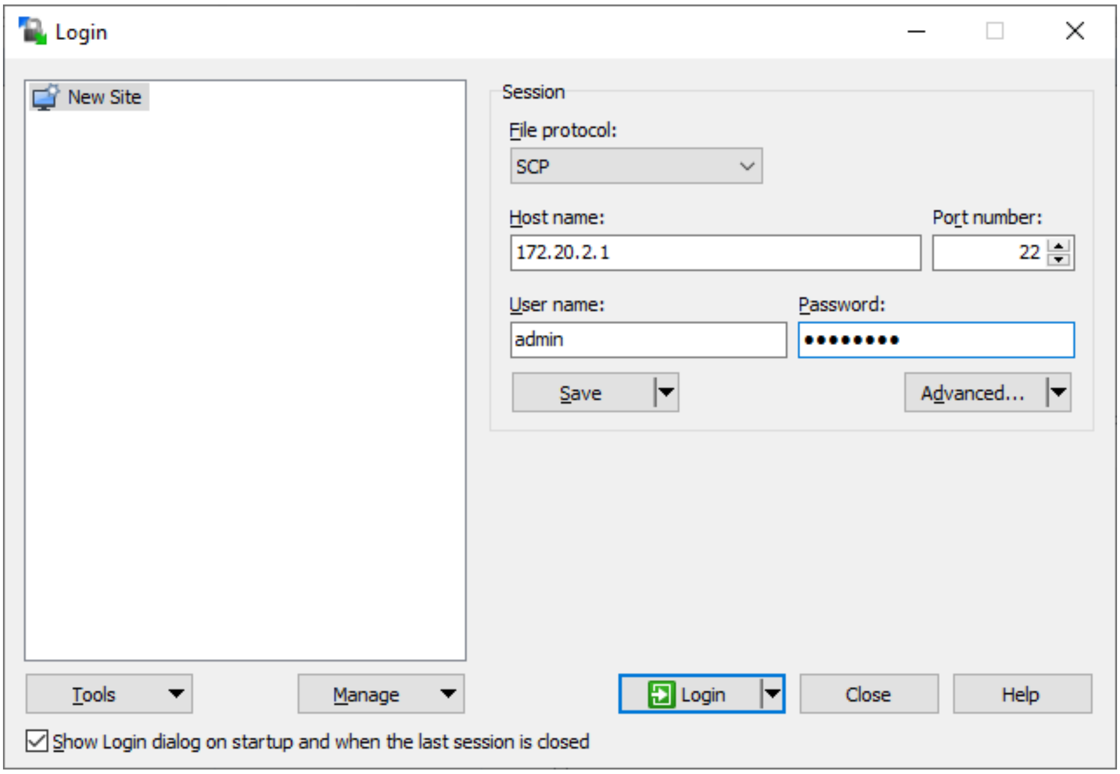
4. Enter the following command:

   ```
   sudo chmod o+r /opt/data/<directory>/<filename>.xml
   ```

   NOTE

   Replace the placeholder name of the XML file in the command with the actual name of the XML file. The XML file is called `vm.xml` in the following screenshots below.

5. Open a file transfer client like WinSCP.
6. Enter the IP address for host access to the Nerve Device under **Host Name**.
7. Enter the credentials for host access to the Nerve Device below under **User name** and **Password**.



File protocol: SCP

Host name: 172.20.2.1

Port number: 22

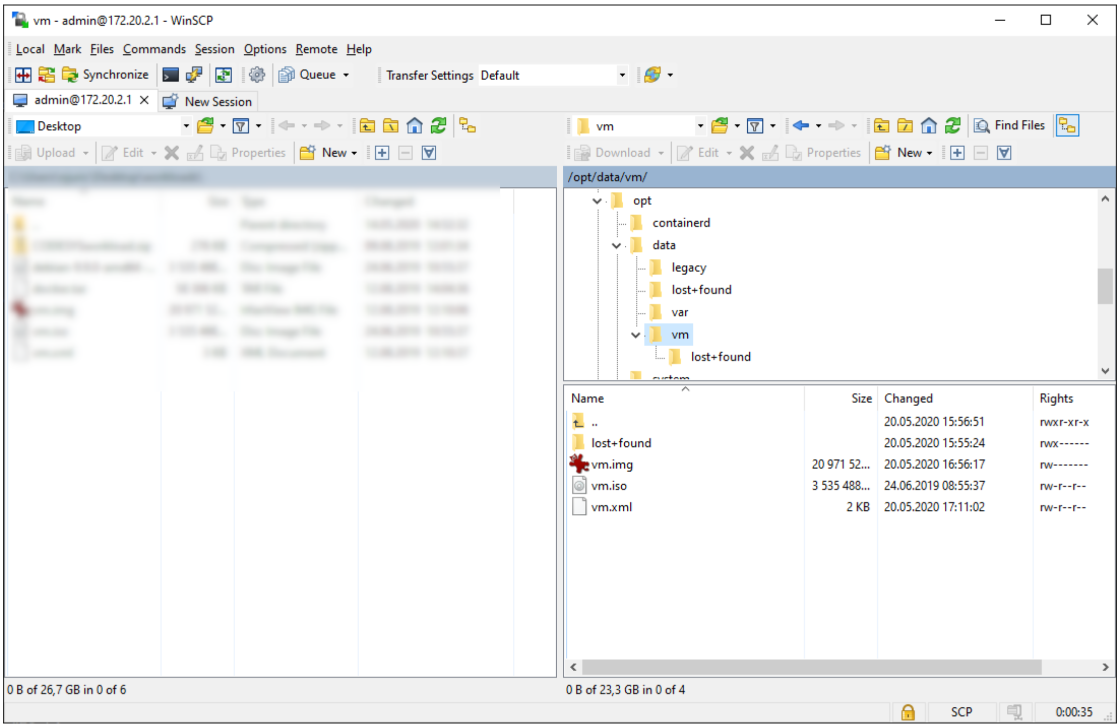
User name: admin

Password: ••••••••

☑ Show Login dialog on startup and when the last session is closed

8. Navigate to `/opt/data/<directory>` on the right side of the window. The **opt** directory is located in the **root** directory.

9. Copy the `<filename>.xml` file to the local workstation.



/opt/data/vm/

Name	Size	Changed	Rights
..		20.05.2020 15:56:51	rwxr-xr-x
lost+found		20.05.2020 15:55:24	rwx------
vm.img	20 971 52...	20.05.2020 16:56:17	rw-------
vm.iso	3 535 488...	24.06.2019 08:55:37	rw-r--r--
vm.xml	2 KB	20.05.2020 17:11:02	rw-r--r--

0 B of 26,7 GB in 0 of 6

0 B of 23,3 GB in 0 of 4

SCP 0:00:35

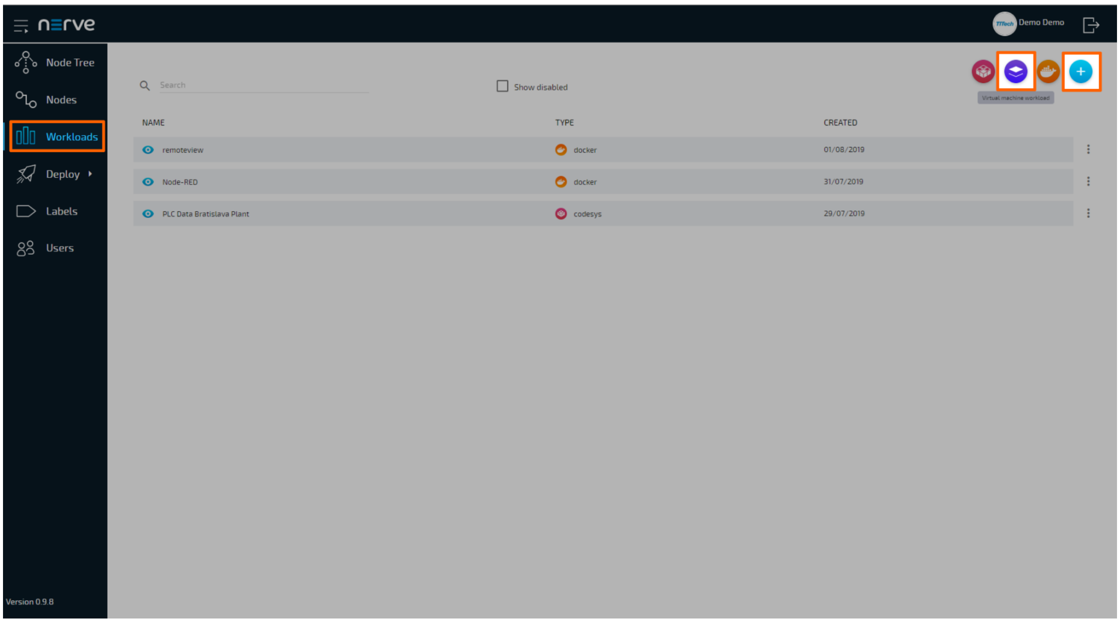
With this all the necessary files to provision a Virtual Machine workload for this virtual machine are ready.

Provisioning a Virtual machine workload

The following instructions cover the basic requirements for provisioning a Virtual Machine workload. Optional settings will be left out. Extended options are addressed in the last section of this chapter. There are two further types of workloads that can be provisioned: CODESYS workloads and Docker workloads. The process for each workload is highlighted in its respective chapter.

1. Log in to the Management System.
2. Select **Workloads** in the left-hand menu.
3. Select the plus symbol in the upper-right corner.

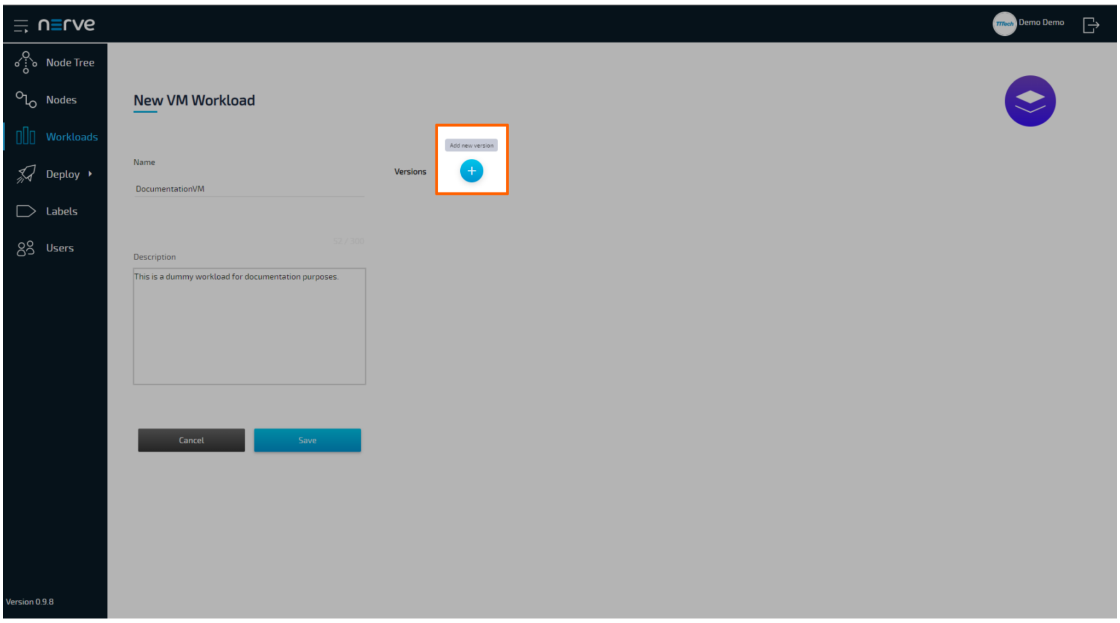
4. Select the virtual machine symbol (**Virtual machine workload**) in the middle of the three symbols that expanded.



NAME	TYPE	CREATED
remoteview	docker	01/08/2019
Node-RED	docker	31/07/2019
PLC Data Bratislava Plant	codesys	29/07/2019

5. In the new window, enter a name for the workload.

6. Select the plus symbol next to **Versions** to add a new version of the workload.



New VM Workload

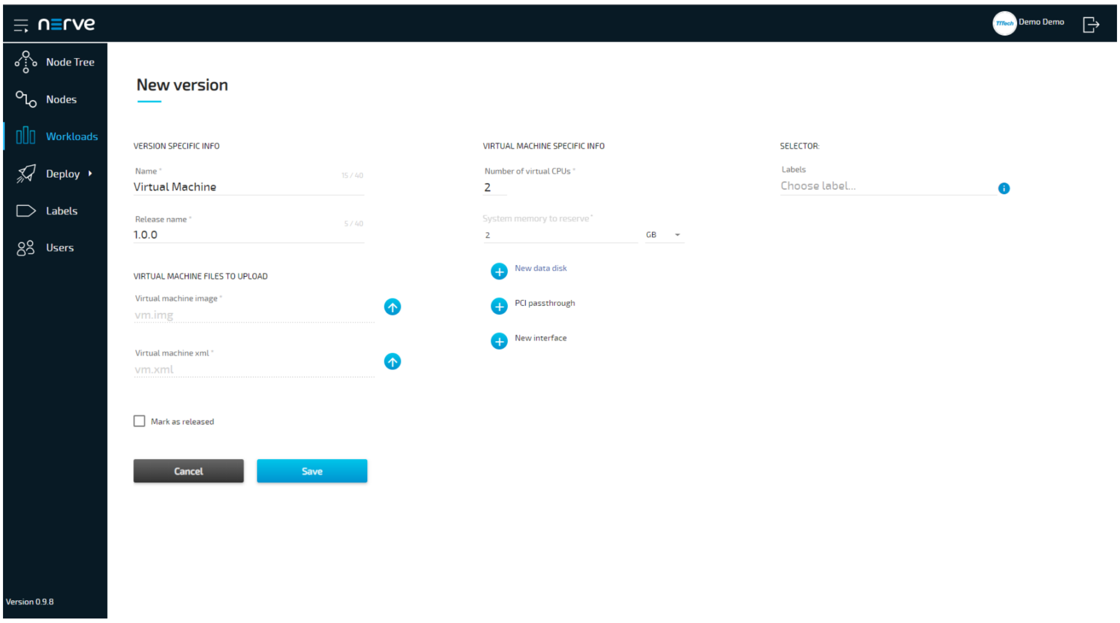
Name
DocumentationVM

52 / 300

Description
This is a dummy workload for documentation purposes.

7. In the next window, enter the following information:

Item	Description
Name	In the new window, enter a **Name** for the version of this workload.
Release name	Enter a **Release name** for the version of this workload.
Virtual machine image	Click the **upward arrow** symbol here to add the virtual machine image. The image has to be an IMG or RAW file.
Virtual machine xml	Click the **upward arrow** symbol here to add the virtual machine XML file.
Number of virtual CPUs	Enter the number of virtual CPUs to use for this virtual machine.
Limit memory to	Assign how much system memory the workload is allowed to use.



New version

VERSION SPECIFIC INFO

Name * Virtual Machine 15 / 40

Release name * 1.0.0 5 / 40

VIRTUAL MACHINE FILES TO UPLOAD

Virtual machine image * vm.img

Virtual machine xml * vm.xml

□ Mark as released

VIRTUAL MACHINE SPECIFIC INFO

Number of virtual CPUs * 2

System memory to reserve * 2 GB

New data disk

PCI passthrough

New interface

SELECTOR:

Labels Choose label...

8. Select **Save** in the lower-left corner.

The workload has now been provisioned and is ready to be deployed in the **Deploy** menu.

> **NOTE**
>
> While some settings are not required to provision a Virtual Machine workload in the Management System, additional settings will have to be filled in for the workload to perform as desired. Depending on the virtual machine that will be deployed, new interfaces might have to be defined. Keep this in mind and make sure to learn the details about the virtual machine.

Settings for Virtual machine workloads

In the instructions above, all optional settings have been left out. Below is an overview of all the options with an explanation to each option.

Setting	Description
VERSION SPECIFIC INFO	**Name** A name for the workload version. This could be a reminder for a certain configuration. Example: "Unlimited" as a name for a virtual machine that has unlimited access to CPU resources. **Release name** A release name for the workload version. This could be a version number. Example: 1.0.1
VIRTUAL MACHINE FILES TO UPLOAD	Two files need to be added here: **Virtual machine image** Upload the virtual machine image with the file extension RAW or IMG here. Do this by clicking the **upward arrow** symbol and selecting the file in the file browser. This is the first file generated in the process before. **Virtual machine xml** Upload the virtual machine XML file here. Do this by clicking the **upward arrow** symbol and selecting the file in the file browser. This is the second file generated in the process before. Note that the settings defined under **Virtual machine specific info** are going to overwrite parts of this XML file.
Virtual machine specific info	**Number of virtual CPUs** Define the number of virtual CPUs to assign to this virtual machine. The CPUs are then reserved exclusively for the Virtual Machine workload and cannot be used by other processes. This setting is mandatory and the workload cannot be provisioned if it is left blank. **Limit memory to** Assign how much system memory the workload is allowed to use. The memory assigned here will be reserved exclusively for this Virtual Machine workload and will not be available for any other processes. This setting is mandatory and the workload cannot be provisioned if it is left blank. **New data disk** Click the plus symbol to add a new data disk for the virtual machine. This data disk functions like an extra hard drive for data separate from the virtual machine. Enter a **Data disk name** and define the **Disk size**. **PCI passthrough** Click the plus symbol to add a PCI passthrough to the virtual machine. Enter the PCI address of the interface to pass through to be directly used by the virtual machine. Note that the PCIe address is specific to a certain hardware. When using this option, the installation targets to nodes with this specific hardware should be limited by using selectors. Refer to the labels chapter for more information on selectors. **New interface** Click the plus symbol to add a new interface. Choose between a bridged interface and a NAT-interface. For NAT interfaces port mappings for TCP and UDP can be defined. The names of the interfaces here have to match the names of the pre-defined network interfaces. Also, make sure to not use reserved ports for the workload. Refer to the networking chapter for more detailed information.

Setting	Description
SELECTOR	**Labels** If labels have been defined and assigned to nodes, add them as selectors to the workload. When deploying a workload, the list of nodes will be filtered automatically to the specified label.
Mark as released	Tick this checkbox to mark this workload as released. Once marked as released, the workload cannot be edited anymore.

Provisioning a Docker workload

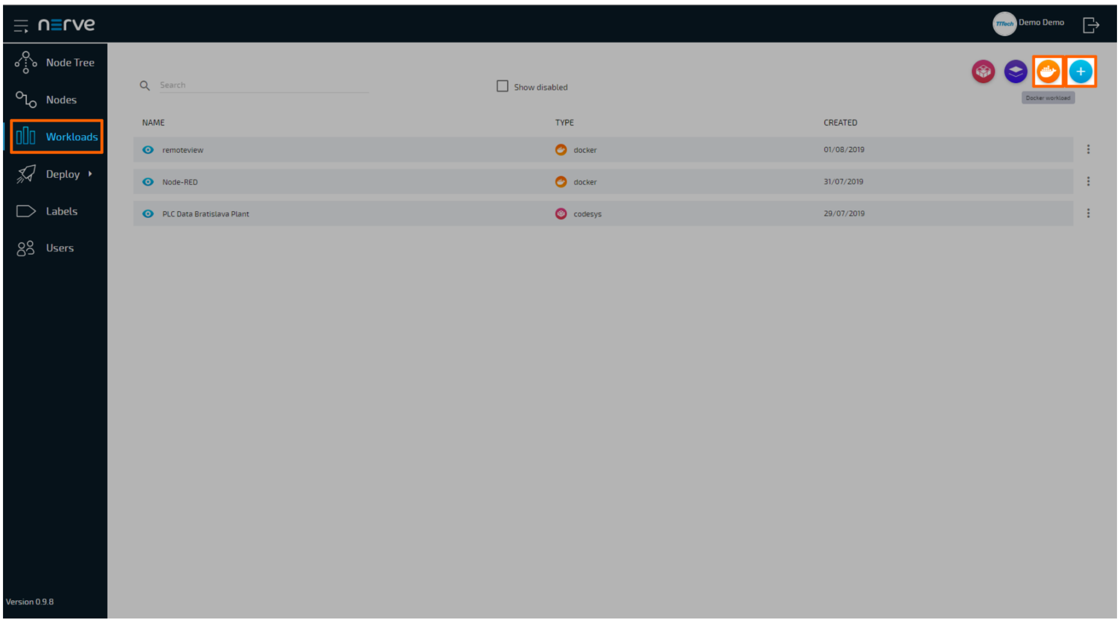
The following instructions cover the basic requirements for provisioning a Docker workload. Optional settings will be left out. Extended options are addressed in the last section of this chapter.
There are two further types of workloads that can be provisioned: CODESYS workloads and Virtual Machine workloads. The process for each workload is highlighted in its respective chapter.

NOTE

A Docker image is required for the following instructions. Refer to the Docker documentation for help on creating a Docker image.

1. Log in to the Management System.
2. Select **Workloads** in the left-hand menu.
3. Select the plus symbol in the upper-right corner.

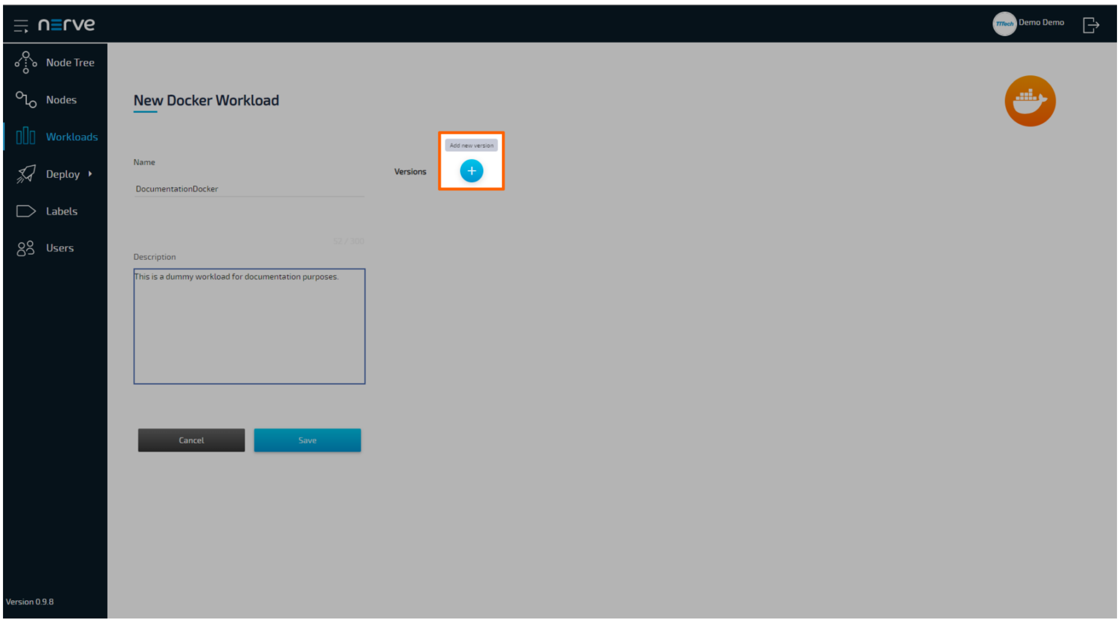
4. Select the Docker symbol (**Docker workload**) on the right of the three symbols that expanded.



NAME	TYPE	CREATED
remoteview	docker	01/08/2019
Node-RED	docker	31/07/2019
PLC Data Bratislava Plant	codesys	29/07/2019

5. In the new window, enter a name for the workload.

6. Select the plus symbol next to **Versions** to add a new version of the workload.



New Docker Workload

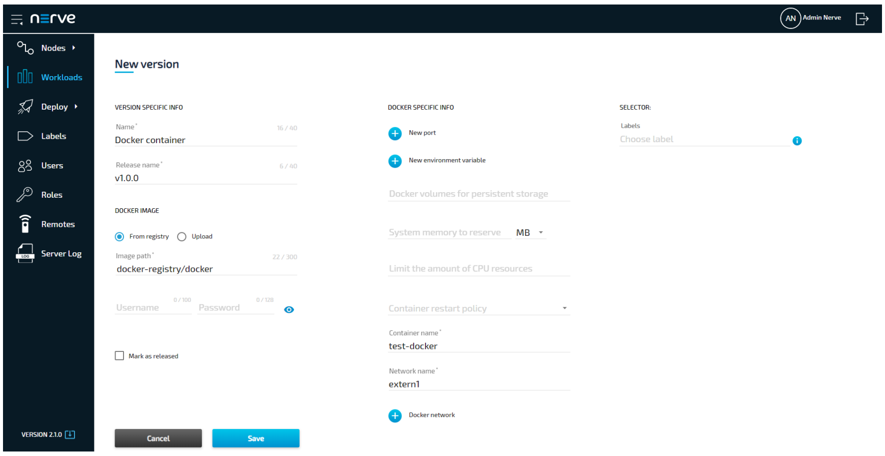
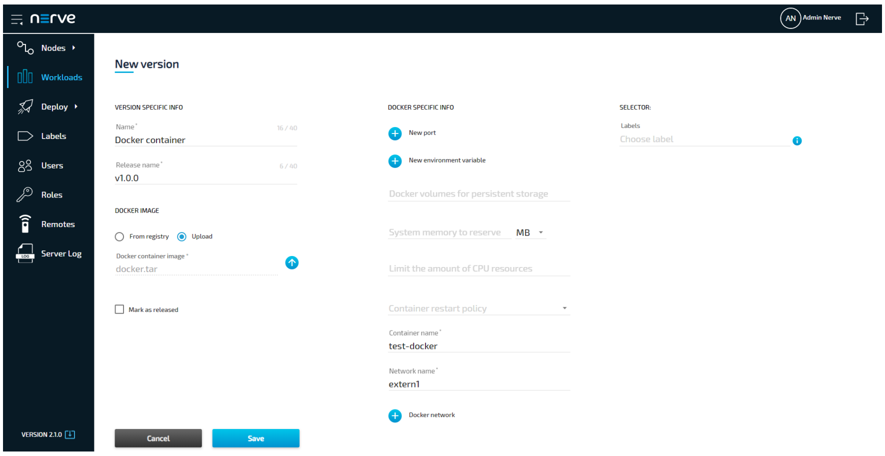
Name: DocumentationDocker

Description: This is a dummy workload for documentation purposes.

Versions

7. In the next window, enter the following information:

Item	Description
Name	Enter a **Name** for the version of this workload.
Release name	Enter a **Release name** for the version of this workload.

Item	Description
DOCKER IMAGE	Select **From registry** or **Upload**. **From registry**: Enter the Docker registry to the Docker container image under **Image path**.  nerve · AN Admin Nerve Nodes · Workloads · Deploy · Labels · Users · Roles · Remotes · Server Log · VERSION 2.1.0 New version VERSION SPECIFIC INFO Name* 16 / 40: Docker container Release name* 6 / 40: v1.0.0 DOCKER IMAGE ● From registry ○ Upload Image path* 22 / 300: docker-registry/docker Username 0 / 100 · Password 0 / 100 □ Mark as released Cancel · Save DOCKER SPECIFIC INFO New port New environment variable Docker volumes for persistent storage System memory to reserve MB Limit the amount of CPU resources Container restart policy Container name*: test-docker Network name*: extern1 Docker network SELECTOR: Labels: Choose label **Upload:** Click the **upward arrow** symbol to open the file browser and upload the Docker container image.  nerve · AN Admin Nerve Nodes · Workloads · Deploy · Labels · Users · Roles · Remotes · Server Log · VERSION 2.1.0 New version VERSION SPECIFIC INFO Name* 16 / 40: Docker container Release name* 6 / 40: v1.0.0 DOCKER IMAGE ○ From registry ● Upload Docker container image*: docker.tar □ Mark as released Cancel · Save DOCKER SPECIFIC INFO New port New environment variable Docker volumes for persistent storage System memory to reserve MB Limit the amount of CPU resources Container restart policy Container name*: test-docker Network name*: extern1 Docker network SELECTOR: Labels: Choose label
Container name	Enter a name for the Docker container. This will be the name of the Docker container on the node.
Network name	Enter the network interface name through which the Docker container can be reached. Refer to Node internal networking for more information. Note that the Docker default network **bridge** is always defined as an interface by default. That means all deployed Docker workloads can be reached through the **bridge** interface.

8. Select **Save** in the lower-left corner.

The workload has now been provisioned and is ready to be deployed in the **Deploy** menu.

NOTE

While some settings are not required to provision a Docker workload in the Management System, additional settings will have to be filled in for the workload to perform as desired. Depending on the Docker container that will be deployed, ports need to be defined and environment variables need to be configured. Keep this in mind and make sure to learn the details about the Docker container. Refer to the description of Docker workload settings below for more information.

Settings for Docker workloads

In the instructions above, all optional settings have been left out. Below is an overview of all the options with an explanation to each option.

Setting	Description
VERSION SPECIFIC INFO	**Name** A name for the workload version. This could be a reminder for a certain configuration. Example: "Unlimited" as a name for a Node-RED version that has unlimited access to CPU resources. **Release name** A release name for the workload version. This could be a version number. Example: 1.0.1
DOCKER IMAGE	Select between two options here and either use a Docker registry URL to link to an online repository or upload the Docker container image from the workstation. **From registry** Specify a URL pointing to the Docker container image under **Image path**. Note the differences between public Docker Hub registries and private registries. Private registries require the full URL to be specified, as well as a username and password if they require authentication. Public Docker Hub registries can be specified in their short form. Examples: • Public Docker Hub registry `nodered/node-red-docker` • Private registry with authentication and a tag at the end `auth.docker.test.host.cloud/workload:v1.3` **Upload** Upload the Docker container image from the workstation. Do so by clicking the **upward arrow** symbol to open the file browser.
Mark as released	Tick this checkbox to mark this workload as released. Once marked as released, the workload cannot be edited anymore.

DOCKER SPECIFIC INFO

New port
Click the plus symbol to define **Host port**, **Container port**, and **Protocol**.

- **Protocol**
  Choose **TCP** or **UDP** here.
- **Host port**
  This is the port through which the Docker workload will be reachable on the host. Make sure to not use reserved ports in the Nerve Blue system for the workload. Refer to the networking chapter for more information.
- **Container port**
  This is the internal port of the Docker container. Note that every Docker container has a default port. Entering a value that is different than the default port will use the entered port instead of the standard port of the Docker container.
  New environment variable
  Click the plus symbol to add an environment variable (**Env. variable**) and its **Variable value**. Make sure to define the appropriate variables and values as they depend on the Docker container that will be deployed,

  Docker volumes for persistent storage
  Persistent storage for a Docker workload can be defined using named volumes. Enter a path in the following format to define a Docker volume for persistent storage: `<volumename>:<containerpath>`
  - `<volumename>`
    Define a name through which the persistent Docker volume can be accessed on the host. This name can be any string.
  - `<containerpath>`
    Define a path inside the Docker container for the storage.

  The volume for persistent storage is not erased when the Docker workload is restarted or undeployed. The data also persists through a node version update. Docker volumes for persistent storage can be used for any workload by using the same volume name again. Docker workload storage can be found in `opt/data/var/lib/docker/volumes` by default.

  Limit memory to
  Assign how much system memory this workload is allowed to use. The memory assigned here is an upper limit that the Docker workload can use and is not exclusively reserved for the Docker workload. Other processes can use these resources as well.

  CPU resource in percentage
  Specify here the percentage of CPU resources the workload is allowed to use. If this field is left blank, the workload is allowed to use all available resources.

  Container restart policy
  Choose the container restart policy here to determine when the Docker container can be restarted.
  - **no**
    The container does not restart automatically.
  - **on-failure**
    The container restarts when it exits due to an error.
  - **always**
    The container restarts every time it stops. However, manually stopping the container is the exception. If a container is manually stopped, it is only restarted when the Docker daemon restarts or the container is restarted manually.
  - **unless-stopped**
    The container only restarts if it is manually stopped.

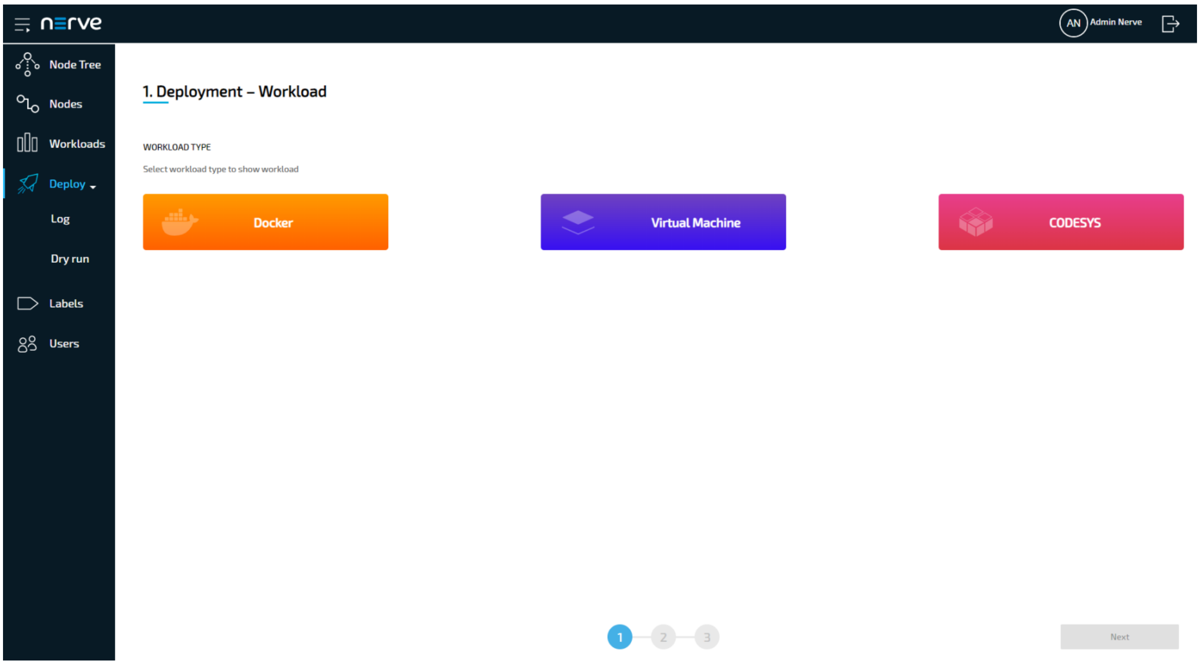
  Container name
  Enter a name for the Docker container. This will be the name of the Docker container on the node.

  Network name

Setting	Description
SELECTOR	**Labels** If labels have been defined and assigned to nodes, add them as selectors to the workload. When deploying a workload, the list of nodes will be filtered automatically to the specified label.

Deploy menu

Workloads that have been provisioned in the Management System are ready to be deployed to nodes through the **Deploy** menu. It expands into three menus:



1. Deployment – Workload

WORKLOAD TYPE

Select workload type to show workload

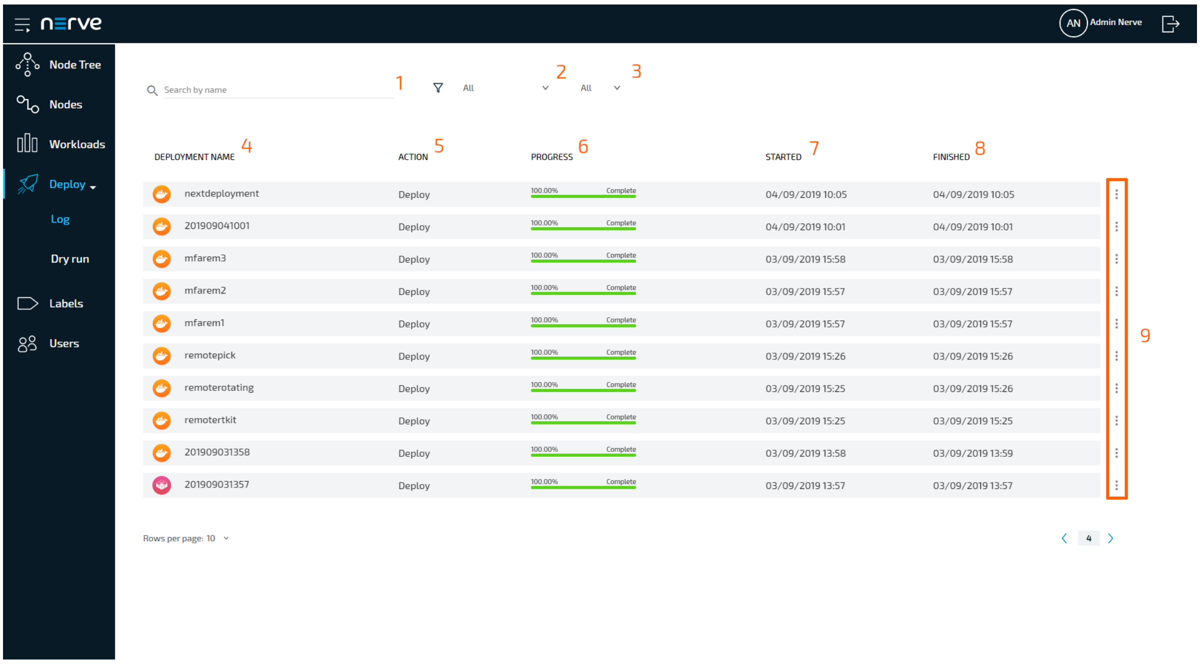
Item	Description
Deploy	This is the landing page of the deployment menu. Deploy workloads to nodes from here.
Log	This is the history of deployments and dry runs.
Dry run	Structurally the same as the deployment process for workloads, simulate the deployment of a workload from this menu.

Log

The log is the history of deployments and dry runs. This includes:

- deployments in progress
- dry runs in progress
- failed deployments
- failed dry runs
- successful deployments
- successful dry runs

It is displayed in reverse chronological order and can be filtered according to search criteria. It also offers some control functionality for active and failed deployments.



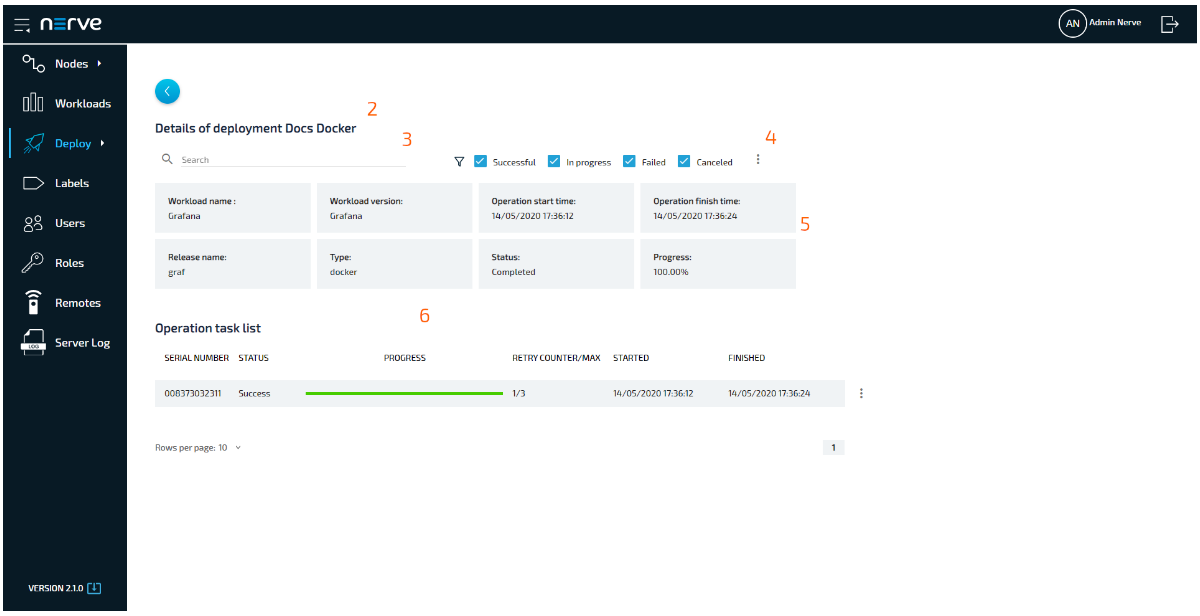
DEPLOYMENT NAME	ACTION	PROGRESS	STARTED	FINISHED
nextdeployment	Deploy	100.00% Complete	04/09/2019 10:05	04/09/2019 10:05
201909041001	Deploy	100.00% Complete	04/09/2019 10:01	04/09/2019 10:01
mfarem3	Deploy	100.00% Complete	03/09/2019 15:58	03/09/2019 15:58
mfarem2	Deploy	100.00% Complete	03/09/2019 15:57	03/09/2019 15:57
mfarem1	Deploy	100.00% Complete	03/09/2019 15:57	03/09/2019 15:57
remotepick	Deploy	100.00% Complete	03/09/2019 15:26	03/09/2019 15:26
remoterotating	Deploy	100.00% Complete	03/09/2019 15:25	03/09/2019 15:26
remotertkit	Deploy	100.00% Complete	03/09/2019 15:25	03/09/2019 15:25
201909031358	Deploy	100.00% Complete	03/09/2019 13:58	03/09/2019 13:59
201909031357	Deploy	100.00% Complete	03/09/2019 13:57	03/09/2019 13:57

Item	Description
Search by name (1)	Enter text here to filter the list by deployment name.
Deployment Type (2)	Select an option from the drop-down menu to filter the list for **Deploys** or **Dry runs**.
Workload Type (3)	Select an option from the drop-down menu to filter the list for a specific workload type: **VM**, **Docker** or **CODESYS**.
DEPLOYMENT NAME (4)	This is the name of the deployment with the workload type displayed as a symbol. The default is the time of deployment unless a deployment name has been entered during the deployment process. Note that this is not the name of the workload.
ACTION (5)	Here the deployment type is displayed: **Deploy** or **Dry run**.
PROGRESS (6)	The progress bar is an indicator for both progress and status. Depending on the status of the deployment it changes its color: • **Green** If a workload was deployed successfully, the bar will be green at a 100%. • **Blue** If a workload is currently being deployed, the bar will be blue, fill up gradually and display the progress of the deployment in percent. • **Red** If the deployment of a workload has failed, the bar will be red at a 100%.
STARTED (7)	This is the date and time the workload deployment was started. The date format is DD/MM/YYYY.
FINISHED (8)	This is the date and time the workload deployment was completed. The date format is DD/MM/YYYY. This field will display **In progress** if a workload is in progress of being deployed.

Item	Description
Ellipsis menu (9)	Select the ellipsis menu to trigger an overlay with the **DELETE** option. Selecting **DELETE** will remove the entry from the log. When a workload is in progress, this entry is grayed out.

Deployment details

Clicking an entry in the log will show the details of the deployment.



Details of deployment Docs Docker

Search

☑ Successful ☑ In progress ☑ Failed ☑ Canceled

Workload name : Grafana

Workload version: Grafana

Operation start time: 14/05/2020 17:36:12

Operation finish time: 14/05/2020 17:36:24

Release name: graf

Type: docker

Status: Completed

Progress: 100.00%

Operation task list

SERIAL NUMBER	STATUS	PROGRESS	RETRY COUNTER/MAX	STARTED	FINISHED
008373032311	Success		1/3	14/05/2020 17:36:12	14/05/2020 17:36:24

Rows per page: 10

1

Item	Description
Back button (1)	Click here to return to the log.
Header (2)	The header states the name of the deployment in the format **Details of deployment <deploymentname>**.
Search (3)	Enter text here to filter the **Operation task list** by device name. The search function can be combined with the status checkboxes to the right of the search bar.
Status checkboxes (4)	The checkboxes to the right of the search bar filter the **Operation task list** by status: • **Success** Deployments that were completed successfully are shown in the list if this checkbox is ticked. • **In progress** Deployments that are currently in progress are shown in the list if this checkbox is ticked. • **Failed** Deployments that could not be completed are shown if this checkbox is ticked. • **Canceled** Deployments that have been aborted by the user are shown in the list if this checkbox is ticked. However, it is not possible to cancel deployments in this version. If a checkbox is ticked, tasks with the corresponding status will be displayed in the operation task list. All checkboxes are ticked by default.

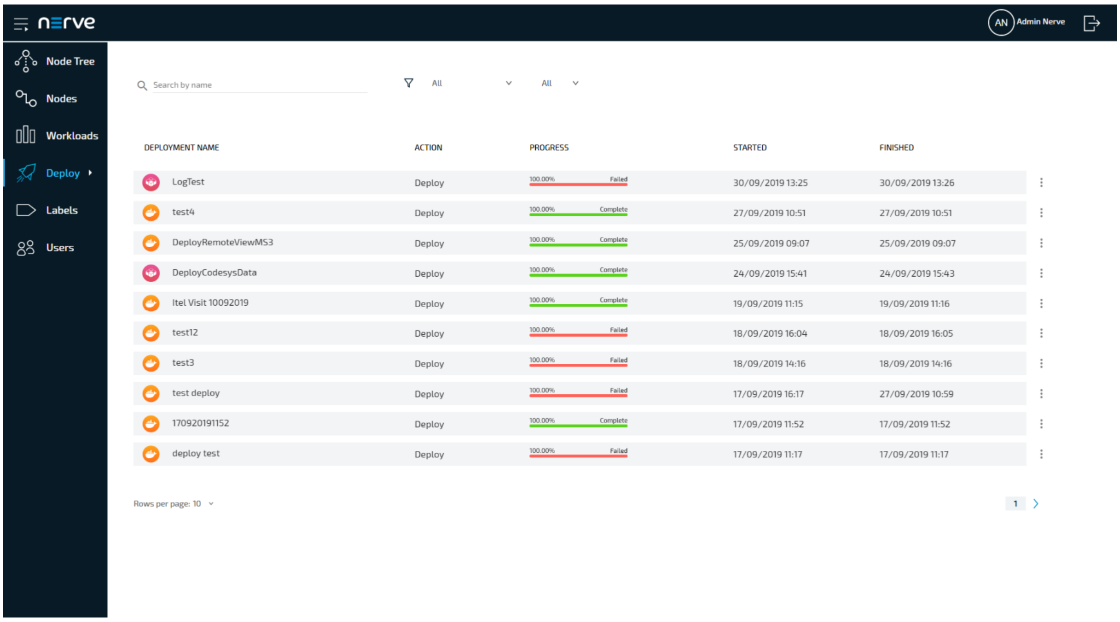
Item	Description
Deployment information (5)	This is information about the deployment. • **Workload name** The name of the workload that has been defined in the provisioning process. • **Workload version** The name of the workload version. • **Operation start time** This is the date and time the workload deployment was started. The date format is DD/MM/YYYY. • **Operation finish time** This is the date and time the workload deployment finished. The date format is DD/MM/YYYY. • **Release name** This is the release name of the workload version. • **Type** This is the type of the workload that has been deployed: **codesys**, **vm** or **docker**. • **Status** This is the status of the deployment. Possible statuses are **Created**, **In progress**, **Completed**, **Error**, **Canceled**. • **Progress** The progress of the deployment in percent.
Operation task list (6)	The operation task list displays details for single deployments that are part of the deployment campaign. The list displays information in six columns: • **SERIAL NUMBER** This is the serial number of the node that is the target of the deployment. • **STATUS** This is the status of the deployment. The information in this column here corresponds with the checkboxes to the right of the search bar: **Success**, **In progress**, **Failed** and **Canceled** • **PROGRESS** This is the progress bar. It displays a different color and gradually fills up according to the progress and status of the deployment. ◦ **Success** If a workload was deployed successfully, the bar will be green at a 100%. ◦ **In progress** If a workload is currently being deployed, the bar will be blue and display the progress of the deployment in percent. ◦ **Failed** If the deployment of a workload has failed, the bar will be red at a 100%. • **RETRY COUNTER/MAX** In case of failure, the Management System will attempt the deployment of a workload up to three times automatically. The left number of the counter shows the number of the current attempt. The right number is the maximum number of attempts. • **STARTED** This is the date and time the workload deployment was started. The date format is DD/MM/YYYY. • **FINISHED** This is the date and time the workload deployment was finished. The date format is DD/MM/YYYY.

Restarting a failed deployment

In case a deployment fails, the Management System will attempt the deployment of a workload up to three times automatically. After that, the deployment can be restarted manually through the ellipsis menu in the operation task list.

1. Select **Deploy > Log** from the menu on the left.

2. Select the failed deployment from the log.



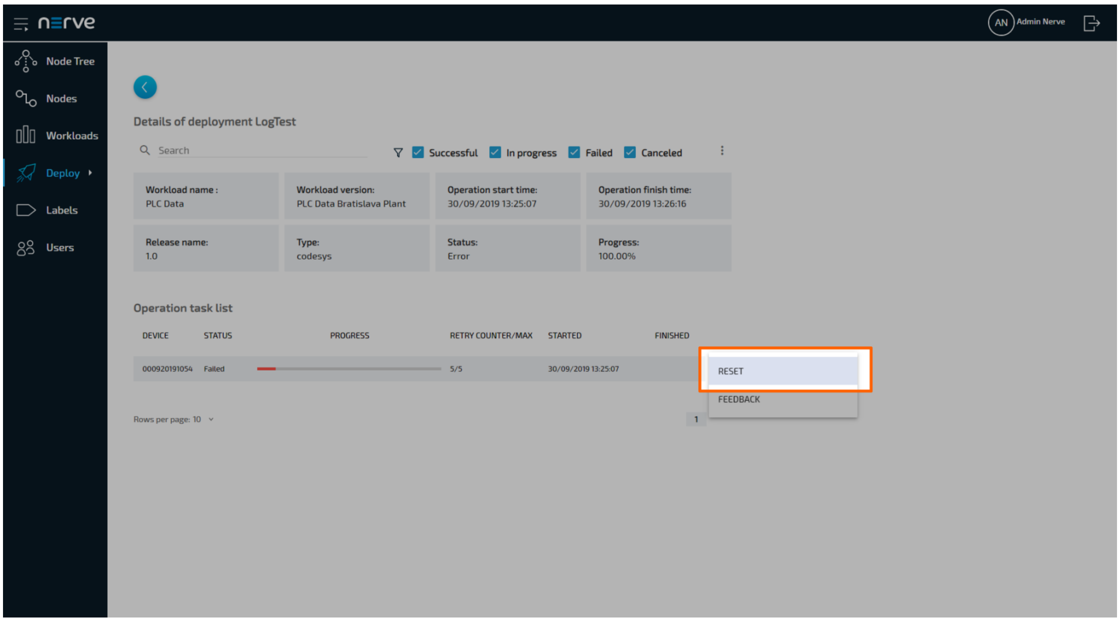
DEPLOYMENT NAME	ACTION	PROGRESS	STARTED	FINISHED
LogTest	Deploy	100.00% Failed	30/09/2019 13:25	30/09/2019 13:26
test4	Deploy	100.00% Complete	27/09/2019 10:51	27/09/2019 10:51
DeployRemoteViewMS3	Deploy	100.00% Complete	25/09/2019 09:07	25/09/2019 09:07
DeployCodesysData	Deploy	100.00% Complete	24/09/2019 15:41	24/09/2019 15:43
Itel Visit 10092019	Deploy	100.00% Complete	19/09/2019 11:15	19/09/2019 11:16
test12	Deploy	100.00% Failed	18/09/2019 16:04	18/09/2019 16:05
test3	Deploy	100.00% Failed	18/09/2019 14:16	18/09/2019 14:16
test deploy	Deploy	100.00% Failed	17/09/2019 16:17	27/09/2019 10:59
170920191152	Deploy	100.00% Complete	17/09/2019 11:52	17/09/2019 11:52
deploy test	Deploy	100.00% Failed	17/09/2019 11:17	17/09/2019 11:17

Rows per page: 10

3. Choose the failed deployment from the operation task list.

4. Select the ellipsis menu to the right of the deployment entry.

5. Select **RESET** in the overlay that appeared.



Details of deployment LogTest

Workload name : PLC Data

Workload version: PLC Data Bratislava Plant

Operation start time: 30/09/2019 13:25:07

Operation finish time: 30/09/2019 13:26:16

Release name: 1.0

Type: codesys

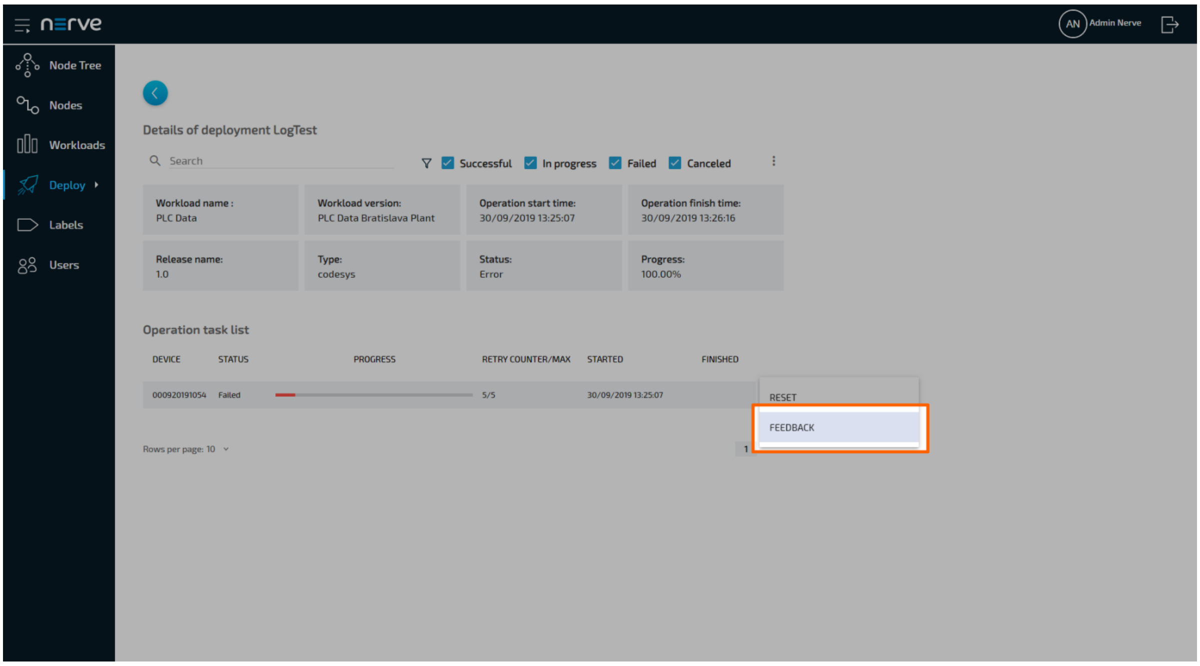
Status: Error

Progress: 100.00%

Operation task list

DEVICE	STATUS	PROGRESS	RETRY COUNTER/MAX	STARTED	FINISHED
000920191054	Failed		5/5	30/09/2019 13:25:07	

The deployment is restarted immediately. To see the error information of the deployment, select **FEEDBACK** in the overlay of the ellipsis menu instead.



Details of deployment LogTest

Search ☑ Successful ☑ In progress ☑ Failed ☑ Canceled

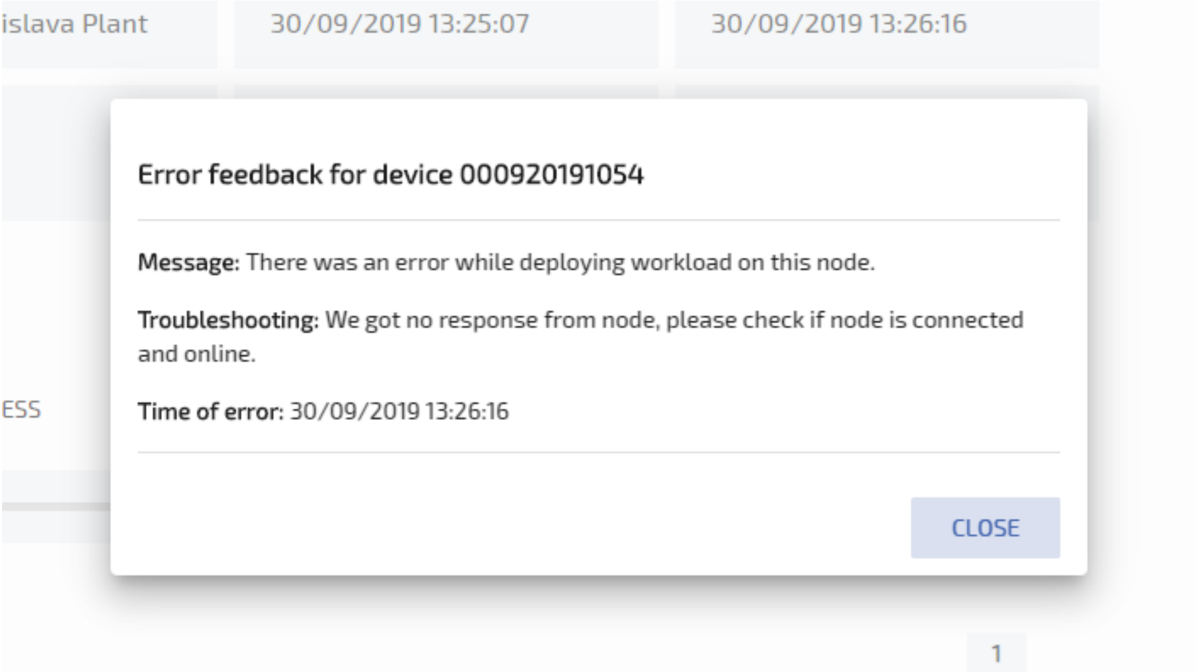
Workload name : PLC Data	Workload version: PLC Data Bratislava Plant	Operation start time: 30/09/2019 13:25:07	Operation finish time: 30/09/2019 13:26:16
Release name: 1.0	Type: codesys	Status: Error	Progress: 100.00%

Operation task list

DEVICE	STATUS	PROGRESS	RETRY COUNTER/MAX	STARTED	FINISHED
000920191054	Failed		5/5	30/09/2019 13:25:07	

Rows per page: 10

This opens a window giving information on the type of error in a message, a short troubleshooting hint and the time the error occurred. Note that troubleshooting hints are not available for every error case.



Error feedback for device 000920191054

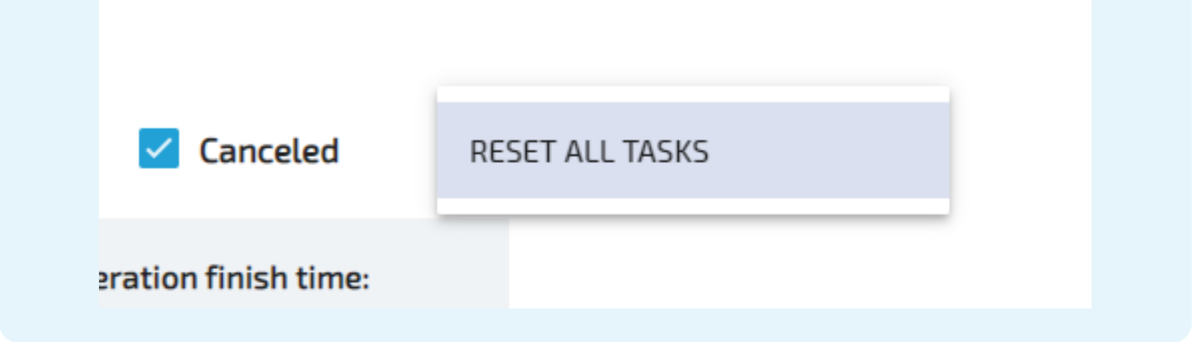
Message: There was an error while deploying workload on this node.

Troubleshooting: We got no response from node, please check if node is connected and online.

Time of error: 30/09/2019 13:26:16

NOTE

To restart all deployments at once, click the ellipsis menu next to the status checkboxes and select **RESET ALL TASKS** from the overlay that appears.



☑ Canceled

RESET ALL TASKS

eration finish time:

Deploying workloads and dry runs

Deployment of workloads and dry runs are covered in a separate chapter: Deploying a Workload.

Deploying a workload

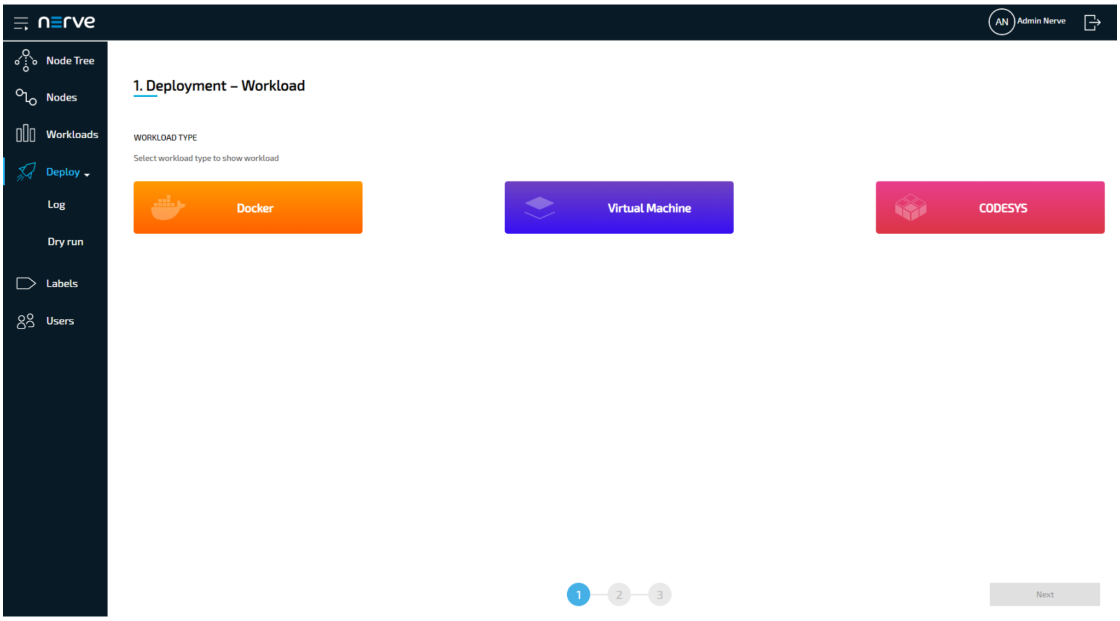
This chapter covers the deployment process of the available workloads: CODESYS workloads, Virtual Machine workloads and Docker workloads. The process of deploying workloads is identical for all three types of workloads. Therefore, the instructions below contain no specific information.

NOTE

To test a deployment first, select **Deploy > Dry run** in the left-hand menu and follow the steps below starting from step 3.

However, note that a successful dry run does not guarantee a successful deployment as it is only a simulation.

1. Select **Deploy** in the left-hand menu.

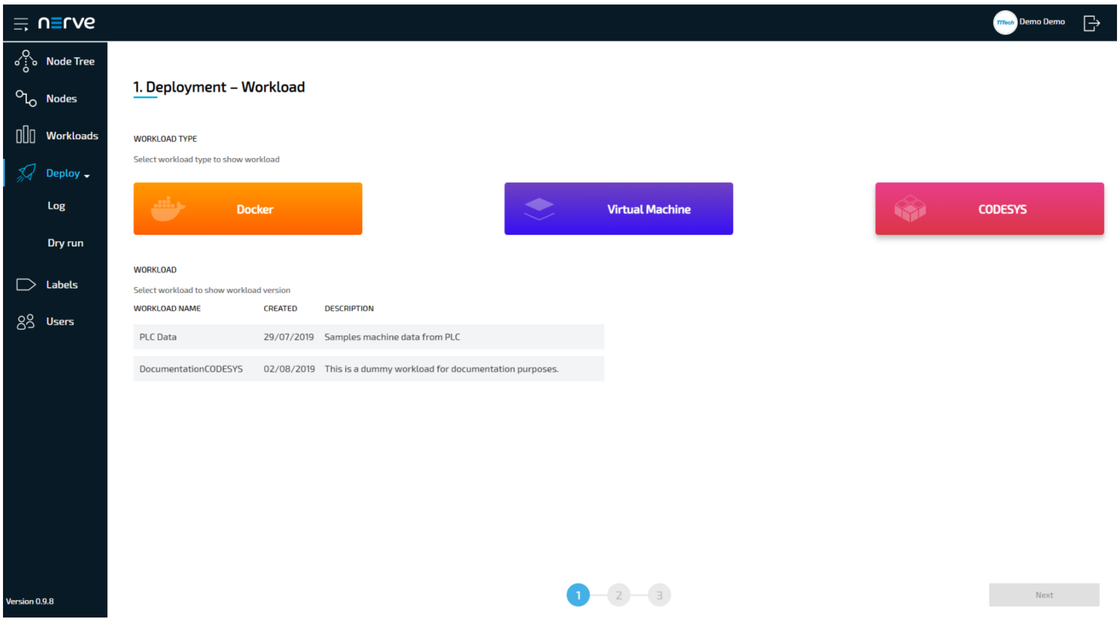


1. Deployment – Workload

WORKLOAD TYPE

Select workload type to show workload

2. Select one of the three icons for workload types. A list of corresponding workloads will appear below.



1. Deployment – Workload

WORKLOAD TYPE

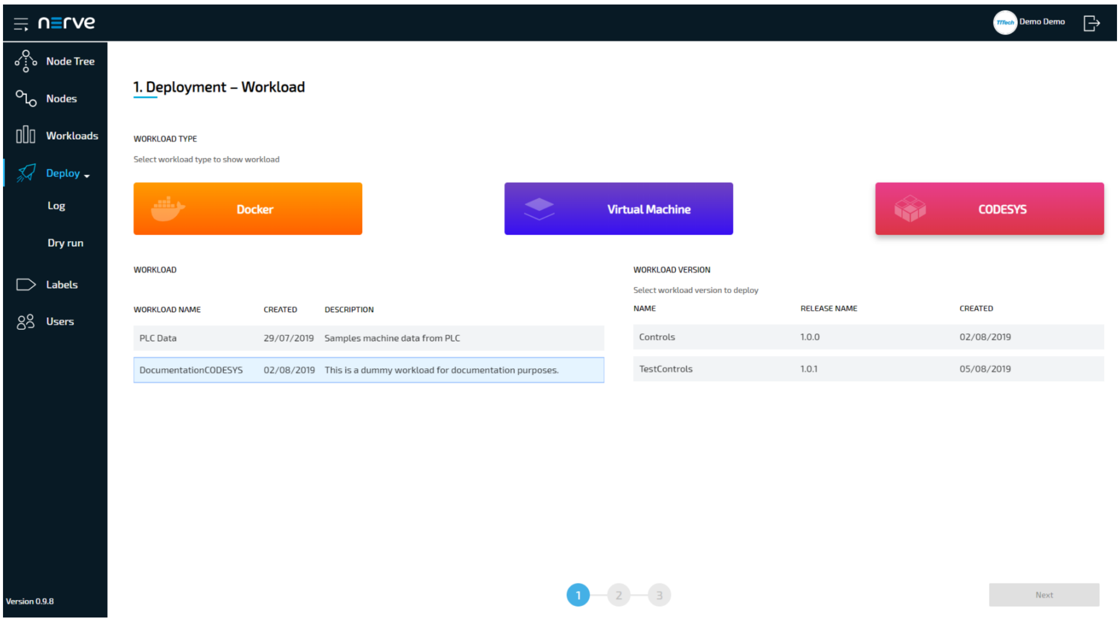
Select workload type to show workload

WORKLOAD

Select workload to show workload version

WORKLOAD NAME	CREATED	DESCRIPTION
PLC Data	29/07/2019	Samples machine data from PLC
DocumentationCODESYS	02/08/2019	This is a dummy workload for documentation purposes.

3. Select a workload from the list. A list of versions of this workload will appear to the right.



1. Deployment – Workload

WORKLOAD TYPE

Select workload type to show workload

WORKLOAD

WORKLOAD NAME	CREATED	DESCRIPTION
PLC Data	29/07/2019	Samples machine data from PLC
DocumentationCODESYS	02/08/2019	This is a dummy workload for documentation purposes.

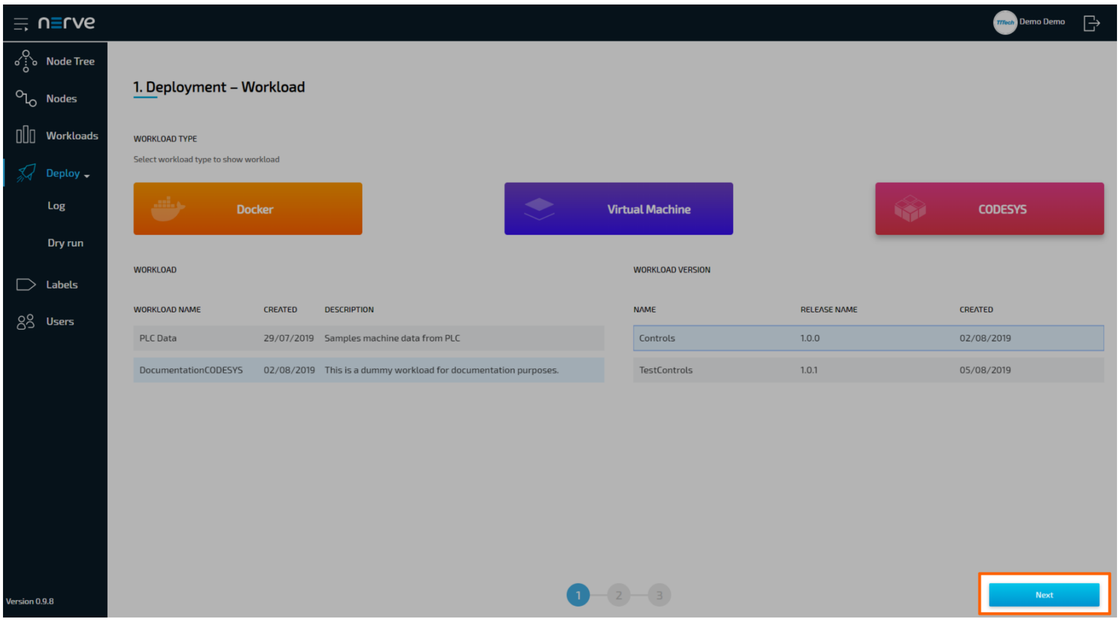
WORKLOAD VERSION

Select workload version to deploy

NAME	RELEASE NAME	CREATED
Controls	1.0.0	02/08/2019
TestControls	1.0.1	05/08/2019

4. Select the version of the workload to deploy.

5. Click **Next** in the bottom-right corner.



1. Deployment – Workload

WORKLOAD TYPE

Select workload type to show workload

WORKLOAD

WORKLOAD NAME	CREATED	DESCRIPTION
PLC Data	29/07/2019	Samples machine data from PLC
DocumentationCODESYS	02/08/2019	This is a dummy workload for documentation purposes.

WORKLOAD VERSION

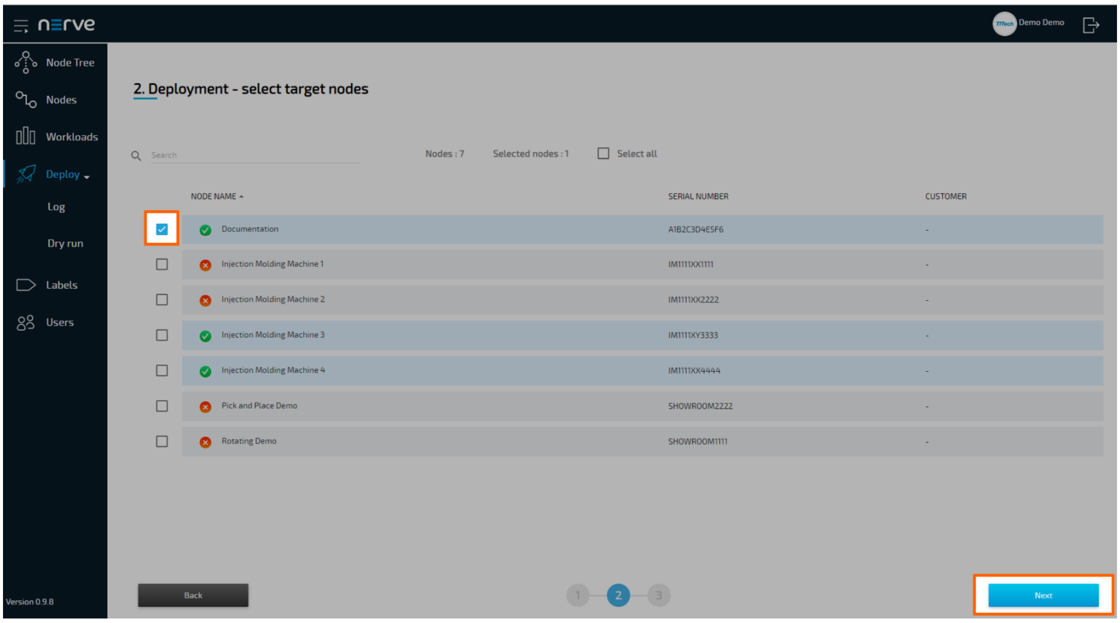
NAME	RELEASE NAME	CREATED
Controls	1.0.0	02/08/2019
TestControls	1.0.1	05/08/2019

6. In the next window, select one or more nodes from the list for deployment by ticking the checkboxes on the left.

NOTE

This list of nodes might not include all nodes that are registered in the Management System. It is automatically filtered depending on the labels the workload has assigned.

7. Select **Next** in the lower-right corner.

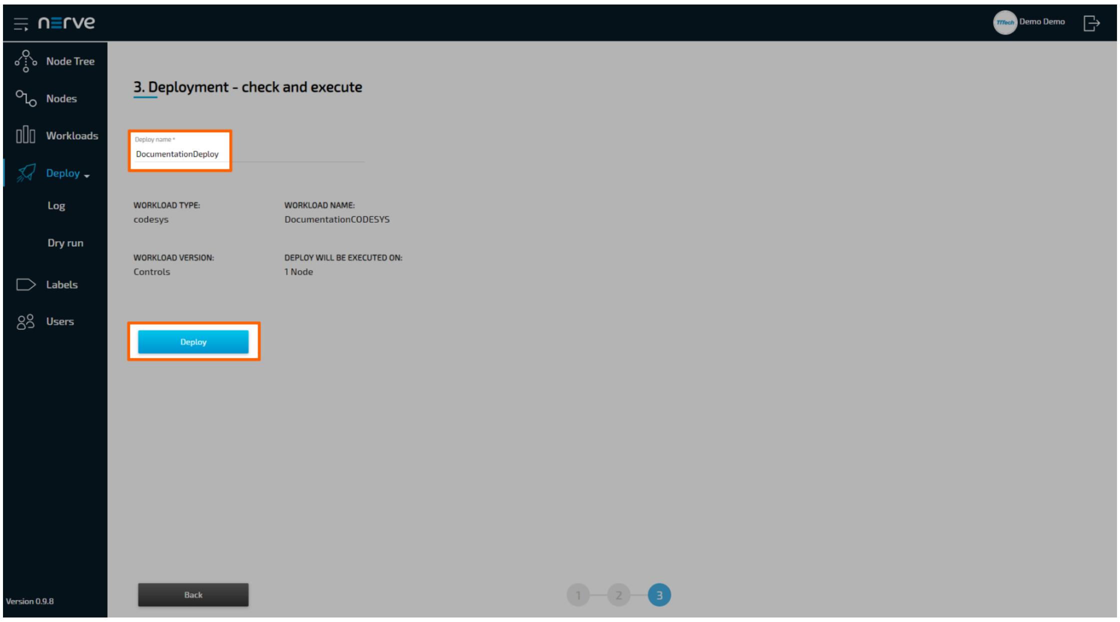


2. Deployment - select target nodes

Search Nodes : 7 Selected nodes : 1 □ Select all

	NODE NAME	SERIAL NUMBER	CUSTOMER
☑	✓ Documentation	A1B2C3D4E5F6	-
□	✗ Injection Molding Machine 1	IM1111XX1111	-
□	✗ Injection Molding Machine 2	IM1111XX2222	-
□	✓ Injection Molding Machine 3	IM1111XY3333	-
□	✓ Injection Molding Machine 4	IM1111XX4444	-
□	✗ Pick and Place Demo	SHOWROOM2222	-
□	✗ Rotating Demo	SHOWROOM1111	-

8. Select **Deploy** to execute the deployment.
Optional: Enter a **Deploy name** *above the* **Summary** *of the workload to make this deployment easy to identify. A timestamp is filled in automatically.*



3. Deployment - check and execute

Deploy name *: DocumentationDeploy

WORKLOAD TYPE: codesys

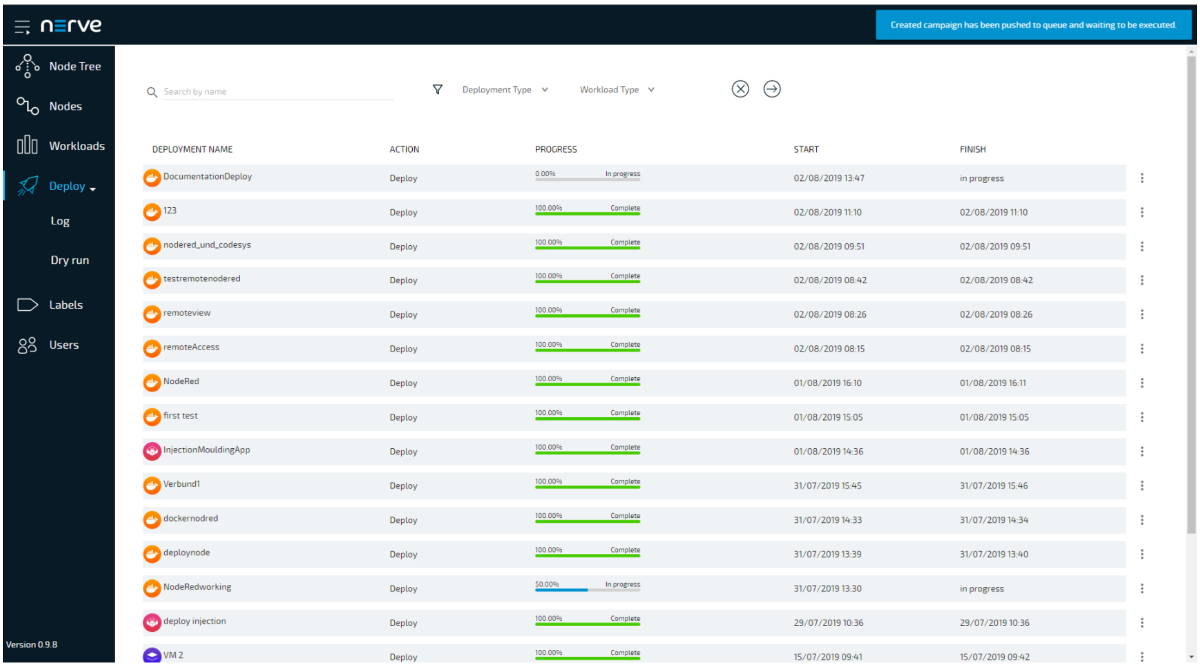
WORKLOAD NAME: DocumentationCODESYS

WORKLOAD VERSION: Controls

DEPLOY WILL BE EXECUTED ON: 1 Node

Deploy

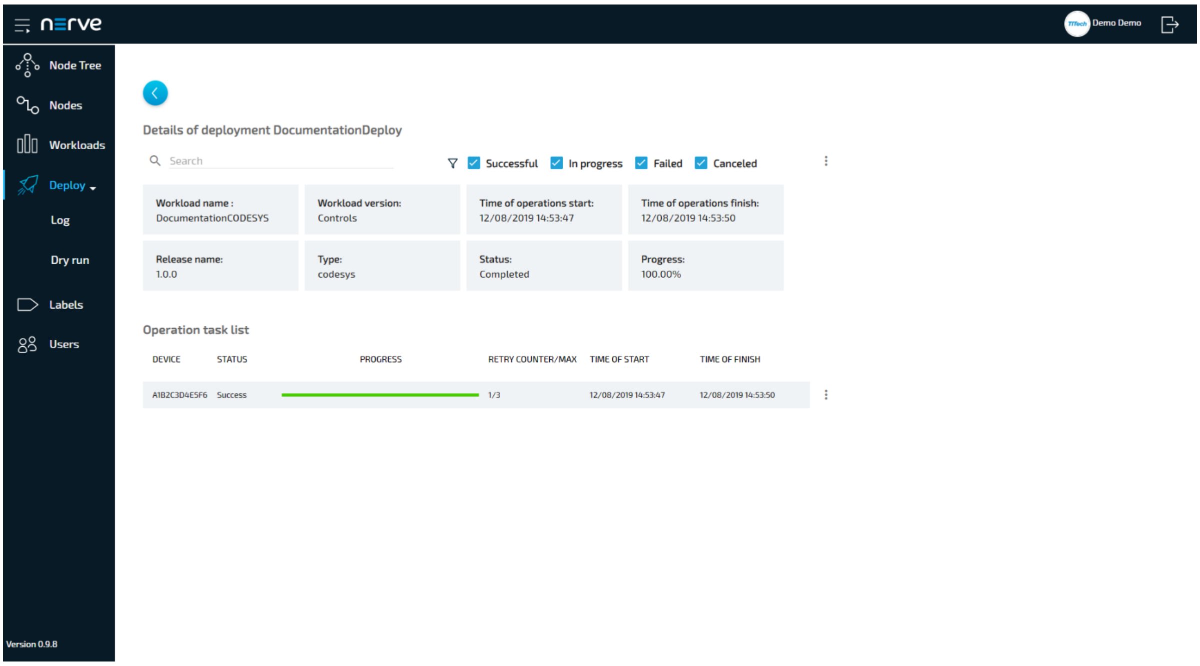
The Management System will continue to the log next. The current deployment is at the top of the list. The **Deploy name** chosen before is the name that identifies the deployment in the log.



Created campaign has been pushed to queue and waiting to be executed.

DEPLOYMENT NAME	ACTION	PROGRESS	START	FINISH
DocumentationDeploy	Deploy	0.00% In progress	02/08/2019 13:47	in progress
123	Deploy	100.00% Complete	02/08/2019 11:10	02/08/2019 11:10
nodered_und_codesys	Deploy	100.00% Complete	02/08/2019 09:51	02/08/2019 09:51
testremotenodered	Deploy	100.00% Complete	02/08/2019 08:42	02/08/2019 08:42
remoteview	Deploy	100.00% Complete	02/08/2019 08:26	02/08/2019 08:26
remoteAccess	Deploy	100.00% Complete	02/08/2019 08:15	02/08/2019 08:15
NodeRed	Deploy	100.00% Complete	01/08/2019 16:10	01/08/2019 16:11
first test	Deploy	100.00% Complete	01/08/2019 15:05	01/08/2019 15:05
InjectionMouldingApp	Deploy	100.00% Complete	01/08/2019 14:36	01/08/2019 14:36
Verbund1	Deploy	100.00% Complete	31/07/2019 15:45	31/07/2019 15:46
dockernodred	Deploy	100.00% Complete	31/07/2019 14:33	31/07/2019 14:34
deploynode	Deploy	100.00% Complete	31/07/2019 13:39	31/07/2019 13:40
NodeRedworking	Deploy	50.00% In progress	31/07/2019 13:30	in progress
deploy injection	Deploy	100.00% Complete	29/07/2019 10:36	29/07/2019 10:36
VM 2	Deploy	100.00% Complete	15/07/2019 09:41	15/07/2019 09:42

The progress of the current deployment is displayed here. Select the log entry of the deployment to see a more detailed view.



Details of deployment DocumentationDeploy

Search ☑ Successful ☑ In progress ☑ Failed ☑ Canceled

Workload name : DocumentationCODESYS

Workload version: Controls

Time of operations start: 12/08/2019 14:53:47

Time of operations finish: 12/08/2019 14:53:50

Release name: 1.0.0

Type: codesys

Status: Completed

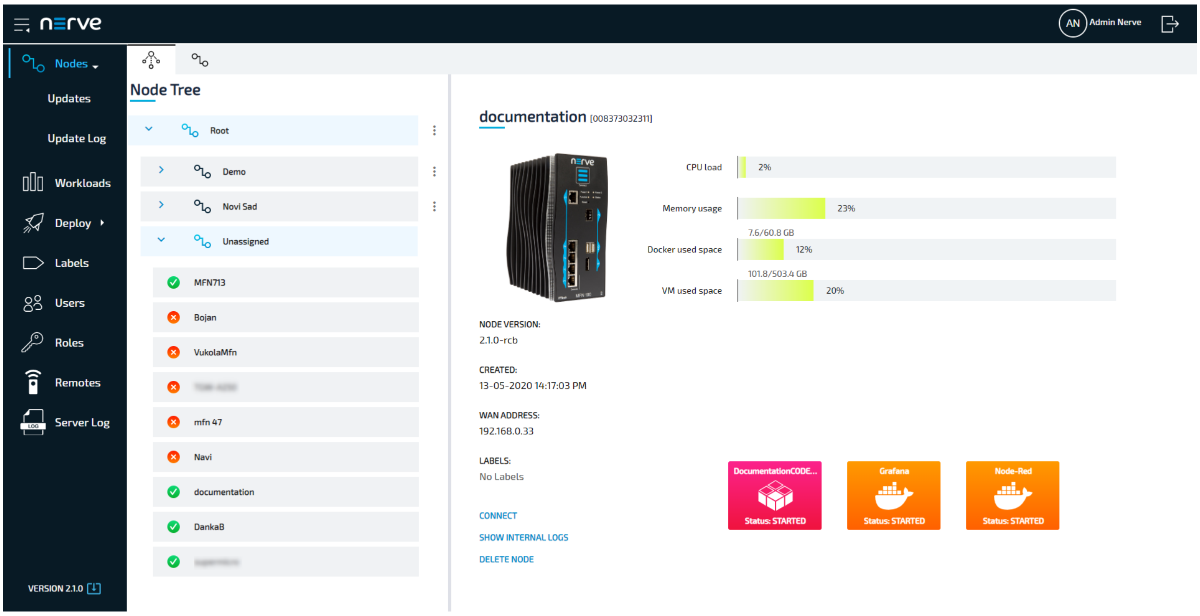
Progress: 100.00%

Operation task list

DEVICE	STATUS	PROGRESS	RETRY COUNTER/MAX	TIME OF START	TIME OF FINISH
A1B2C3D4E5F6	Success		1/3	12/08/2019 14:53:47	12/08/2019 14:53:50

The workload has been deployed and can be controlled in the node tree. Select **Nodes** in the navigation on the left and select the node tree tab

on the right. Select the node with the deployed workload to find the tile of the deployed workload.



documentation [008373032311]

CPU load 2%

Memory usage 23%

7.6/60.8 GB

Docker used space 12%

101.8/503.4 GB

VM used space 20%

NODE VERSION: 2.1.0-rcb

CREATED: 13-05-2020 14:17:03 PM

WAN ADDRESS: 192.168.0.33

LABELS: No Labels

Reach the workload control area by clicking the tile of each workload. All workloads are started as soon as they are deployed.

NOTE

CODESYS applications can only be controlled through the Local UI.

Removing logical volumes after unsuccessful VM deployment

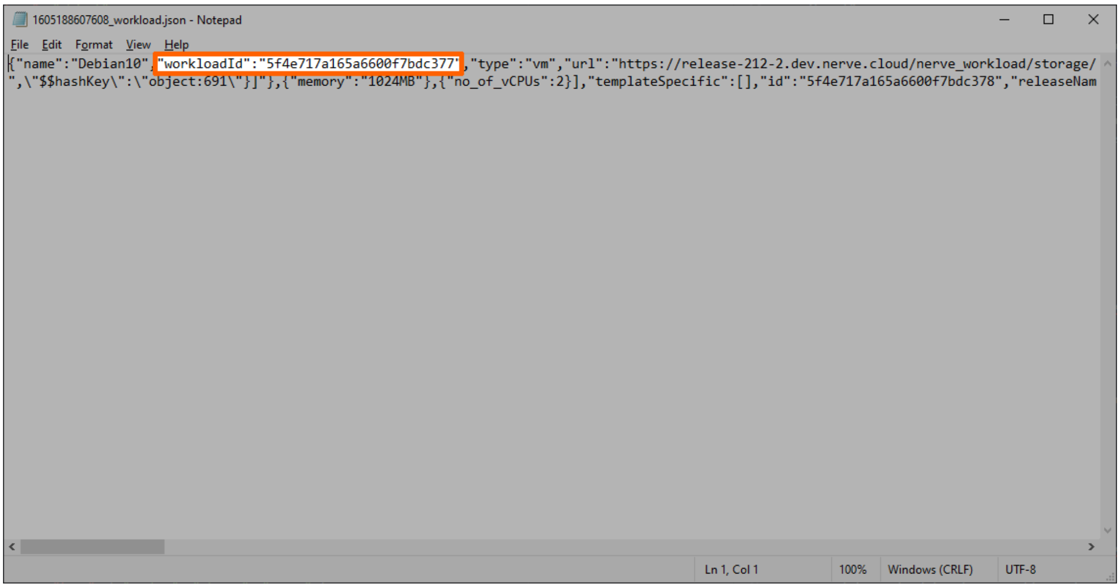
If a virtual machine workload fails in the download phase of the deployment process, it is possible that the logical volumes created for the workload stay behind. This does not negatively impact the system. However, the disk space reserved for the virtual machine workload will stay reserved unless it is removed. Follow the instructions below to find out the workload ID of the failed deployment and how to remove the logical volume.

Finding out the workload ID

Every workload has a unique ID that is used for naming the logical volume when it is created during the deployment process. This workload ID can be found in a JSON file when a workload is exported from the Management System.

1. Access the Management System.
2. Export the virtual machine workload the deployment of which has failed by following Exporting a workload. The workload is downloaded as a TAR file.
3. Navigate to the folder where the TAR file is located.
4. Extract the TAR file.
5. Open the JSON file with a text editor. The TAR file only contains one JSON file.

6. Look for `"workloadId"` in the file. It is the second entry in the file.


1605188607608_workload.json - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

```
{"name":"Debian10","workloadId":"5f4e717a165a6600f7bdc377","type":"vm","url":"https://release-212-2.dev.nerve.cloud/nerve_workload/storage/
",\"$$hashKey\":\"object:691\"}]"},{"memory":"1024MB"},{"no_of_vCPUs":2}],"templateSpecific":[],"id":"5f4e717a165a6600f7bdc378","releaseNam
```

7. Note down the ID. It is required to identify the logical volume that needs to be removed.

Removing the logical volumes

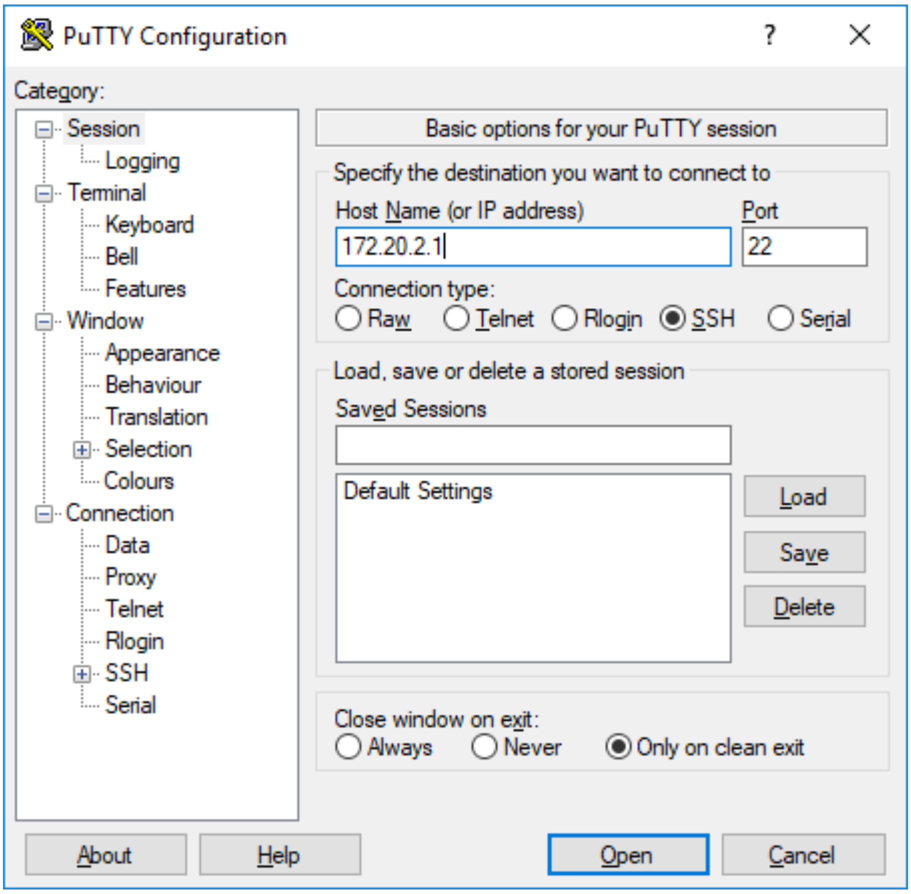
With the workload ID of the unsuccessfully deployed workload, the logical volumes can be correctly identified and removed. For this, the workstation needs to be connected to the physical port of the Nerve Device associated with host access, and the network adapter IP address of the workstation needs to be configured in the correct range. This information is device specific. Refer to the device guide for information on the Nerve Device.

NOTE

The following instructions require access to the Linux host system of Nerve Blue. Using host access requires expert Linux knowledge as system internal changes can be performed. Note that changes may impact the Nerve Blue system.

1. Open an SSH client like PuTTY.

2. Enter the IP address for host access to the Nerve Device under **Host Name (or IP address)** to log in to the host of the Nerve Device.



Host Name (or IP address): 172.20.2.1

Port: 22

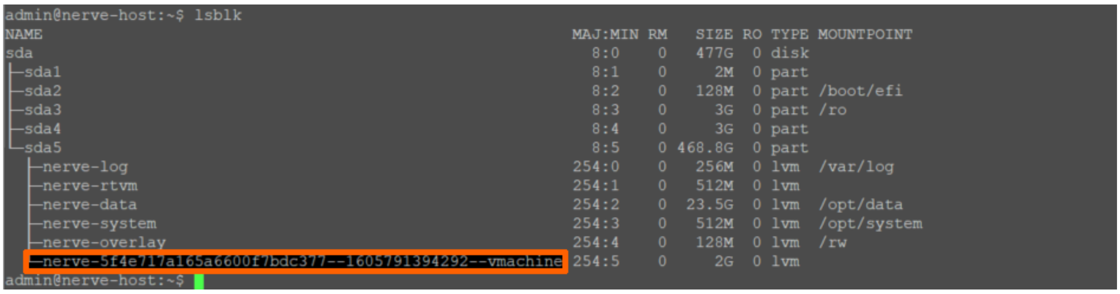
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Close window on exit: ○ Always ○ Never ● Only on clean exit

3. Log in with the credentials for host access to the Nerve Device.

4. Enter `lsblk` to display a list of volumes.

5. Find the logical volume containing the `workloadId` from the JSON file.



```
admin@nerve-host:~$ lsblk
NAME                                                        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                                                           8:0    0   477G  0 disk
├─sda1                                                        8:1    0     2M  0 part
├─sda2                                                        8:2    0   128M  0 part /boot/efi
├─sda3                                                        8:3    0     3G  0 part /ro
├─sda4                                                        8:4    0     3G  0 part
└─sda5                                                        8:5    0 468.8G  0 part
  ├─nerve-log                                               254:0    0   256M  0 lvm  /var/log
  ├─nerve-rtvm                                              254:1    0   512M  0 lvm
  ├─nerve-data                                              254:2    0  23.5G  0 lvm  /opt/data
  ├─nerve-system                                            254:3    0   512M  0 lvm  /opt/system
  ├─nerve-overlay                                           254:4    0   128M  0 lvm  /rw
  └─nerve-5f4e717a165a6600f7bdc377--1605791394292--vmachine 254:5    0     2G  0 lvm
admin@nerve-host:~$
```

6. Enter the following command to remove the logical volumes:

```
sudo lvremove /dev/nerve/<volume-name> -y
```

NOTE

Note that the steps above can also be performed through a remote connection. Refer to Configuring a remote tunnel to a node for more information.

Remote connections

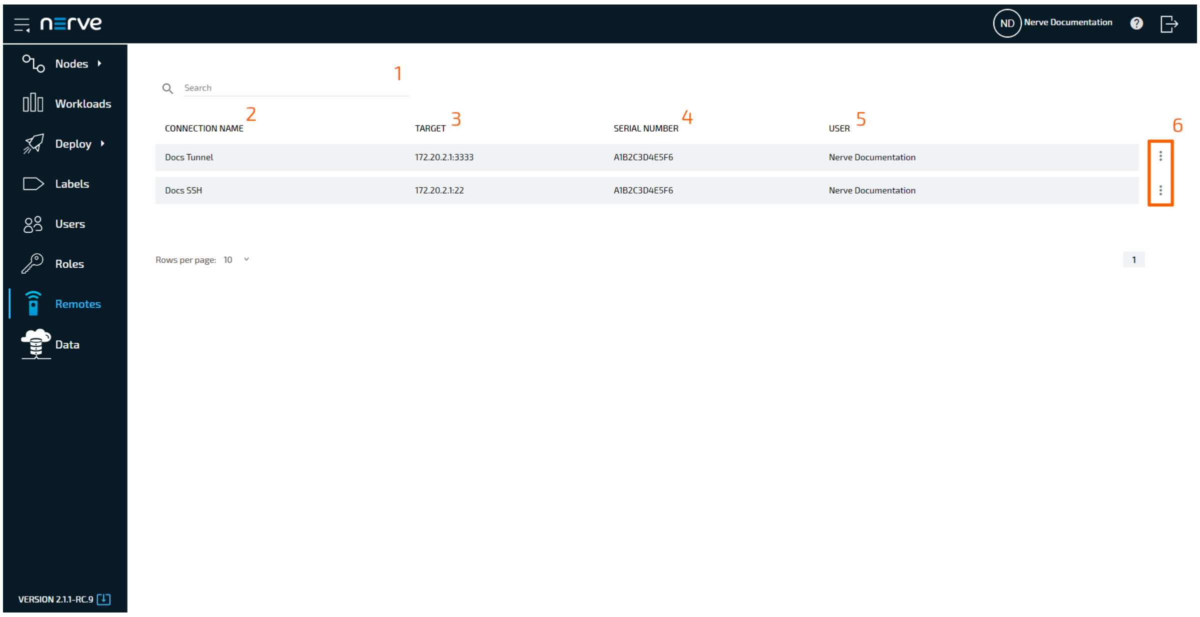
Remote connections are a fully integrated feature of Nerve Blue. They are available in two flavors: remote screens and remote tunnels.

- Remote screens are connections that are established between the Management System and a target. They are visualized by the Management System in a new browser tab and support SSH, RDP and VNC protocols.
- Remote tunnels are connections that are established from the local workstation to a target, similar to a VPN connection. They allow access to services and servers on the target from the user's local workstation. Remote tunnels are managed and established in the Nerve Connection Manager application and the Management System. The locally opened connection endpoint can then be used in a web browser, with SSH clients, or with remote desktop applications, depending on the target.

The targets of these remote connections can be nodes, workloads or external devices, which can be accessed from the node through the network.

Remote connections to workloads can be defined in existing workloads. Note that a workload does not have to be deployed again if a remote connection has been added. Defining a remote connection to a workload adds the remote connection to the workload across the Management System, meaning that it will also be available if the workload has already been deployed to nodes.

Select **Remotes** in the navigation on the left to view a list of currently established remote connections.

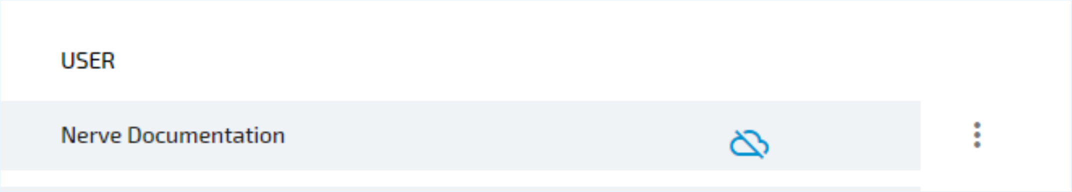


CONNECTION NAME	TARGET	SERIAL NUMBER	USER
Docs Tunnel	172.20.2.1:3333	A1B2C3D4E5F6	Nerve Documentation
Docs SSH	172.20.2.1:22	A1B2C3D4E5F6	Nerve Documentation

Item	Description
Search bar (1)	Use the search bar to filter remote connections by name or serial number.
CONNECTION NAME (2)	This is the name of the remote connection that is defined when the remote connection is configured.
TARGET (3)	This is the target of the remote connection. Note that this is not the name of the node. The hostname that was defined in the remote connection configuration is displayed here.
SERIAL NUMBER (4)	This is the serial number of the node to which the remote connection was established. In case of remote connections to workloads, the serial number of the node will be displayed to which the workload was deployed. For remote connections to external devices, the serial number of the node that the external device is connected to will be displayed.
USER (5)	This shows which user is using the established remote connection. If the same remote connection is used by two users, the remote connection will be listed again with a different user in te **User** column.
Ellipsis menu (6)	Clicking here opens an overlay that allows terminating connections.

NOTE

- Note that the list of active remote connections is not updated in real-time. Refresh the page to see changes.
- If a node goes offline while a remote connection is established, a cloud symbol will be displayed next to the remote connection to indicate that the connection to the node is interrupted.



Remote screens

A remote screen is established from the Management System to the target. It is opened in a new tab in the used web browser as soon as the remote connection is established.

Below are instructions on how to create SSH, VNC and RDP connections to nodes and workloads in the Management System.

NOTE

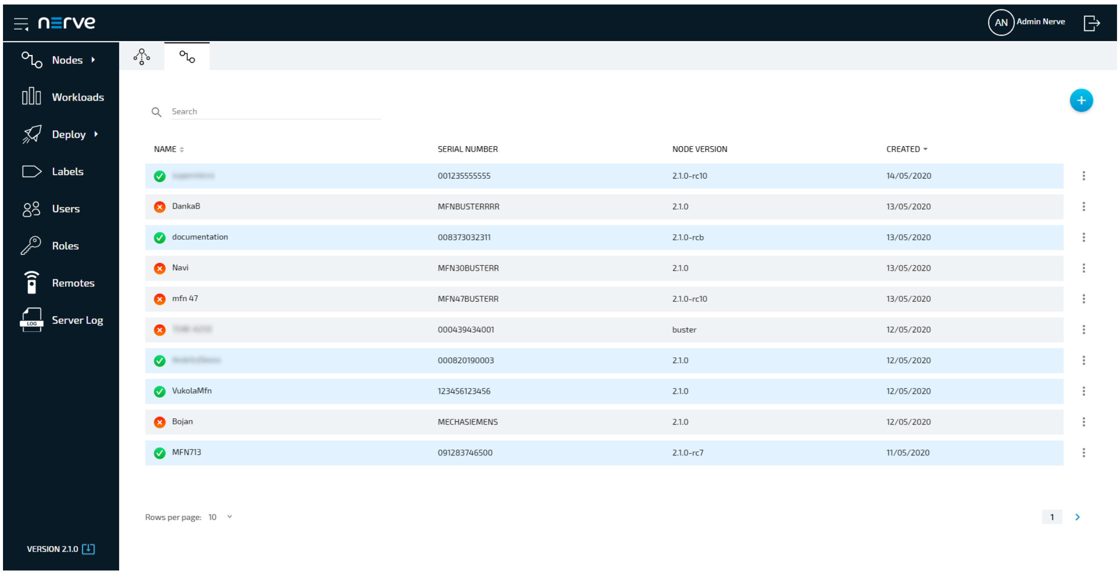
If the target of the remote connection is the host of the Nerve Blue system, use the IP address of the host: `172.20.2.1`. Using `localhost` is not supported.

Configuring an SSH connection to a node

An SSH connection to a node can be used for accessing the host operating system of the node or an external device connected to the node that is reachable through an SSH connection.

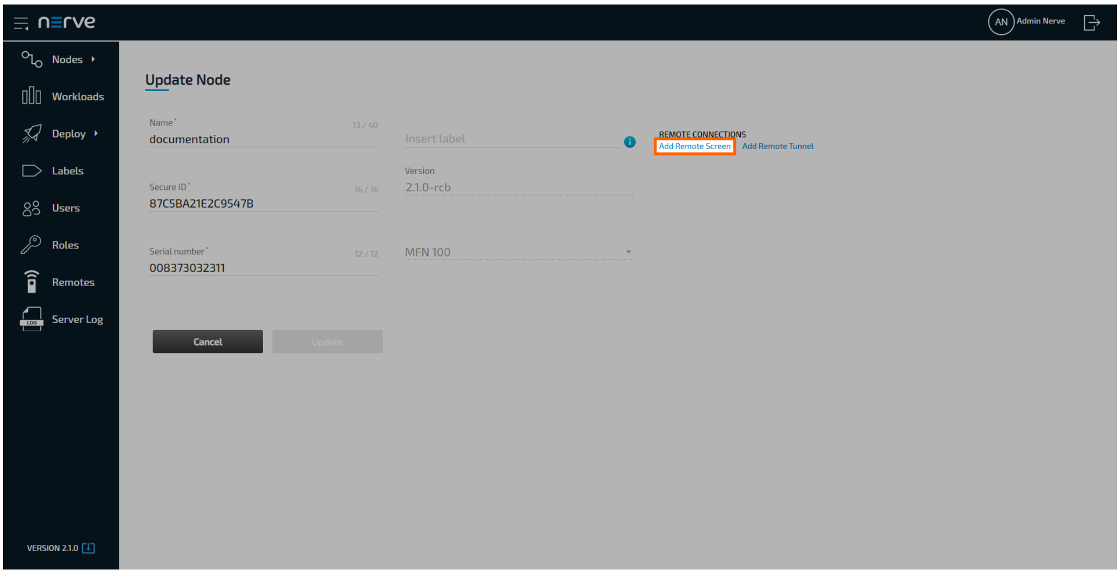
1. Select **Nodes** from the navigation on the left.
2. Select the nodes tab

   on the right to display the list of registered nodes.
3. Select a node from the list to which a remote connection will be established.



NAME	SERIAL NUMBER	NODE VERSION	CREATED
✓ [redacted]	001235555555	2.1.0-rc10	14/05/2020
✗ DankaB	MFNBUSTERRRR	2.1.0	13/05/2020
✓ documentation	008373032311	2.1.0-rcb	13/05/2020
✗ Navi	MFN30BUSTERR	2.1.0	13/05/2020
✗ mfn 47	MFN47BUSTERR	2.1.0-rc10	13/05/2020
✗ [redacted]	000439434001	buster	12/05/2020
✓ [redacted]	000820190003	2.1.0	12/05/2020
✓ VukolaMfn	123456123456	2.1.0	12/05/2020
✗ Bojan	MECHASIEMENS	2.1.0	12/05/2020
✓ MFN713	091283746500	2.1.0-rc7	11/05/2020

4. Click **Add Remote Screen** under **REMOTE CONNECTIONS** on the right side.



Update Node

Name* 13 / 40
documentation

Insert label

REMOTE CONNECTIONS
Add Remote Screen   Add Remote Tunnel

Version
2.1.0-rcb

Secure ID* 16 / 16
87C5BA21E2C9547B

Serial number* 12 / 12
008373032311

MFN 100

5. Enter a name for the remote connection in the new window.

NOTE

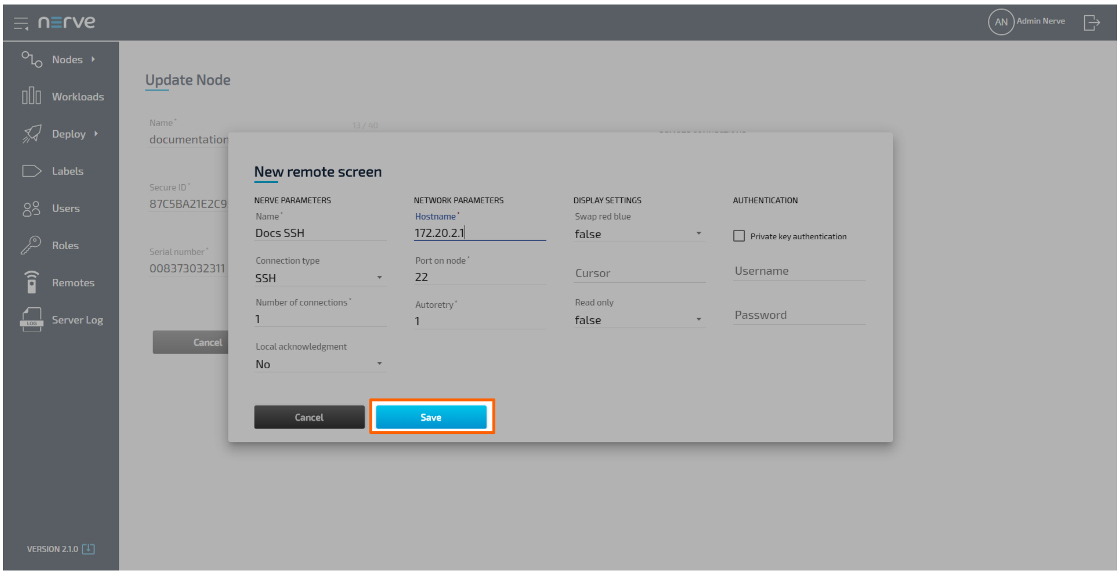
Make sure to use a unique name for every connection on a node to avoid confusion.

6. Select **SSH** from the drop-down menu under **Connection type**.
7. Enter the port used for SSH connection. The default port `22` is automatically filled in.
8. Enter the remaining information if applicable:

NERVE PARAMETERS	**Number of connections** Enter the maximum number of simultaneous connections. The default value is `1`. **Local acknowledgment** Select **Yes** or **No** from the drop-down menu. Selecting **Yes** will require approval of the remote connection in the Local UI before the connection can be established. If **No** is selected, the settings in the Local UI do not apply. Refer to Approving a remote connection for information on how to approve remote connections in the Local UI.
NETWORK PARAMETERS	**Hostname** Enter the IP address or the hostname of the target here. **Autoretry** Set the number of retries if the remote connection fails. The default value is `1`.
DISPLAY SETTINGS	The display settings offer configuration options that affect visualization. **Swap red blue** If colors appear to not be displayed correctly, select **true** from the drop-down menu. This can occur when using VNC servers. Select **false** otherwise. **Cursor** This setting determines if the cursor is rendered locally or remotely. Enter `local` for a local cursor or enter `remote` for a remote cursor. If set to `remote`, the mouse pointer will be rendered remotely, and the local position of the mouse pointer will be indicated by a small dot. A remote mouse cursor will have added input lag compared to a local cursor. However, a remote cursor might be necessary if the server does not support sending the cursor image to the client. **Read only** Select **true** or **false** from the drop-down menu. If set to **true**, no input will be accepted on the connection. Select **false** to allow input.

AUTHENTICATION	Enter **Username** and **Password** or tick the checkbox next to **Private key authentication** to use a private SSH key. Note that ticking the checkbox changes the interface. Enter the username and choose one of the methods to add the private SSH key: ◦ Click **Choose File** to open the local file browser and select the private SSH key file. ◦ Drag and drop the private SSH key file into the dotted line box saying **Drop Private Key Here**. ◦ Copy the private SSH key and paste it into the empty input field. Note that entering wrong login credentials will cause an error when the remote screen is established. If an error occurs, close the browser tab. Check the login credentials and re-establish the remote screen.

9. Select **Save** to add the remote connection.



Name* documentation

Secure ID* 87C5BA21E2C9

Serial number* 008373032311

New remote screen

NERVE PARAMETERS

Name* Docs SSH

Connection type SSH

Number of connections* 1

Local acknowledgment No

NETWORK PARAMETERS

Hostname* 172.20.2.1

Port on node* 22

Autoretry* 1

DISPLAY SETTINGS

Swap red blue false

Cursor

Read only false

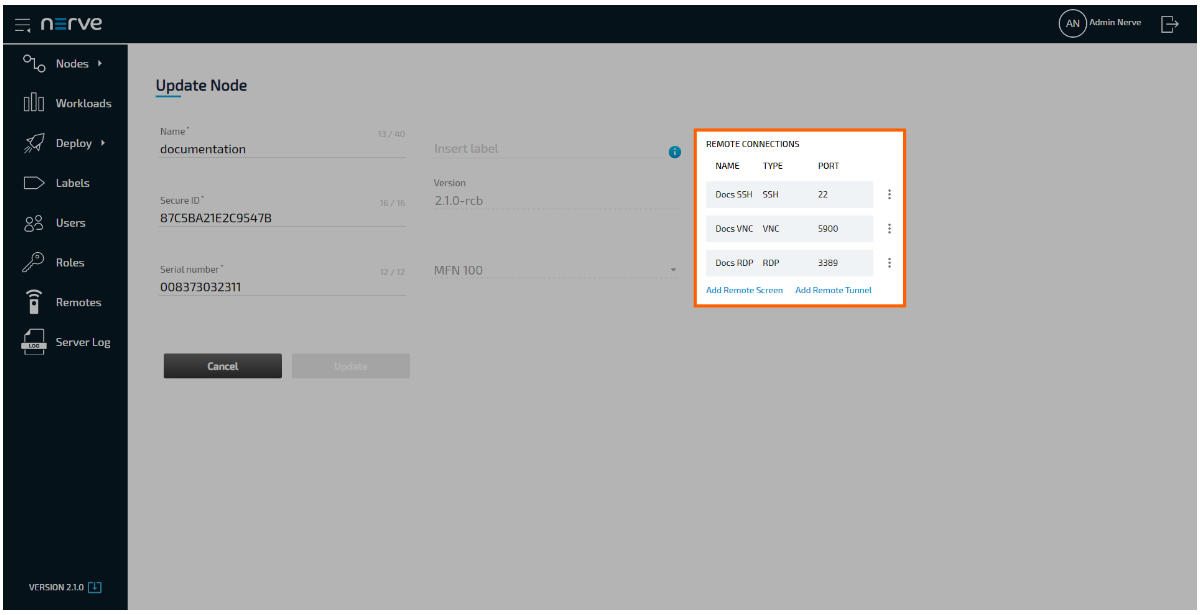
AUTHENTICATION

□ Private key authentication

Username

Password

The connection is now displayed under **REMOTE CONNECTIONS** on the right side, showing the **NAME**, **TYPE** and **PORT** of the remote connection.



Update Node

Name* 13 / 40: documentation

Insert label

Secure ID* 16 / 16: 87C5BA21E2C9547B

Version: 2.1.0-rcb

Serial number* 12 / 12: 008373032311

MFN 100

REMOTE CONNECTIONS

NAME	TYPE	PORT
Docs SSH	SSH	22
Docs VNC	VNC	5900
Docs RDP	RDP	3389

Add Remote Screen   Add Remote Tunnel

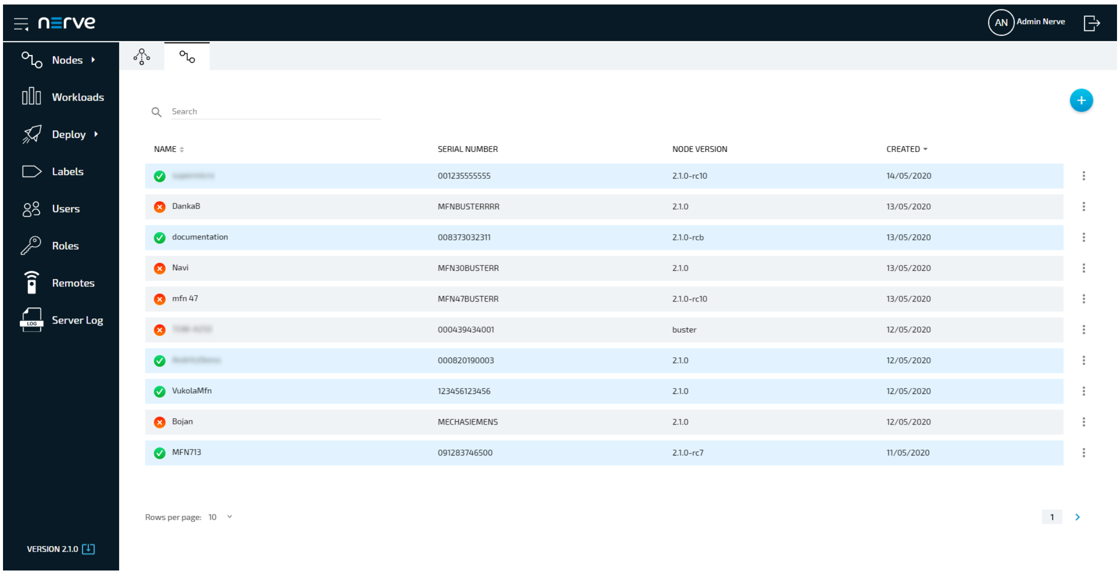
Cancel   Update

Configuring a VNC connection to a node

A VNC connection to a node can be used to connect to a Linux environment on an external device, which is connected to the node or to the same network that the node is connected to.

1. Select **Nodes** from the navigation on the left.
2. Select the nodes tab

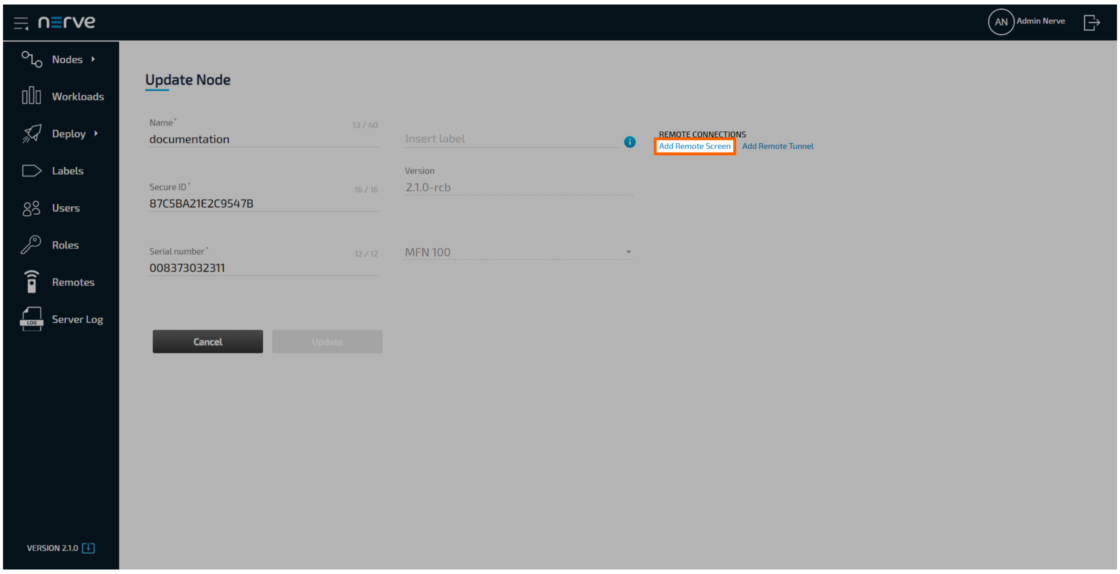
   on the right to display the list of registered nodes.
3. Select a node from the list to which a remote connection will be established.



NAME	SERIAL NUMBER	NODE VERSION	CREATED
✓ [redacted]	001235555555	2.1.0-rc10	14/05/2020
✗ DankaB	MFNBUSTERRRR	2.1.0	13/05/2020
✓ documentation	008373032311	2.1.0-rcb	13/05/2020
✗ Navi	MFN30BUSTERR	2.1.0	13/05/2020
✗ mfn 47	MFN47BUSTERR	2.1.0-rc10	13/05/2020
✗ [redacted]	000439434001	buster	12/05/2020
✓ [redacted]	000820190003	2.1.0	12/05/2020
✓ VukolaMfn	123456123456	2.1.0	12/05/2020
✗ Bojan	MECHASIEMENS	2.1.0	12/05/2020
✓ MFN713	091283746500	2.1.0-rc7	11/05/2020

Rows per page: 10   1 ›

4. Click **Add Remote Screen** under **REMOTE CONNECTIONS** on the right side.



Update Node

Name* 13 / 40: documentation

Insert label

Secure ID* 16 / 16: 87C5BA21E2C9547B

Version: 2.1.0-rcb

Serial number* 12 / 12: 008373032311

MFN 100

5. Enter a name for the remote connection in the new window.

> **NOTE**
>
> Make sure to use a unique name for every connection on a node to avoid confusion.

6. Select **VNC** from the drop-down menu under **Connection type**.

7. Enter the port used for VNC connection. The default port `5900` is automatically filled in.

8. Enter the password that was set for VNC connections at the target.

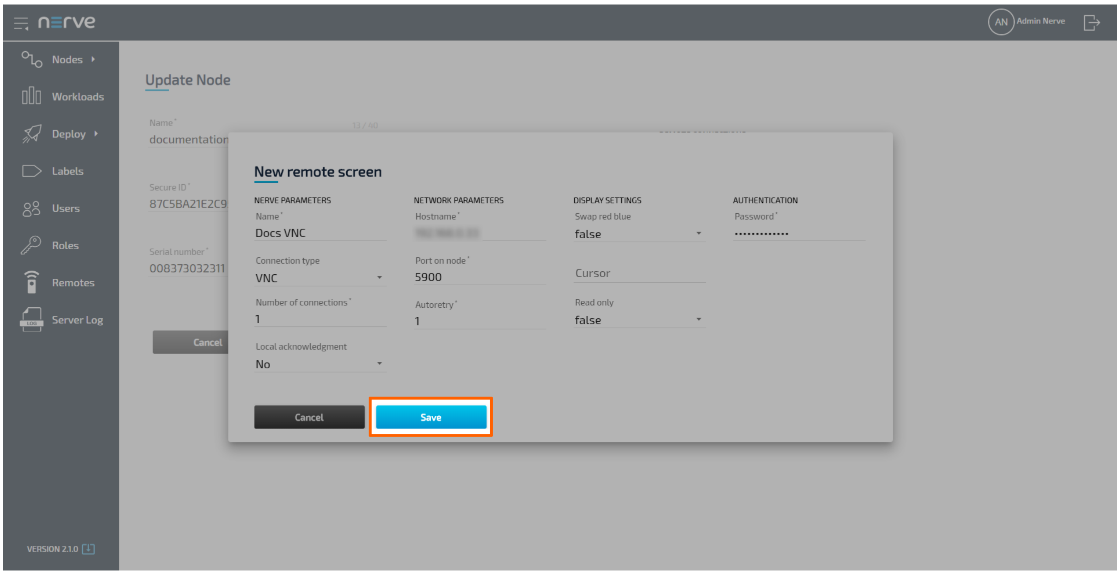
> **NOTE**
>
> Entering wrong login credentials will cause an error when the remote screen is established. If an error occurs, close the browser tab. Check the login credentials and re-establish the remote screen.

9. Enter the remaining information if applicable:

NERVE PARAMETERS	**Number of connections** Enter the maximum number of simultaneous connections. The default value is `1`. **Local acknowledgment** Select **Yes** or **No** from the drop-down menu. Selecting **Yes** will require approval of the remote connection in the Local UI before the connection can be established. If **No** is selected, the settings in the Local UI do not apply. Refer to Approving a remote connection for information on how to approve remote connections in the Local UI.

NETWORK PARAMETERS	**Hostname** Enter the IP address or the hostname of the target here. **Autoretry** Set the number of retries if the remote connection fails. The default value is `1`.
DISPLAY SETTINGS	The display settings offer configuration options that affect visualization. **Swap red blue** If colors appear to not be displayed correctly, select **true** from the drop-down menu. This can occur when using VNC servers. Select **false** otherwise. **Cursor** This setting determines if the cursor is rendered locally or remotely. Enter `local` for a local cursor or enter `remote` for a remote cursor. If set to `remote`, the mouse pointer will be rendered remotely, and the local position of the mouse pointer will be indicated by a small dot. A remote mouse cursor will have added input lag compared to a local cursor. However, a remote cursor might be necessary if the server does not support sending the cursor image to the client. **Read only** Select **true** or **false** from the drop-down menu. If set to **true**, no input will be accepted on the connection. Select **false** to allow input.

10. Select **Save** to add the remote connection.



Update Node

Name* documentation 13 / 40

Secure ID* 87C5BA21E2C9

Serial number* 008373032311

New remote screen

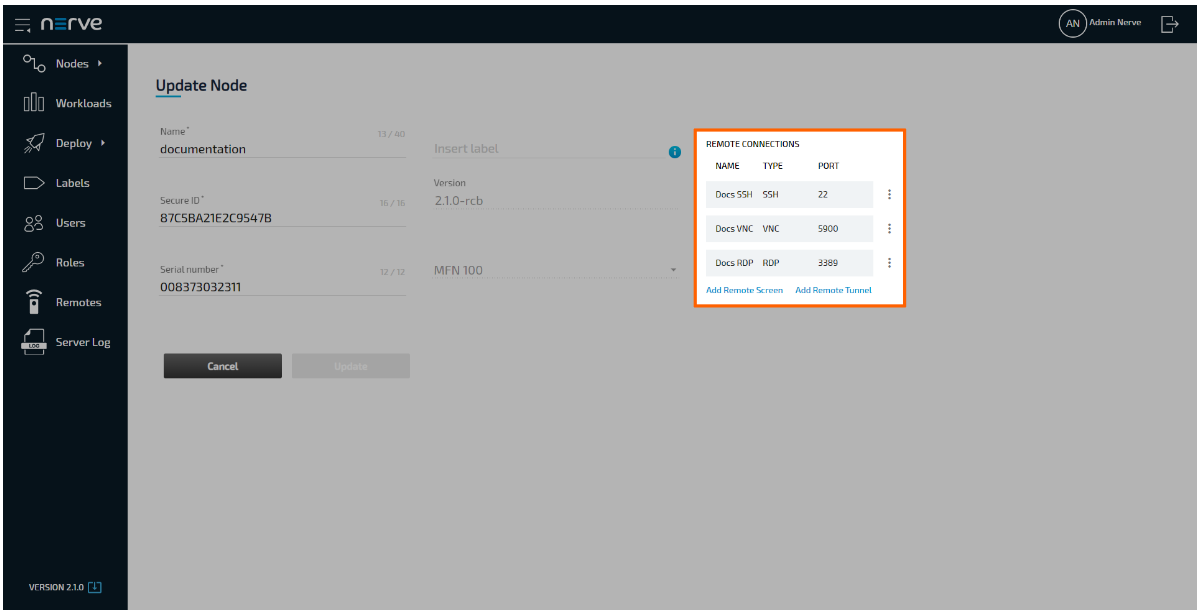
NERVE PARAMETERS
Name* Docs VNC
Connection type VNC
Number of connections* 1
Local acknowledgment No

NETWORK PARAMETERS
Hostname* [redacted]
Port on node* 5900
Autoretry* 1

DISPLAY SETTINGS
Swap red blue false
Cursor
Read only false

AUTHENTICATION
Password* ·············

The connection is now displayed under **REMOTE CONNECTIONS** on the right side, showing the **NAME**, **TYPE** and **PORT** of the remote connection.



Update Node

Name* 13 / 40
documentation

Insert label

Secure ID* 16 / 16
87C5BA21E2C9547B

Version
2.1.0-rcb

Serial number* 12 / 12
008373032311

MFN 100

REMOTE CONNECTIONS

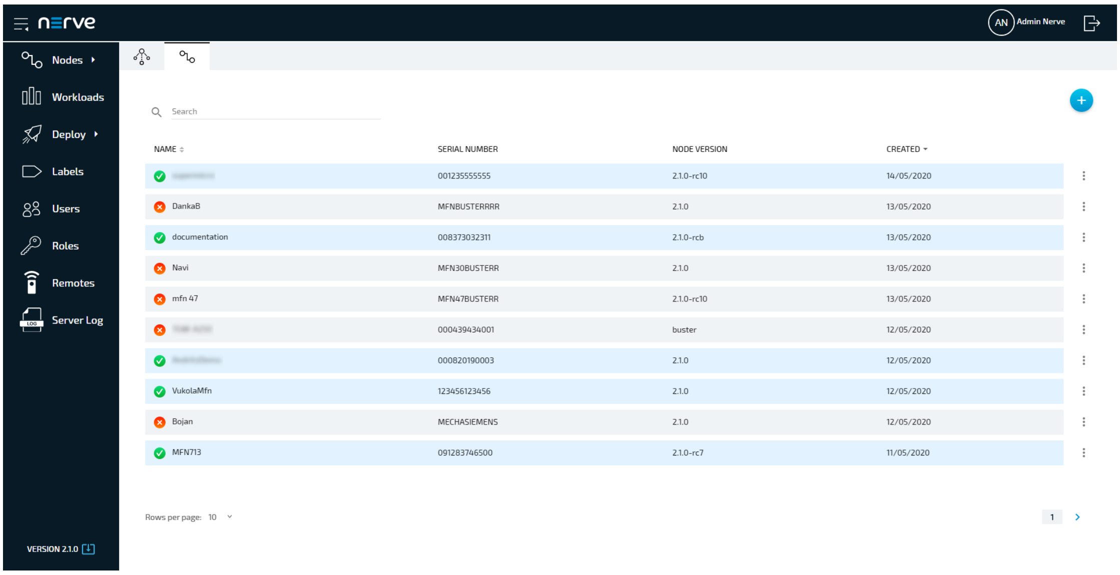
NAME	TYPE	PORT
Docs SSH	SSH	22
Docs VNC	VNC	5900
Docs RDP	RDP	3389

Configuring an RDP connection to a node

An RDP connection to a node can be used to connect to a Windows environment on an external device, which is connected to the node or to the same network that the node is connected to.

1. Select **Nodes** from the navigation on the left.
2. Select the nodes tab

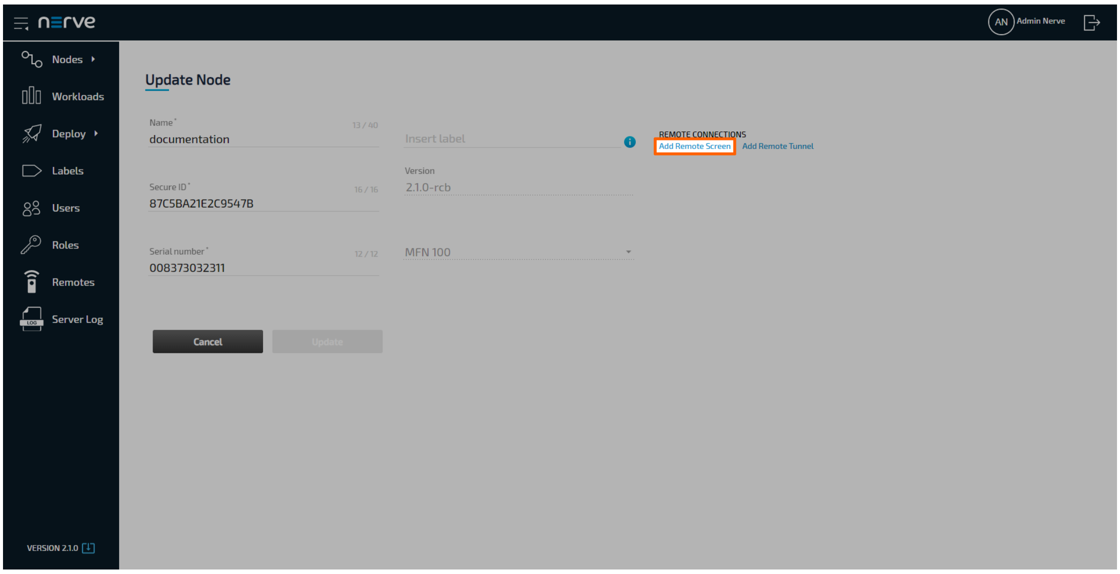
   on the right to display the list of registered nodes.
3. Select a node from the list to which a remote connection will be established.



NAME	SERIAL NUMBER	NODE VERSION	CREATED
[redacted]	001235555555	2.1.0-rc10	14/05/2020
DankaB	MFNBUSTERRRR	2.1.0	13/05/2020
documentation	008373032311	2.1.0-rcb	13/05/2020
Navi	MFN30BUSTERR	2.1.0	13/05/2020
mfn 47	MFN47BUSTERR	2.1.0-rc10	13/05/2020
[redacted]	000439434001	buster	12/05/2020
[redacted]	000820190003	2.1.0	12/05/2020
VukolaMfn	123456123456	2.1.0	12/05/2020
Bojan	MECHASIEMENS	2.1.0	12/05/2020
MFN713	091283746500	2.1.0-rc7	11/05/2020

Rows per page: 10 1

4. Click **Add Remote Screen** under **REMOTE CONNECTIONS** on the right side.



Update Node

Name * 13 / 40: documentation

Insert label

Version: 2.1.0-rcb

Secure ID * 16 / 16: 87C5BA21E2C9547B

Serial number * 12 / 12: 008373032311

MFN 100

5. Enter a name for the remote connection in the new window.

> **NOTE**
>
> Make sure to use a unique name for every connection on a node to avoid confusion.

6. Select **RDP** from the drop-down menu under **Connection type**.

7. Enter the port used for RDP connection. The default port `3389` is automatically filled in.

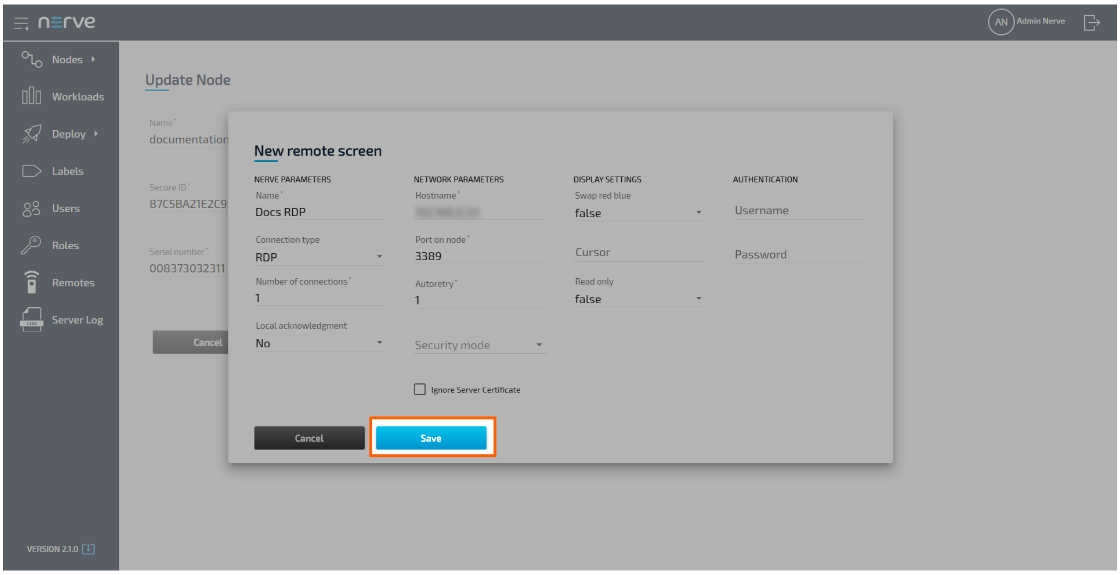
8. Enter the remaining information if applicable:

NERVE PARAMETERS	**Number of connections** Enter the maximum number of simultaneous connections. The default value is `1`. **Local acknowledgment** Select **Yes** or **No** from the drop-down menu. Selecting **Yes** will require approval of the remote connection in the Local UI before the connection can be established. If **No** is selected, the settings in the Local UI do not apply. Refer to Approving a remote connection for information on how to approve remote connections in the Local UI.

NETWORK PARAMETERS	**Hostname** Enter the IP address or the hostname of the target here. **Autoretry** Set the number of retries if the remote connection fails. The default value is `1`. **Security mode** This mode dictates how data will be encrypted and what type of authentication will be performed, if any. Select an option from the drop-down menu. Possible values are: ◦ **ANY** This is the default if the field is left blank. Automatically select the security mode based on the security protocols supported by both the client and the server. ◦ **NLA (Network Level Authentication)** This mode uses TLS encryption and requires the username and password to be given in advance. Unlike RDP mode, the authentication step is performed before the remote desktop session actually starts, avoiding the need for the Windows server to allocate significant resources for users that may not be authorized. ◦ **RDP encryption** This is the standard RDP encryption. It is generally only used for older Windows servers or in cases where a standard Windows login screen is desired. Newer versions of Windows have this mode disabled by default and will only accept NLA unless explicitly configured otherwise. ◦ **TLS encryption** Select this for RDP authentication and encryption implemented via TLS (Transport Layer Security). The TLS security mode is primarily used in load balanced configurations where the initial RDP server may redirect the connection to a different RDP server. **Ignore Server Certificate** If checked, the certificate returned by the server will be ignored, even if that certificate cannot be validated. This is useful if the server and the connection to the server is universally trusted, and if the server's certificate cannot be validated (for example, if it is self-signed).

DISPLAY SETTINGS	The display settings offer configuration options that affect visualization. **Swap red blue** If colors appear to not be displayed correctly, select **true** from the drop-down menu. This can occur when using VNC servers. Select **false** otherwise. **Cursor** This setting determines if the cursor is rendered locally or remotely. Enter `local` for a local cursor or enter `remote` for a remote cursor. If set to `remote`, the mouse pointer will be rendered remotely, and the local position of the mouse pointer will be indicated by a small dot. A remote mouse cursor will have added input lag compared to a local cursor. However, a remote cursor might be necessary if the server does not support sending the cursor image to the client. **Read only** Select **true** or **false** from the drop-down menu. If set to **true**, no input will be accepted on the connection. Select **false** to allow input.
AUTHENTICATION	Enter **Username** and **Password** for Windows login. Note that entering wrong login credentials will cause an error when the remote screen is established. If an error occurs, close the browser tab. Check the login credentials and re-establish the remote screen.

9. Select **Save** to add the remote connection.



Update Node

Name: documentation

Secure ID: 87C5BA21E2C9

Serial number: 008373032311

New remote screen

NERVE PARAMETERS

Name: Docs RDP

Connection type: RDP

Number of connections: 1

Local acknowledgment: No

NETWORK PARAMETERS

Hostname: [redacted]

Port on node: 3389

Autoretry: 1

Security mode

□ Ignore Server Certificate

DISPLAY SETTINGS

Swap red blue: false

Cursor

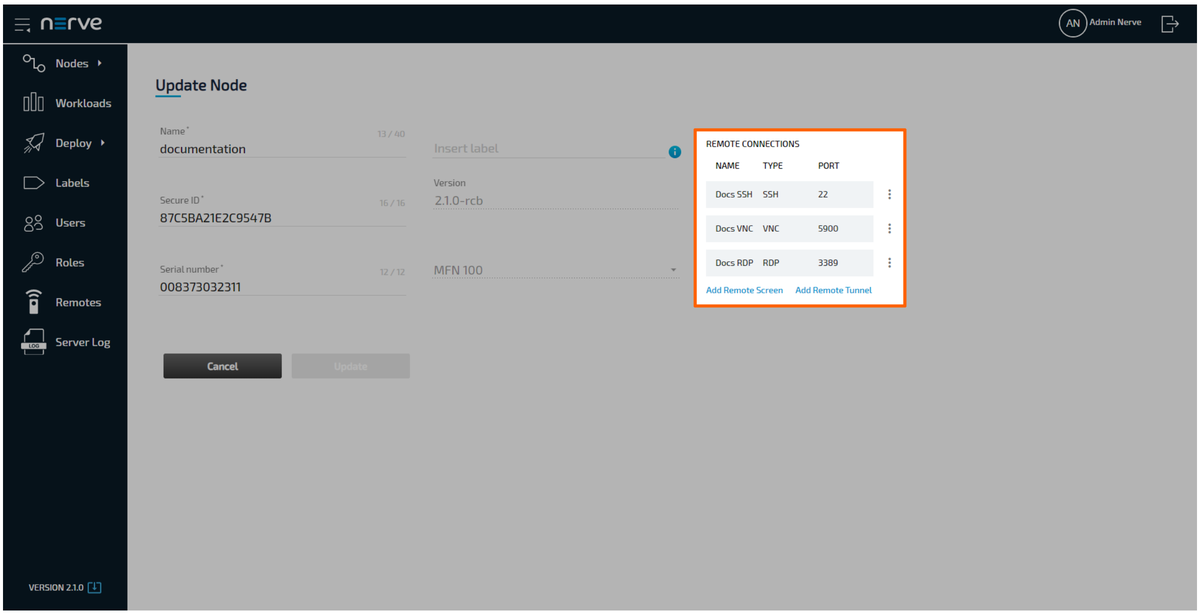
Read only: false

AUTHENTICATION

Username

Password

The connection is now displayed under **REMOTE CONNECTIONS** on the right side, showing the **NAME**, **TYPE** and **PORT** of the remote connection.



Update Node

Name * 13 / 40
documentation

Insert label

Secure ID * 16 / 16
87C5BA21E2C9547B

Version
2.1.0-rcb

Serial number * 12 / 12
008373032311

MFN 100

REMOTE CONNECTIONS

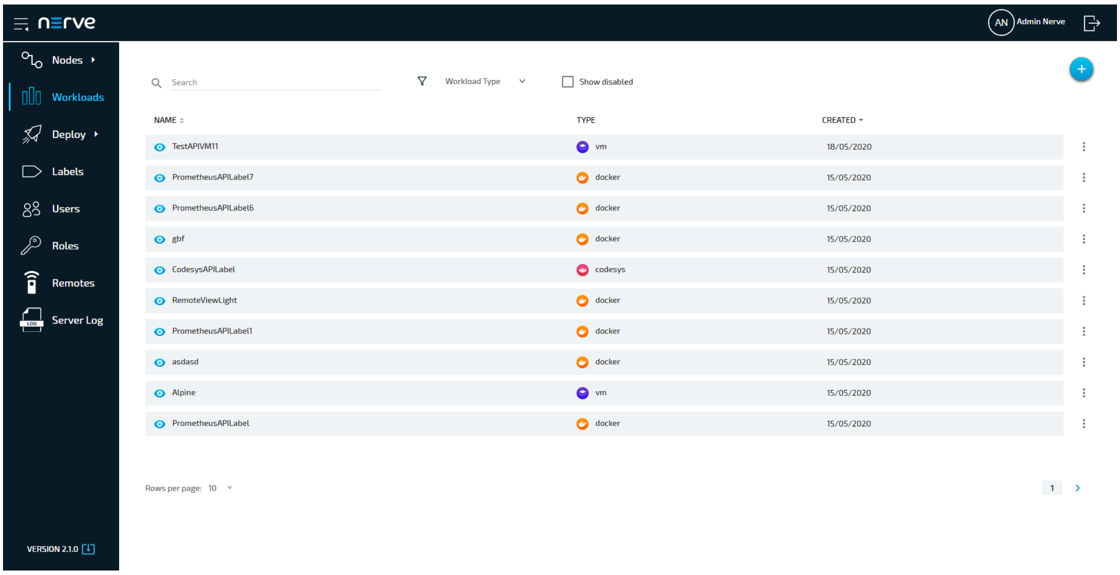
NAME	TYPE	PORT
Docs SSH	SSH	22
Docs VNC	VNC	5900
Docs RDP	RDP	3389

Add Remote Screen Add Remote Tunnel

Configuring a remote screen to a workload

A remote screen to a workload can be configured, regardless of a workload being deployed or not. Configuring a remote screen for a workload will immediately add the remote screen to the workload on all nodes that it has been deployed to. Note that remote screens to CODESYS workloads cannot be established.

1. Select **Workloads** in the navigation on the left.
2. Select a workload from the list.



NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

Rows per page: 10

3. Select the workload version to which the remote connection will be added.

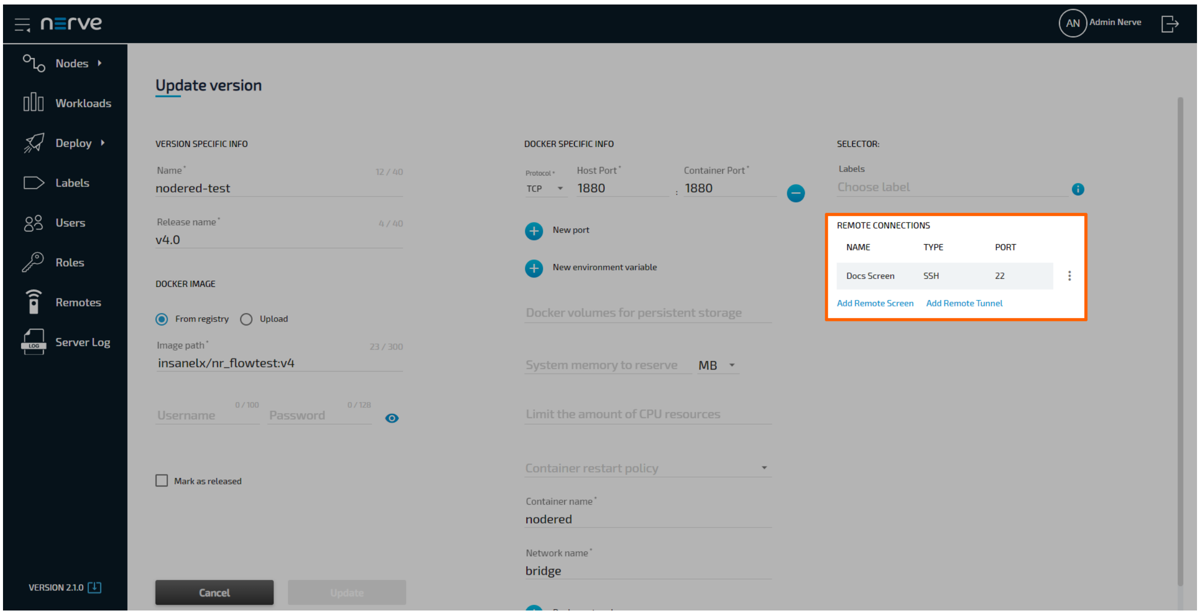
NOTE

Note that the configured remote connection will only be available for the version that was selected.

4. Click **Add Remote Screen** under **REMOTE CONNECTIONS** on the right side.
5. Follow steps 4 to 9 in the instructions above for SSH, VNC, or RDP connections.

> **NOTE**
>
> Note that adding the hostname is not required when configuring a remote screen to a Docker workload. The system automatically detects the hostname when the workload is deployed.
>
> In case of Virtual Machine workloads, the hostname entry is not displayed for VNC connections. For SSH and RDP connections, enter the IP address or hostname under **VM hostname / IP**.

The connection is saved and now displayed under **REMOTE CONNECTIONS** on the right side, showing the **NAME**, **TYPE** and **PORT** of the remote connection.



Update version

VERSION SPECIFIC INFO

Name* nodered-test (12 / 40)

Release name* v4.0 (4 / 40)

DOCKER IMAGE

● From registry ○ Upload

Image path* insanelx/nr_flowtest:v4 (23 / 300)

Username (0 / 100) Password (0 / 128)

□ Mark as released

DOCKER SPECIFIC INFO

Protocol* TCP Host Port* 1880 : Container Port* 1880

Docker volumes for persistent storage

System memory to reserve MB

Limit the amount of CPU resources

Container restart policy

Container name* nodered

Network name* bridge

SELECTOR:

Labels: Choose label

REMOTE CONNECTIONS

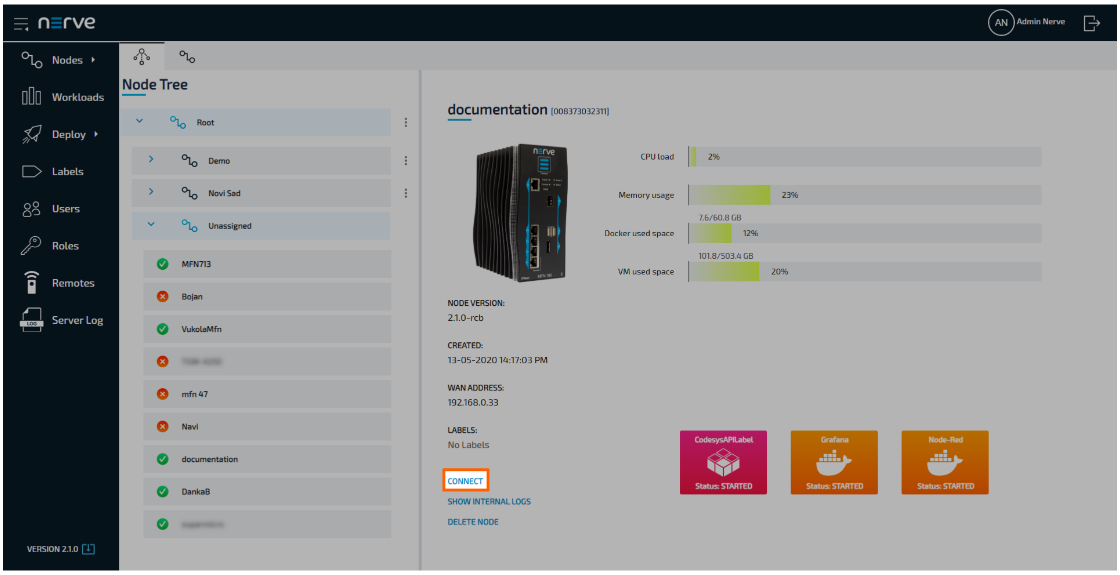
NAME	TYPE	PORT
Docs Screen	SSH	22

Add Remote Screen Add Remote Tunnel

Using a remote screen to a node or external device

Established remote screens are listed under **Remotes** in the navigation on the left until they are terminated.

1. Select **Nodes** in the navigation on the left.
2. Select the node tree tab on the right to display registered nodes in the node tree.
3. Select a node with a remote screen from the node tree.
4. Click **CONNECT** in the node details on the right.



documentation [008373032311]

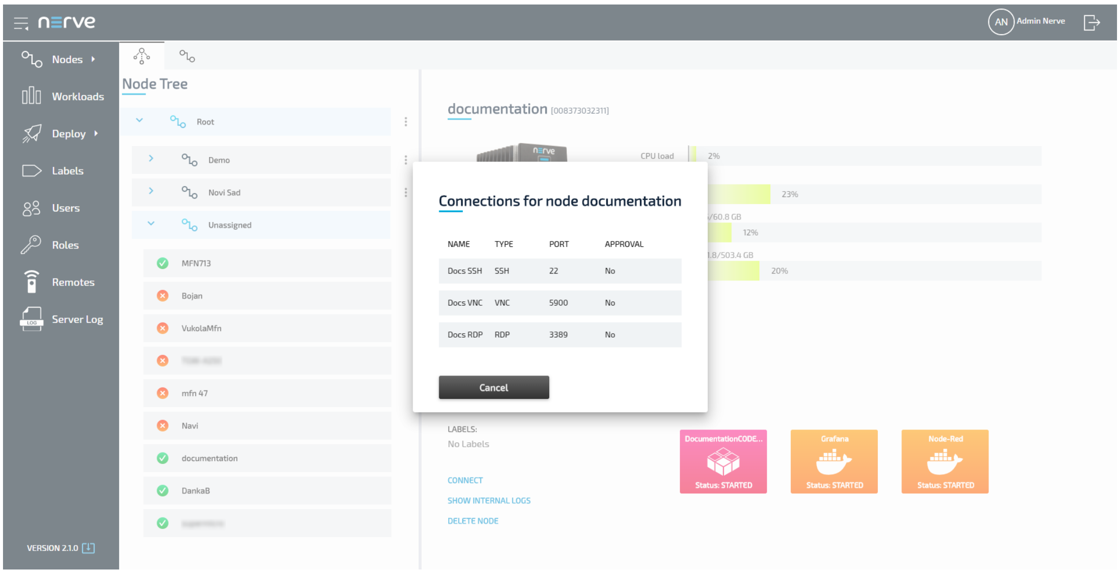
NODE VERSION:
2.1.0-rcb

CREATED:
13-05-2020 14:17:03 PM

WAN ADDRESS:
192.168.0.33

LABELS:
No Labels

5. Select the remote connection from the list in the new window.



Connections for node documentation

NAME	TYPE	PORT	APPROVAL
Docs SSH	SSH	22	No
Docs VNC	VNC	5900	No
Docs RDP	RDP	3389	No

The remote screen will be opened and displayed in a new browser tab after a few seconds if **Local acknowledgement** has been set to **No**. If set to **Yes**, the remote connection has to be approved in the Local UI. Refer to Approving a remote connection for more information.

NOTE

Make sure not to exceed the defined number of connections of the same remote screen. This causes an error and the connection has to be terminated and established again. If there is a connection error, close the tab, terminate and re-establish the connection.

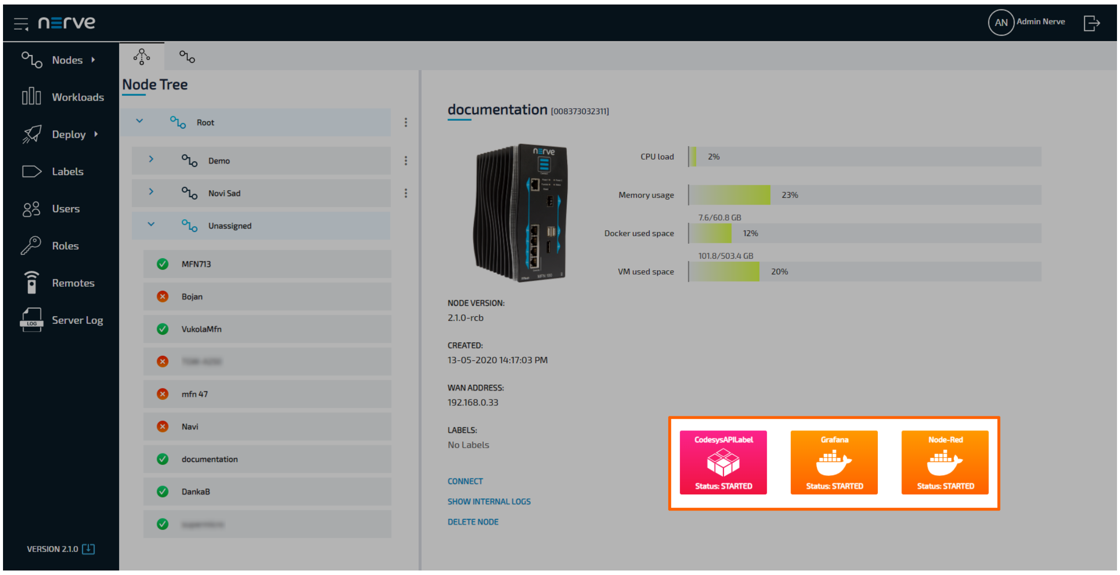
Using a remote screen to a workload

Established remote screens are listed under **Remotes** in the navigation on the left until they are terminated.

1. Select **Nodes** in the navigation on the left.

2. Select the node tree tab

   on the right to display registered nodes in the node tree.
3. Select the node that has a deployed workload with a remote connection.
4. Select the workload.



documentation [008373032311]

CPU load 2%

Memory usage 23%

7.6/60.8 GB

Docker used space 12%

101.8/503.4 GB

VM used space 20%

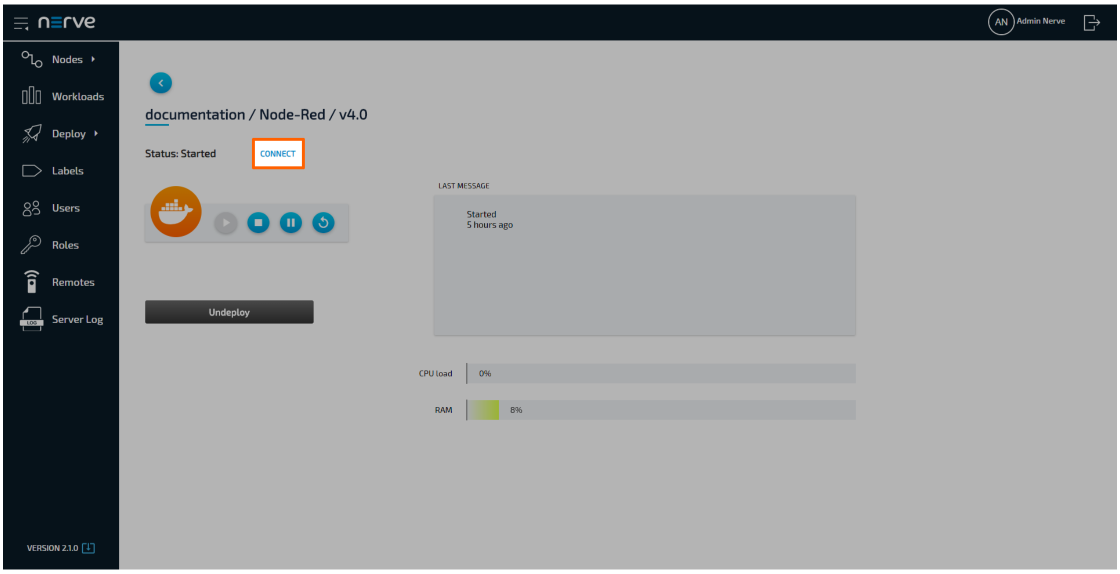
NODE VERSION:
2.1.0-rcb

CREATED:
13-05-2020 14:17:03 PM

WAN ADDRESS:
192.168.0.33

LABELS:
No Labels

5. Click **CONNECT** next to the workload status.



documentation / Node-Red / v4.0

Status: Started

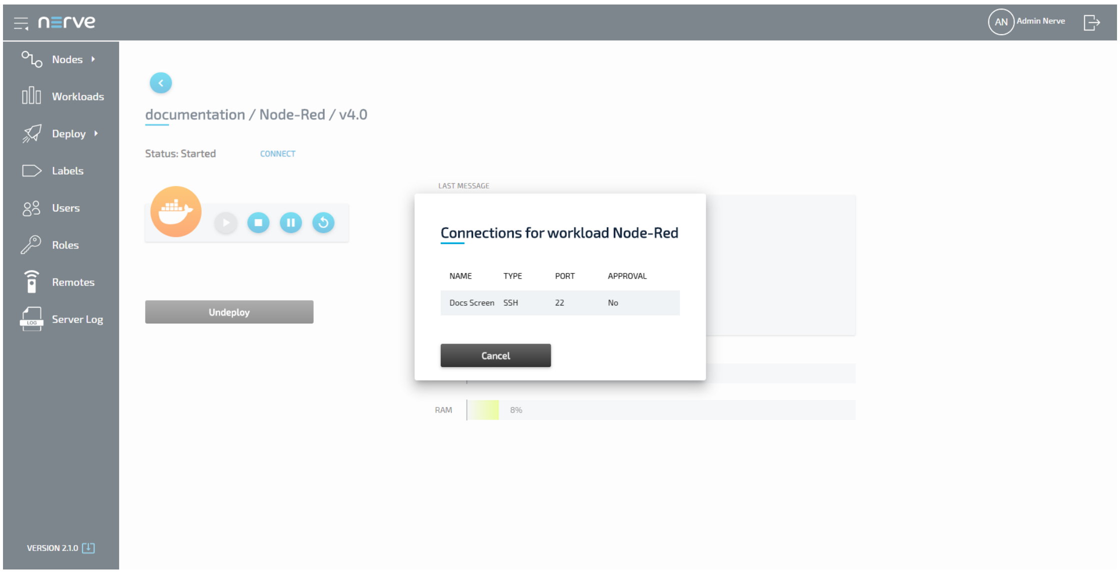
LAST MESSAGE

Started
5 hours ago

CPU load 0%

RAM 8%

6. Select the remote connection from the list in the new window.



documentation / Node-Red / v4.0

Status: Started

Connections for workload Node-Red

NAME	TYPE	PORT	APPROVAL
Docs Screen	SSH	22	No

The remote screen will be opened and displayed in a new browser tab after a few seconds if **Local acknowledgement** has been set to **No**. If set to **Yes**, the remote connection has to be approved in the Local UI. Refer to Approving a remote connection for more information.

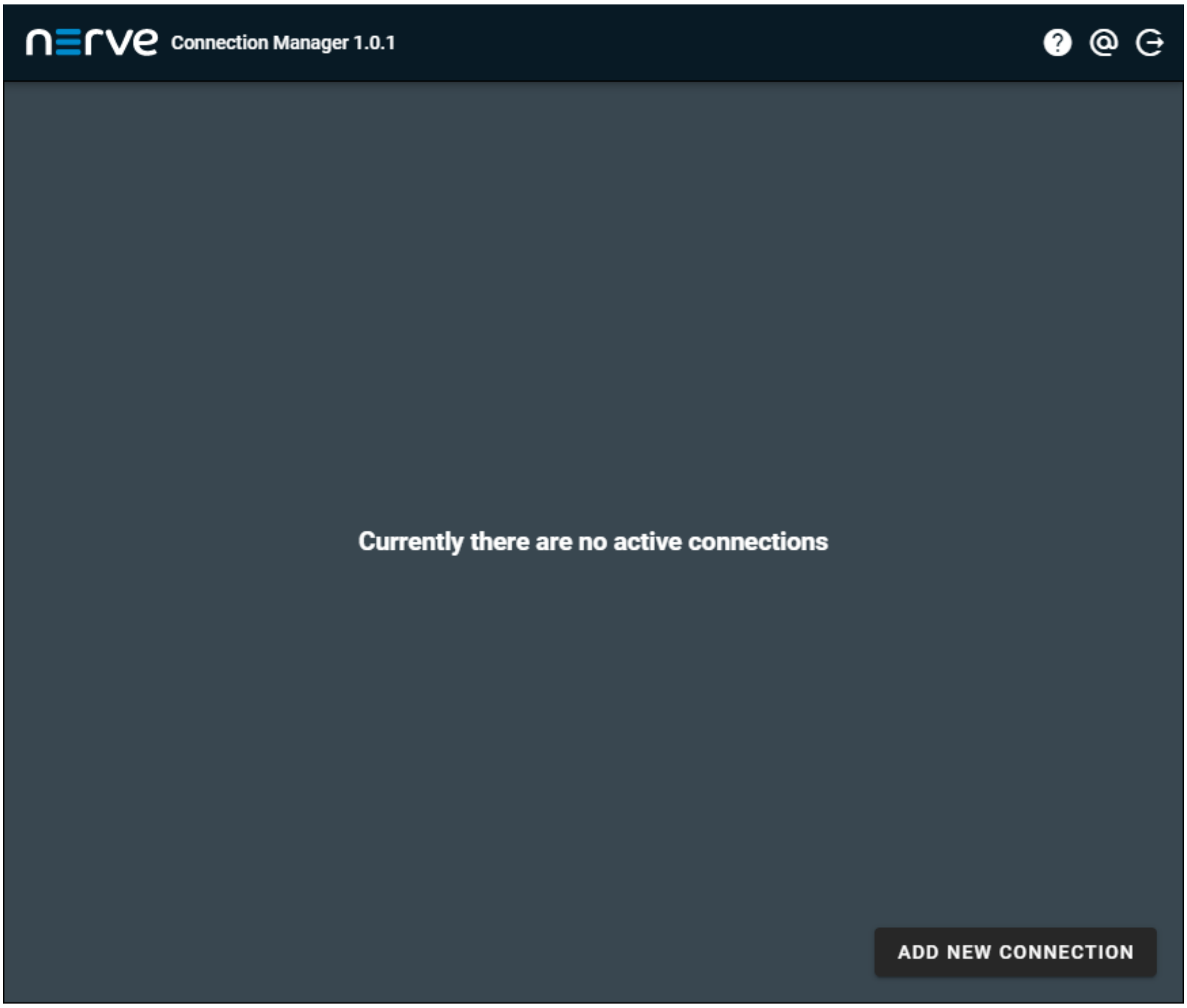
NOTE

Make sure not to exceed the defined number of connections of the same remote screen. This causes an error and the connection has to be terminated and established again. If there is a connection error, close the tab, terminate and re-establish the connection.

Remote tunnels

The Nerve Connection Manager is an application that is installed locally on the workstation. It is required for establishing and using remote connections from the local workstation. Download the Nerve Connection Manager from the Nerve Software Center first.

The Nerve Connection Manager installation file is an executable file. Open the installation file and follow the installation process. The filename of the installation file is `Nerve Connection Manager Setup <version>.exe` on Windows or `Nerve Connection Manager Setup <version>.deb` on Linux.



nerve Connection Manager 1.0.1
Currently there are no active connections
ADD NEW CONNECTION

Once installed, the Nerve Connection Manager will be associated with `nerverm://` links that are generated in the Management System. Clicking such a link will automatically open the Nerve Connection Manager.

NOTE

If the target of the remote connection is the host of the Nerve Blue system, use the IP address of the host: `172.20.2.1`. Using `localhost` is not supported.

Compatibility of the Nerve Connection Manager

Make sure that the correct version of the Nerve Connection Manager is installed according to the version of the Management System that is used:

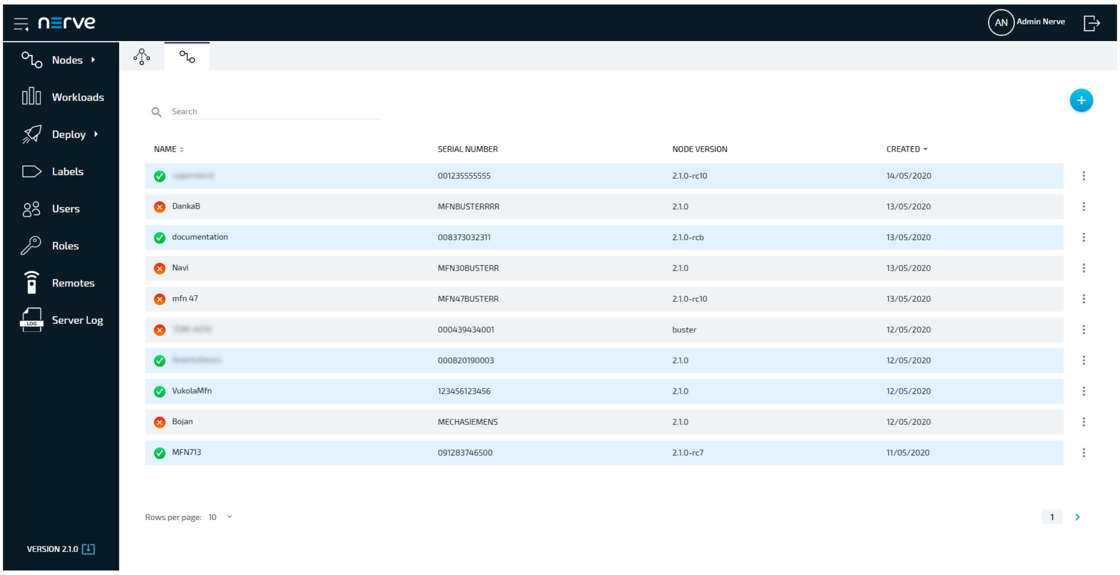
Management System	Nerve Connection Manager
v2.1.X	v1.0.1
v2.1.0	v1.0.0

Configuring a remote tunnel to a node

Depending on the target, a remote tunnel to a node can be used in a web browser, with SSH clients, or with remote desktop applications, for example.

1. Select **Nodes** in the navigation on the left.
2. Select the nodes tab

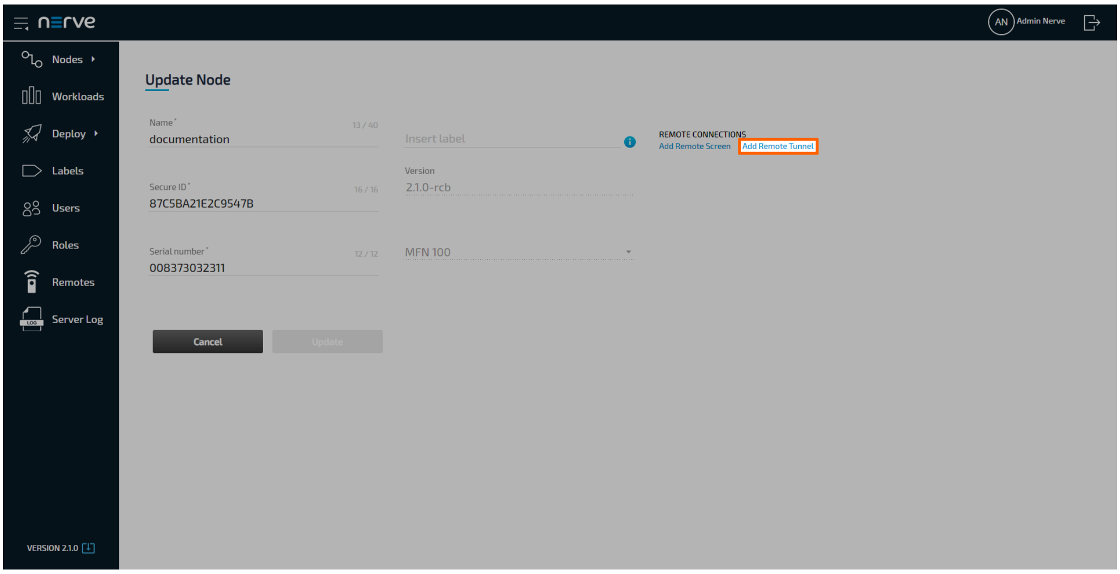
   on the right to display the list of registered nodes.
3. Select a node from the list.



NAME	SERIAL NUMBER	NODE VERSION	CREATED
✓ [redacted]	001235555555	2.1.0-rc10	14/05/2020
✗ DankaB	MFNBUSTERRRR	2.1.0	13/05/2020
✓ documentation	008373032311	2.1.0-rcb	13/05/2020
✗ Navi	MFN30BUSTERR	2.1.0	13/05/2020
✗ mfn 47	MFN47BUSTERR	2.1.0-rc10	13/05/2020
✗ [redacted]	000439434001	buster	12/05/2020
✓ [redacted]	000820190003	2.1.0	12/05/2020
✓ VukolaMfn	123456123456	2.1.0	12/05/2020
✗ Bojan	MECHASIEMENS	2.1.0	12/05/2020
✓ MFN713	091283746500	2.1.0-rc7	11/05/2020

Rows per page: 10

4. Select **Add Remote Tunnel** under **REMOTE CONNECTIONS** on the right side.



Update Node

Name* 13 / 40
documentation

Insert label

Secure ID* 16 / 16
87C5BA21E2C9547B

Version
2.1.0-rcb

Serial number* 12 / 12
008373032311

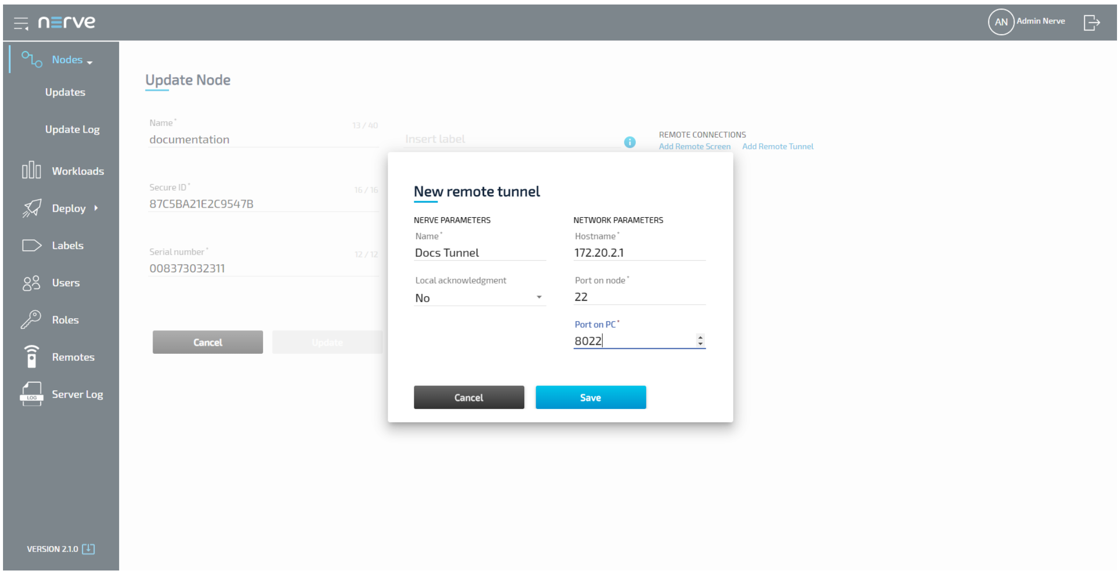
MFN 100

REMOTE CONNECTIONS
Add Remote Screen Add Remote Tunnel

5. Enter the following information:

NERVE PARAMETERS	**Name** Enter a name for the remote connection. Make sure to use a unique name for every connection on a node to avoid confusion. **Local acknowledgment** Select **Yes** or **No** from the drop-down menu. Selecting **Yes** will require approval of the remote connection in the Local UI before the connection can be established. If **No** is selected, the settings in the Local UI do not apply. Refer to Approving a remote connection for information on how to approve remote connections in the Local UI.
NETWORK PARAMETERS	**Hostname** Enter the IP address or the hostname of the target here. **Port on node** Enter the port the target listens on. **Port on PC** Enter the port that will be used for communication on the local workstation. The port entered here serves as a default port that can be changed in the Nerve Connection Manager in case it is already in use. Note that some systems might restrict usage of ports under 1024. This is true for Linux systems especially. Enter port numbers higher than 1024 to avoid possible port conflicts.

6. Select **Save** to save the remote connection configuration.



Update Node

Name* 13 / 40
documentation

Secure ID* 16 / 16
87C5BA21E2C9547B

Serial number* 12 / 12
008373032311

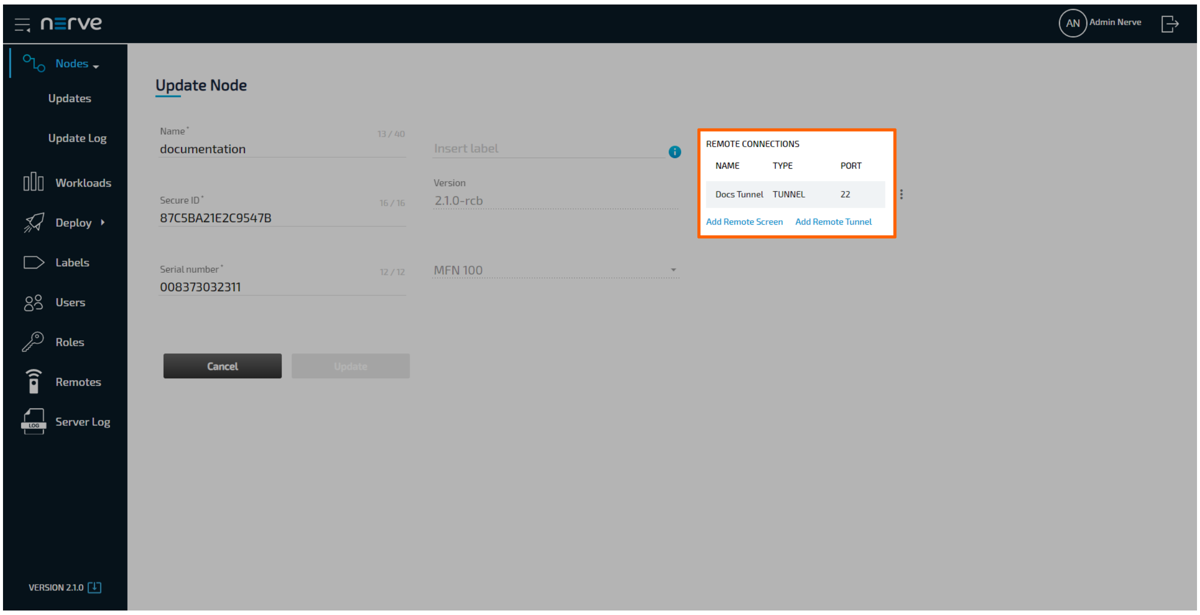
REMOTE CONNECTIONS

New remote tunnel

NERVE PARAMETERS
Name*
Docs Tunnel
Local acknowledgment
No

NETWORK PARAMETERS
Hostname*
172.20.2.1
Port on node*
22
Port on PC*
8022

The connection is saved and now displayed under **REMOTE CONNECTIONS** on the right side, showing the **NAME**, **TYPE** and **PORT** of the remote connection.



Update Node

Name* 13 / 40
documentation

Secure ID* 16 / 16
87C5BA21E2C9547B

Serial number* 12 / 12
008373032311

Insert label

Version
2.1.0-rcb

MFN 100

REMOTE CONNECTIONS

NAME	TYPE	PORT
Docs Tunnel	TUNNEL	22

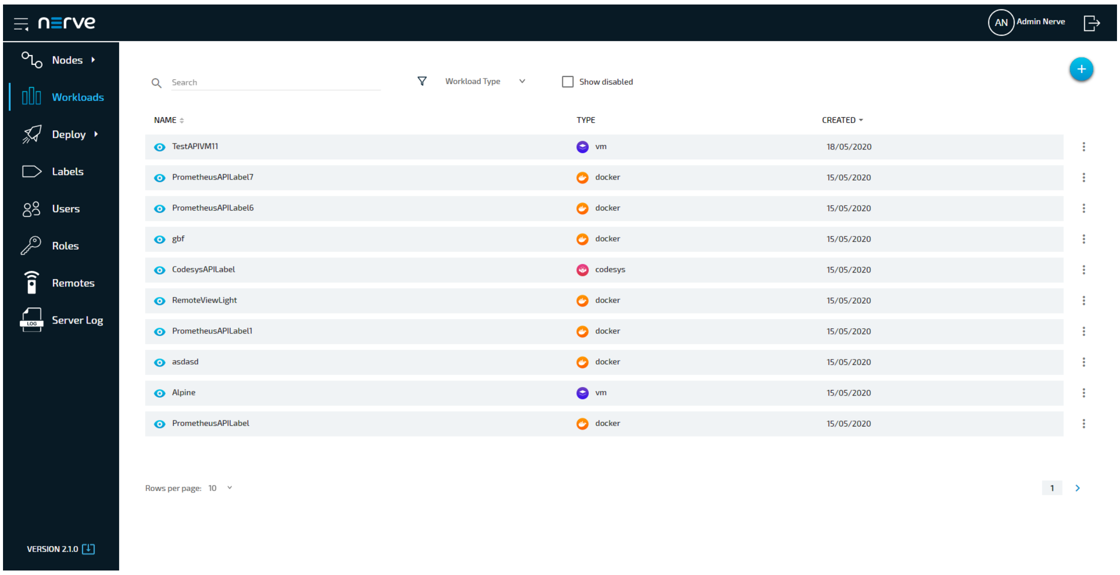
Add Remote Screen Add Remote Tunnel

Configuring a remote tunnel to a workload

A remote tunnel to a workload can be configured, regardless of a workload being deployed or not. Configuring a remote tunnel for a workload will immediately add the remote tunnel to the workload on all nodes that it has been deployed to. Depending on the target, a remote tunnel to a workload can be used in a web browser, with SSH clients, or with remote desktop applications, for example.

1. Select **Workloads** in the navigation on the left.

2. Select a workload from the list.



NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

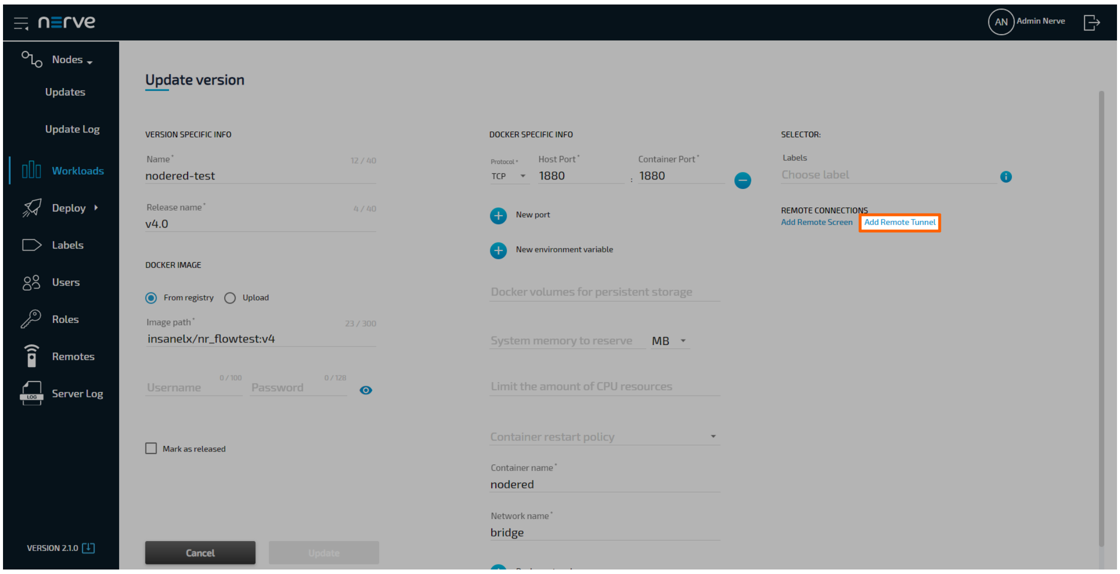
Rows per page: 10

1

3. Select the workload version to which a remote connection will be established.

> **NOTE**
>
> Note that the configured remote connection will only be available for the version that was selected.

4. Select **Add Remote Tunnel** under **REMOTE CONNECTIONS** on the right side.



Update version

VERSION SPECIFIC INFO

Name* nodered-test 12 / 40

Release name* v4.0 4 / 40

DOCKER IMAGE

● From registry ○ Upload

Image path* insanelx/nr_flowtest:v4 23 / 300

Username 0 / 100 Password 0 / 128

□ Mark as released

DOCKER SPECIFIC INFO

Protocol* TCP Host Port* 1880 : Container Port* 1880

Docker volumes for persistent storage

System memory to reserve MB

Limit the amount of CPU resources

Container restart policy

Container name* nodered

Network name* bridge

SELECTOR:

Labels Choose label

REMOTE CONNECTIONS

Add Remote Screen Add Remote Tunnel

5. Enter the following information:

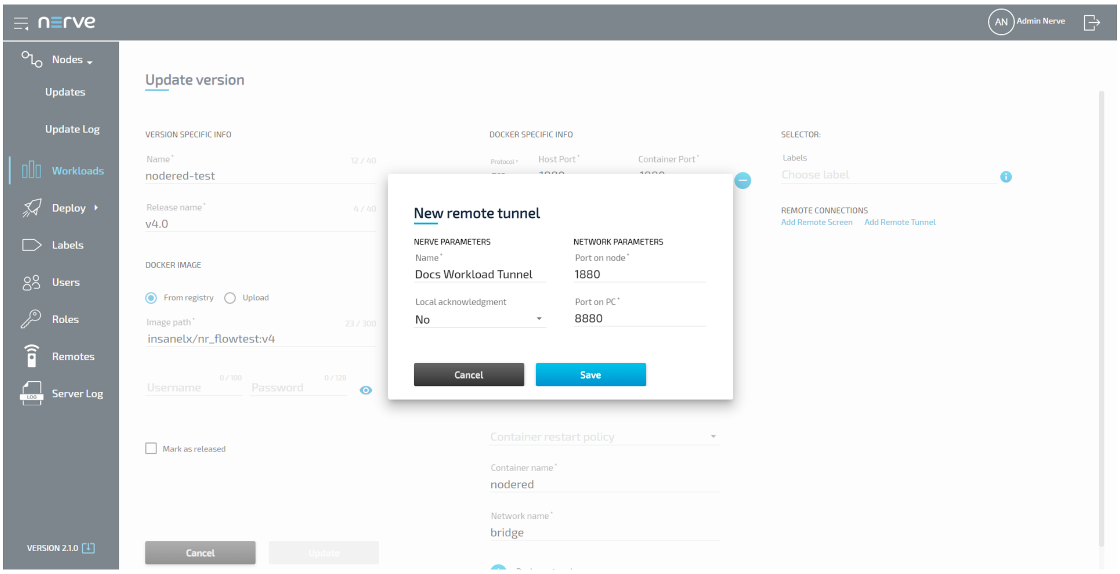
NERVE PARAMETERS	**Name** Enter a name for the remote connection. Make sure to use a unique name for every connection on a node to avoid confusion. **Local acknowledgment** Select **Yes** or **No** from the drop-down menu. Selecting **Yes** will require approval of the remote connection in the Local UI before the connection can be established. If **No** is selected, the settings in the Local UI do not apply. Refer to Approving a remote connection for information on how to approve remote connections in the Local UI.
NETWORK PARAMETERS	**Port on node** Enter the port the target listens on. **Port on PC** Enter the port that will be used for communication on the local workstation. The port entered here serves as a default port that can be changed in the Nerve Connection Manager in case it is already in use. Note that some systems might restrict usage of ports under 1024. This is true for Linux systems especially. Enter port numbers higher than 1024 to avoid possible port conflicts.

NOTE

Note that adding the hostname is not required when configuring a remote tunnel to a Docker workload. The system automatically detects the hostname when the workload is deployed.

For CODESYS workloads, the **Hostname** and **Port on node** fields are filled in by the default. They contain the IP address and default port of the CODESYS runtime.

6. Select **Save** to save the remote connection configuration.



New remote tunnel

NERVE PARAMETERS

Name: Docs Workload Tunnel

Local acknowledgment: No

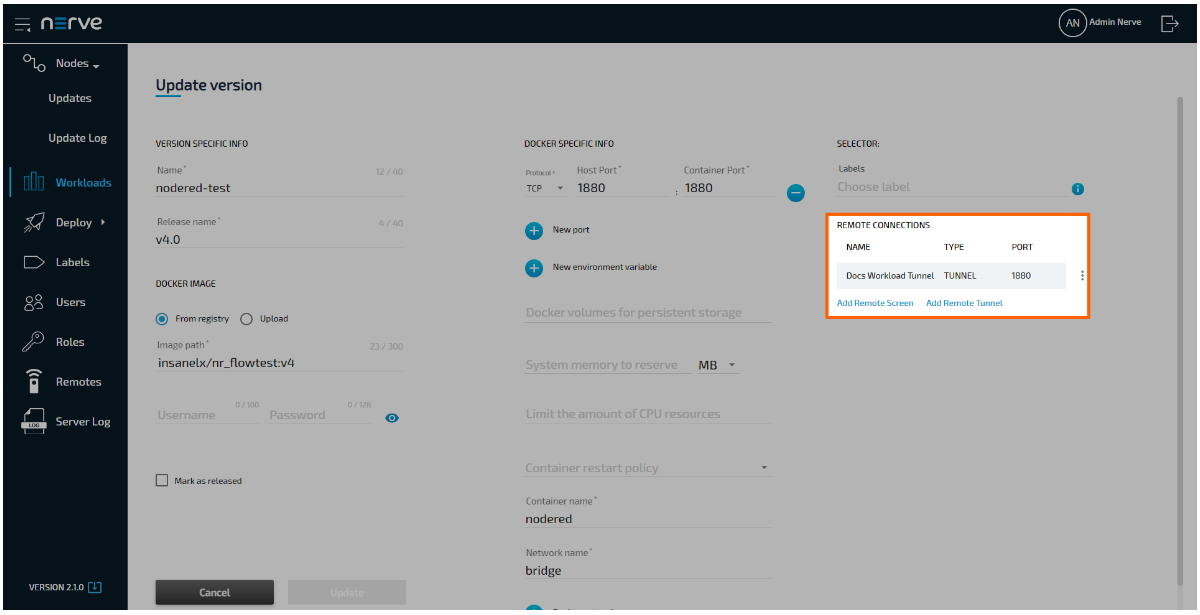
NETWORK PARAMETERS

Port on node: 1880

Port on PC: 8880

Cancel | Save

The connection is saved and now displayed under **REMOTE CONNECTIONS** on the right side, showing the **NAME**, **TYPE** and **PORT** of the remote connection.



VERSION SPECIFIC INFO

Name: nodered-test 12 / 40

Release name: v4.0 4 / 40

DOCKER IMAGE

● From registry ○ Upload

Image path: insanelx/nr_flowtest:v4 23 / 300

□ Mark as released

DOCKER SPECIFIC INFO

Protocol: TCP Host Port: 1880 Container Port: 1880

Container name: nodered

Network name: bridge

SELECTOR:

Labels: Choose label

REMOTE CONNECTIONS

NAME	TYPE	PORT
Docs Workload Tunnel	TUNNEL	1880

Using a remote tunnel to a node or external device

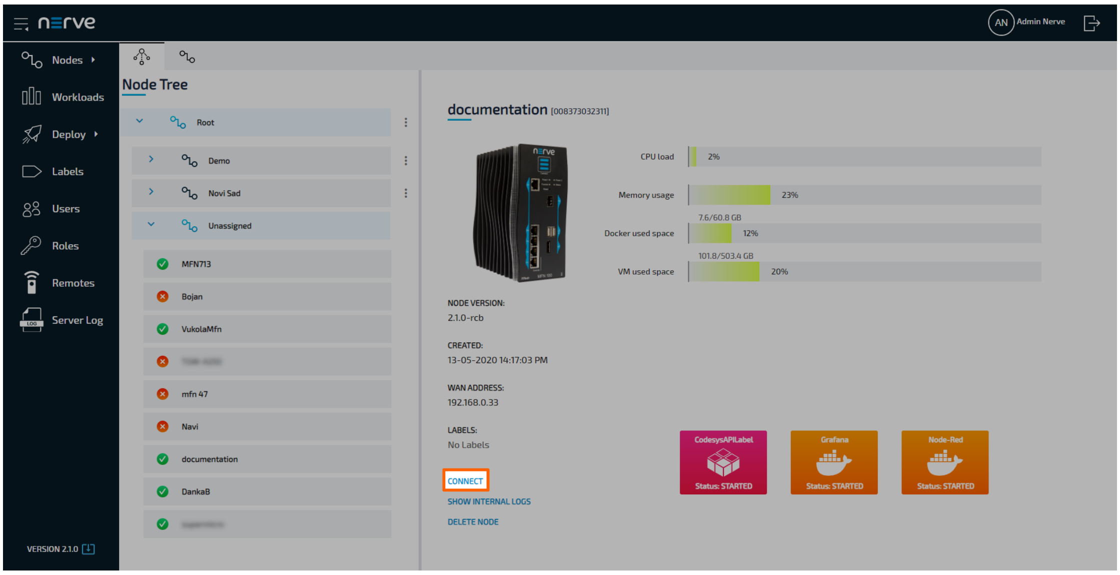
Note that the Nerve Connection Manager is required to use a remote tunnel. Download the Nerve Connection Manager from the Nerve Software Center and install it first.

Established remote tunnels are listed under **Remotes** in the navigation on the left until they are terminated.

1. Select **Nodes** in the navigation on the left.
2. Select the node tree tab

   on the right to display registered nodes in the node tree.
3. Select a node with a remote tunnel from the node tree.

4. Click **CONNECT** in the node details on the right.



documentation [008373032311]

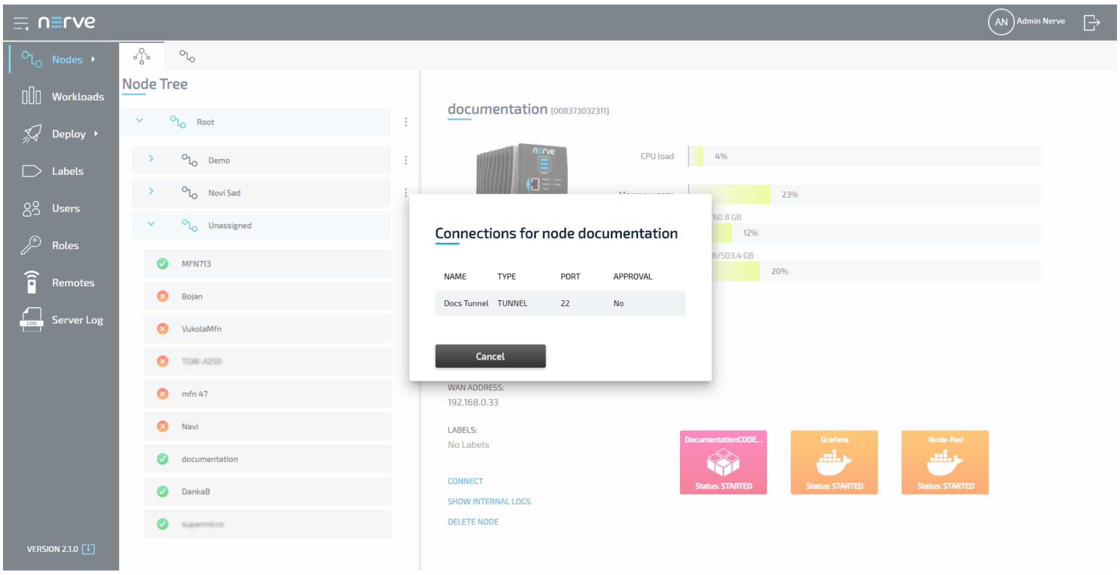
NODE VERSION:
2.1.0-rcb

CREATED:
13-05-2020 14:17:03 PM

WAN ADDRESS:
192.168.0.33

LABELS:
No Labels

5. Select the remote connection from the list in the new window. Note that remote tunnels have the type **TUNNEL**.



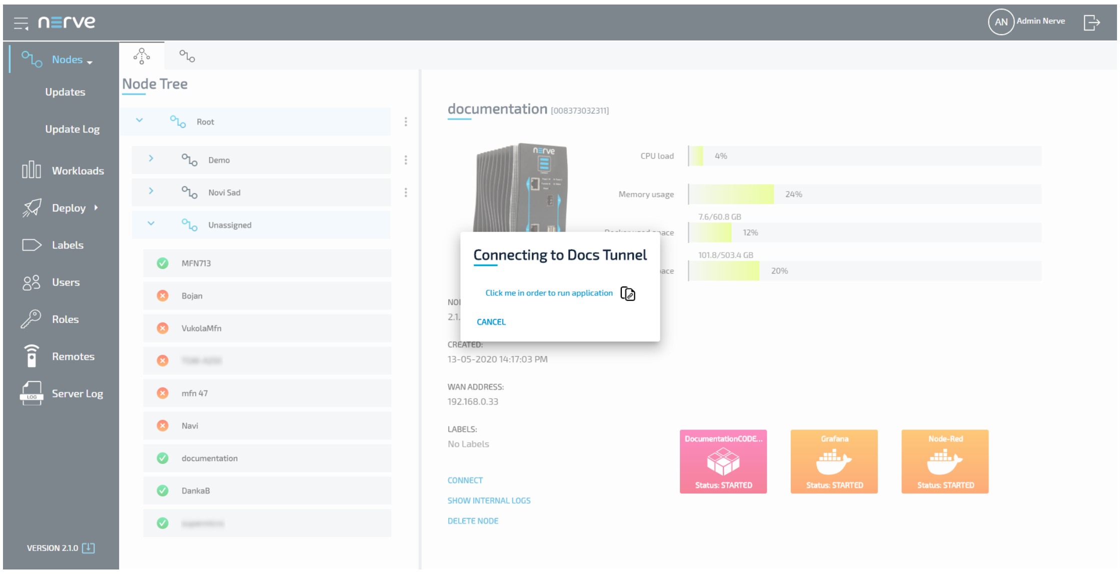
Connections for node documentation

NAME	TYPE	PORT	APPROVAL
Docs Tunnel	TUNNEL	22	No

WAN ADDRESS:
192.168.0.33

LABELS:
No Labels

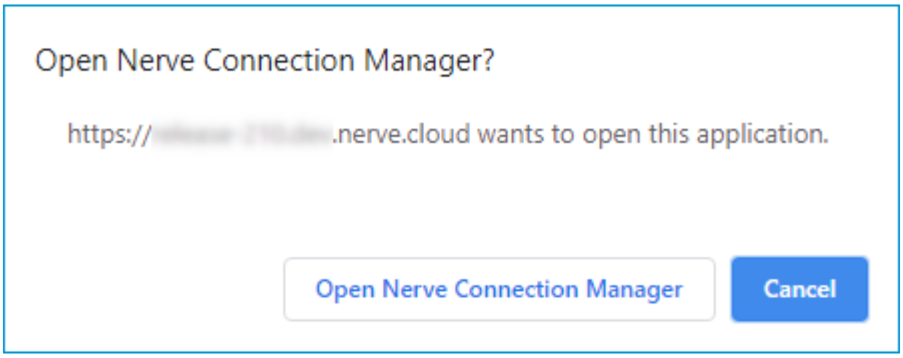
6. Select **Click me in order to run application** in the new window.



NOTE

If **Local acknowledgment** is set to **Yes**, the Management System will wait for approval until the remote connection has been locally approved before displaying the window above. Refer to Approving a remote connection for more information.

7. If the Nerve Connection Manager installed correctly, confirm the browser message that the Nerve Connection Manager shall be opened.
   Depending on the browser that is used, this message will differ. The Nerve Connection Manager will start automatically once the message is confirmed.

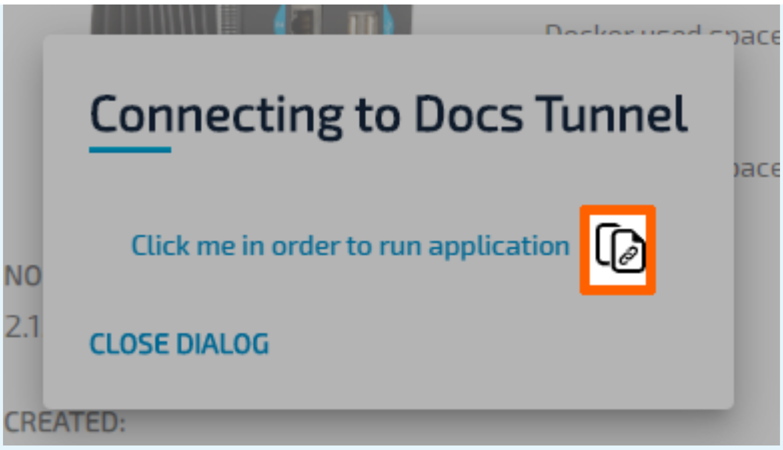


Open Nerve Connection Manager?

https://[redacted].nerve.cloud wants to open this application.

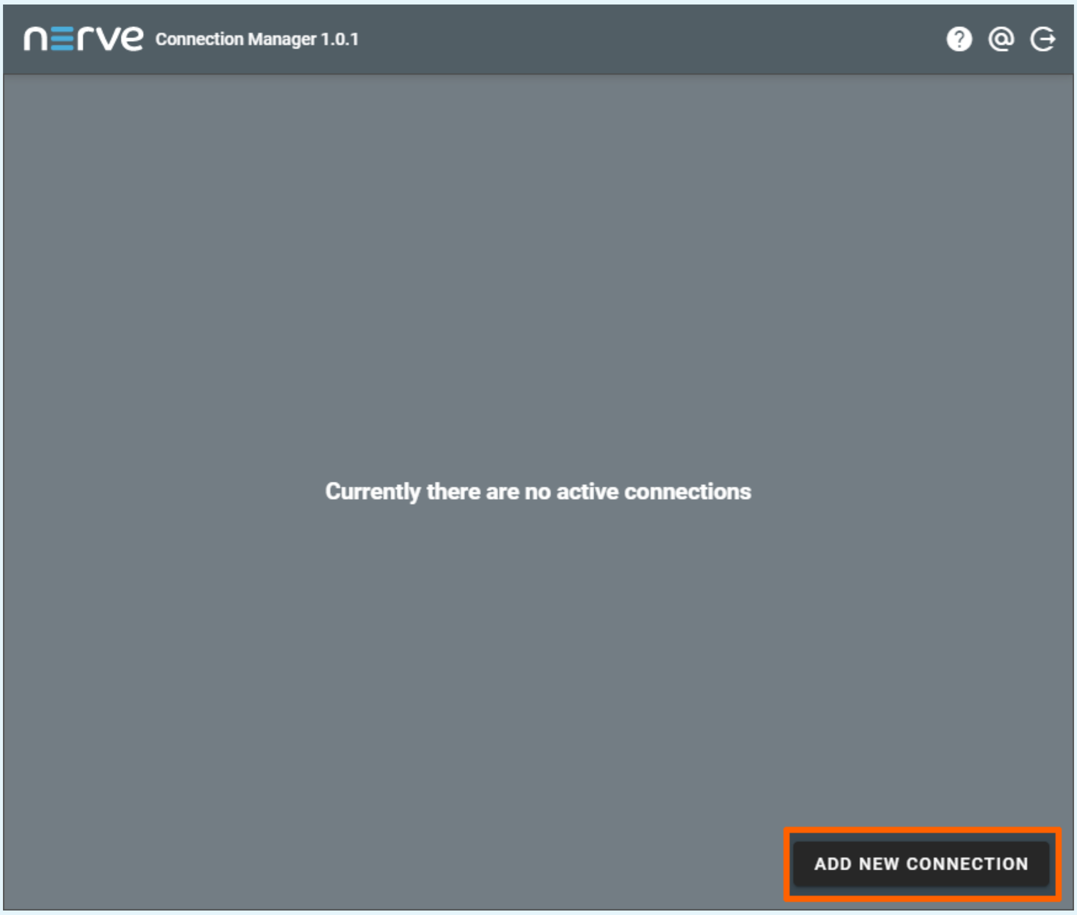
NOTE

If the Nerve Connection Manager does not start automatically, select the copy to clipboard symbol next to **Click me in order to run application** in the Management System. This copies the remote connection URL.


Connecting to Docs Tunnel

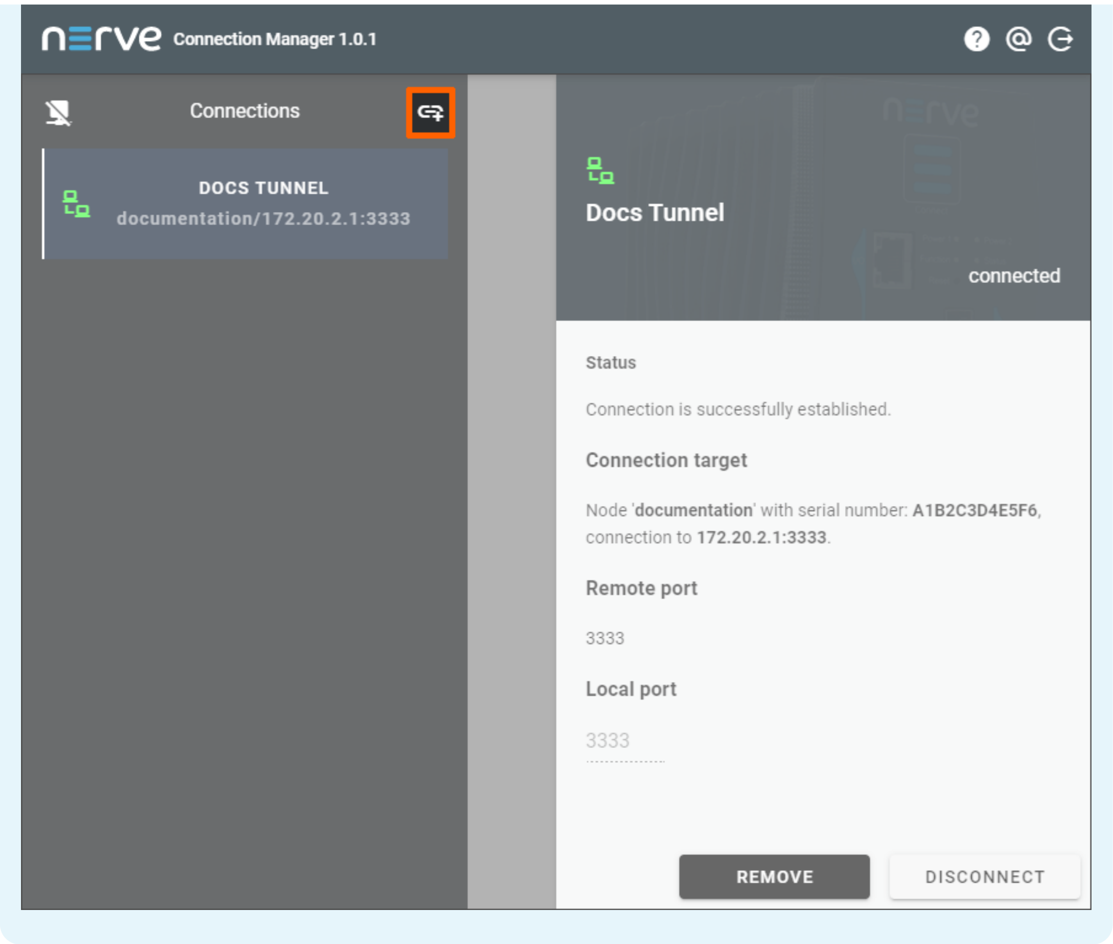
Click me in order to run application

Start the Nerve Connection Manager manually and add the new connection by clicking **ADD NEW CONNECTION** in the lower right and pasting the URL.



Currently there are no active connections

If an established connection already exists in the Nerve Connection Manager, select the **Add new connection** symbol next to **Connections** on the left side of the window.



Status

Connection is successfully established.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

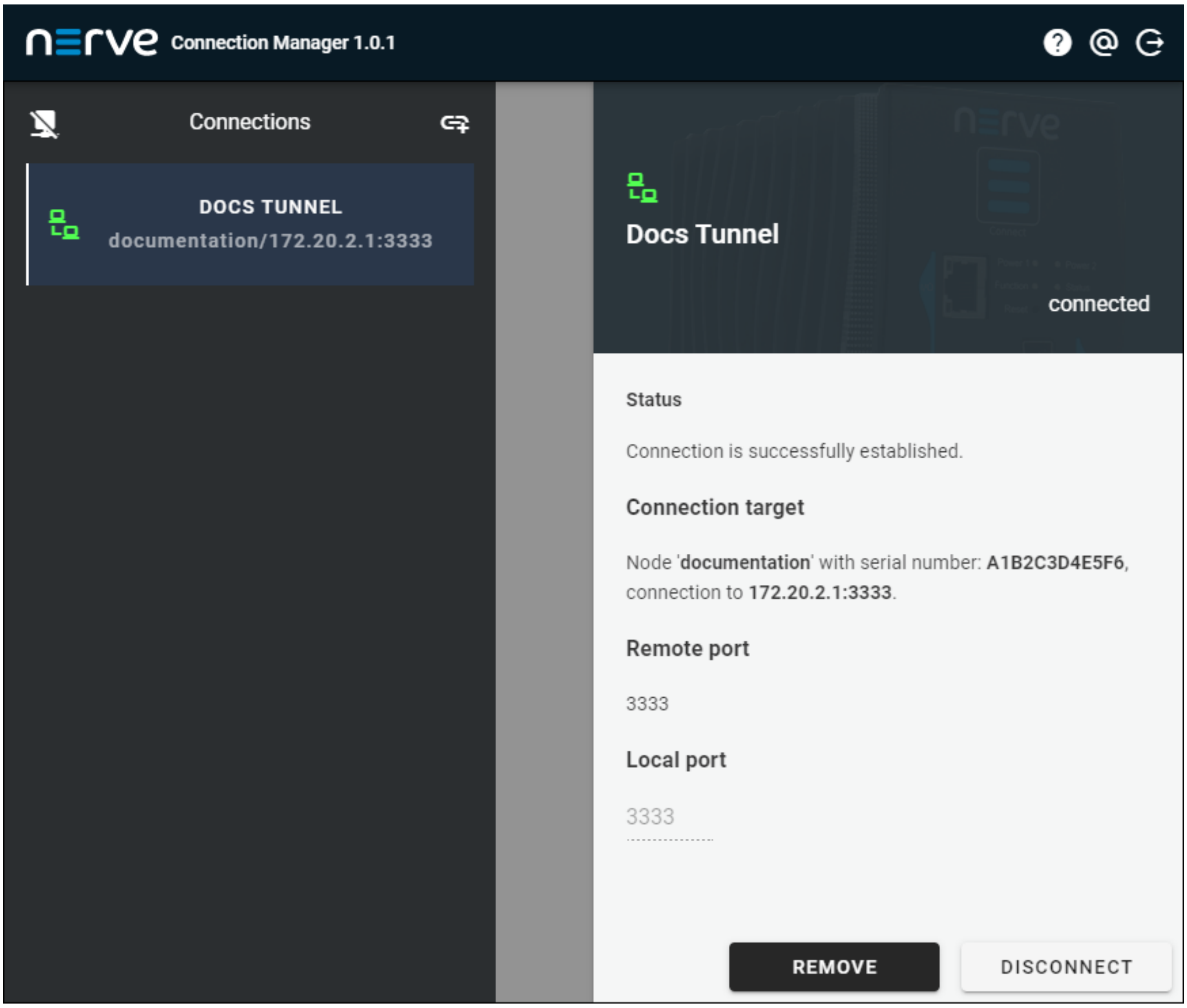
Remote port

3333

Local port

3333

The remote connection will be established once the Nerve Connection Manager starts.



Status

Connection is successfully established.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

Remote port

3333

Local port

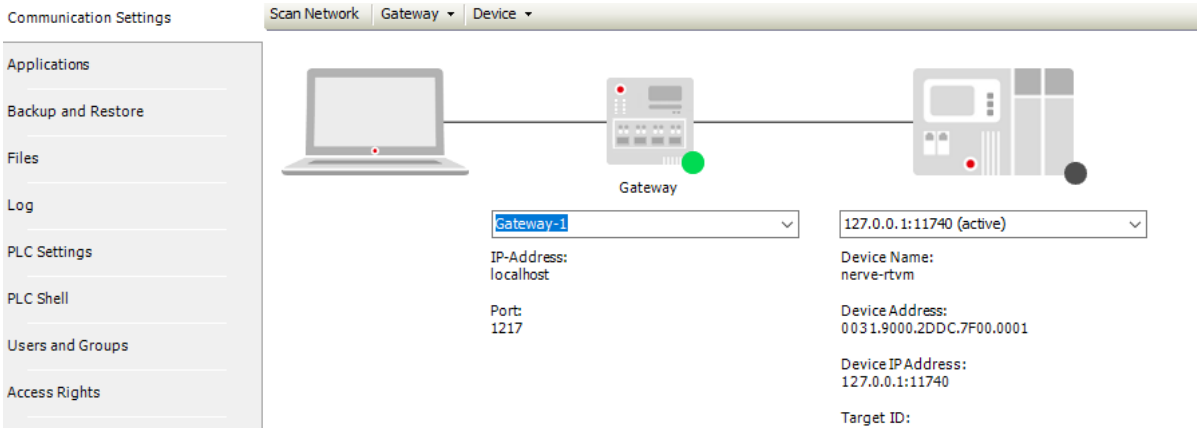
3333

Data about the establish remote tunnel is displayed on the right half of the Nerve Connection Manager window, showing the **Status**, **Connection target**, **Remote port** and **Local port** with a summary on the left side under the remote tunnel name.

NOTE

If the local port on the local workstation is already in use or occupied by the system, the Nerve Connection Manager will not establish a connection. **Local port** will be marked on the right. Enter a different port in this field that is not used on the workstation in order to establish the remote tunnel.

The connection can now be used from the local workstation by using `localhost:<localport>` through PuTTY in order to establish an SSH connection or in a web browser. The screenshot below shows how to connect to a node through a remote tunnel using the CODESYS Development System. In an open project, double-click **Device (Nerve_MFN_100)** in the tree view on the left. Go to **Communication Settings** in the middle of the window and enter `127.0.0.1:<portonpc>` in the text box under the device on the right. Replace `<portonpc>` with the port number that was defined under **Port on PC** in the Management System.



Gateway-1

IP-Address:
localhost

Port:
1217

127.0.0.1:11740 (active)

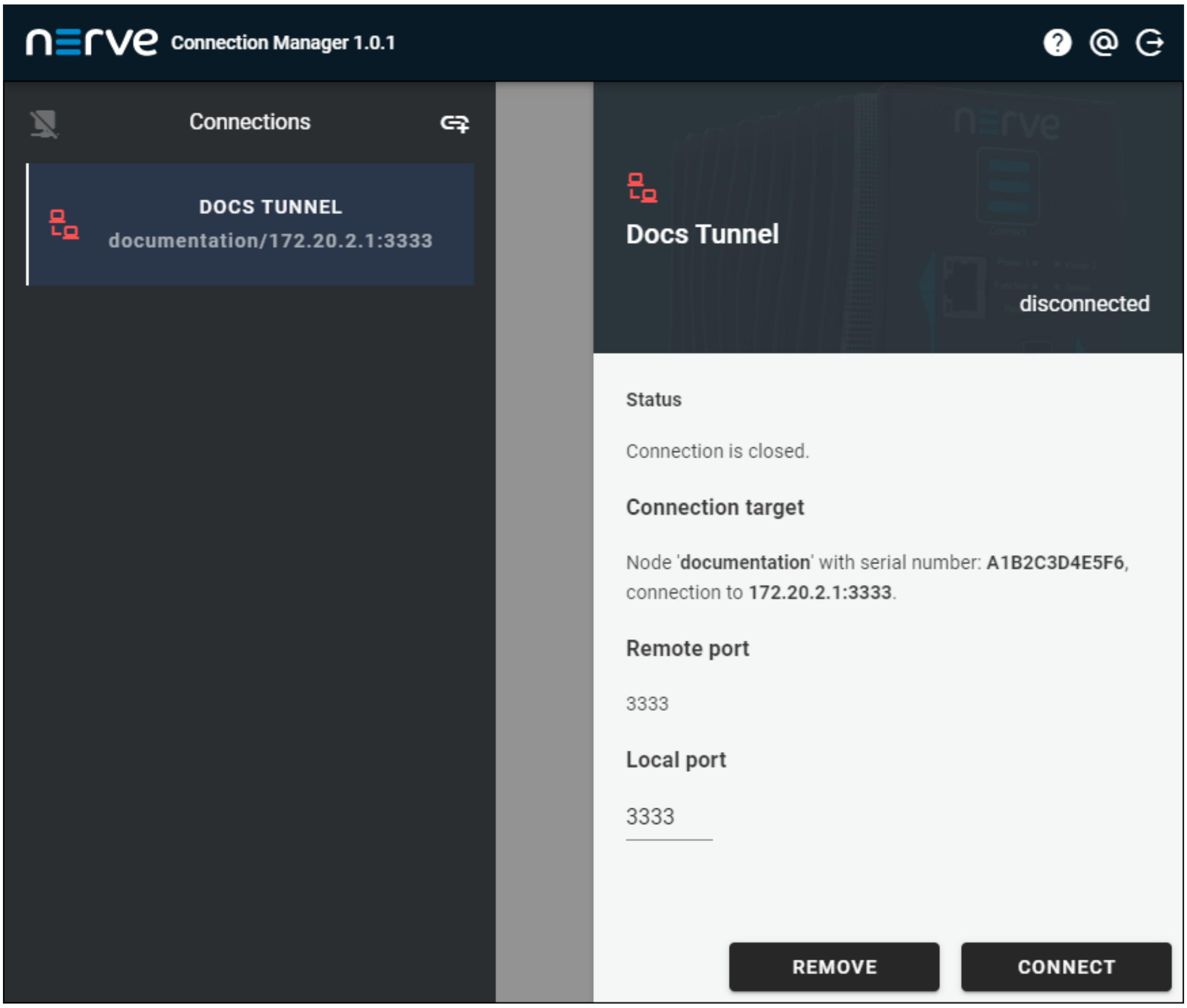
Device Name:
nerve-rtvm

Device Address:
0031.9000.2DDC.7F00.0001

Device IP Address:
127.0.0.1:11740

Target ID:

Disconnect from the remote tunnel by clicking **DISCONNECT** in the lower right corner of the Nerve Connection Manager.



DOCS TUNNEL
documentation/172.20.2.1:3333

Docs Tunnel

disconnected

Status

Connection is closed.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

Remote port

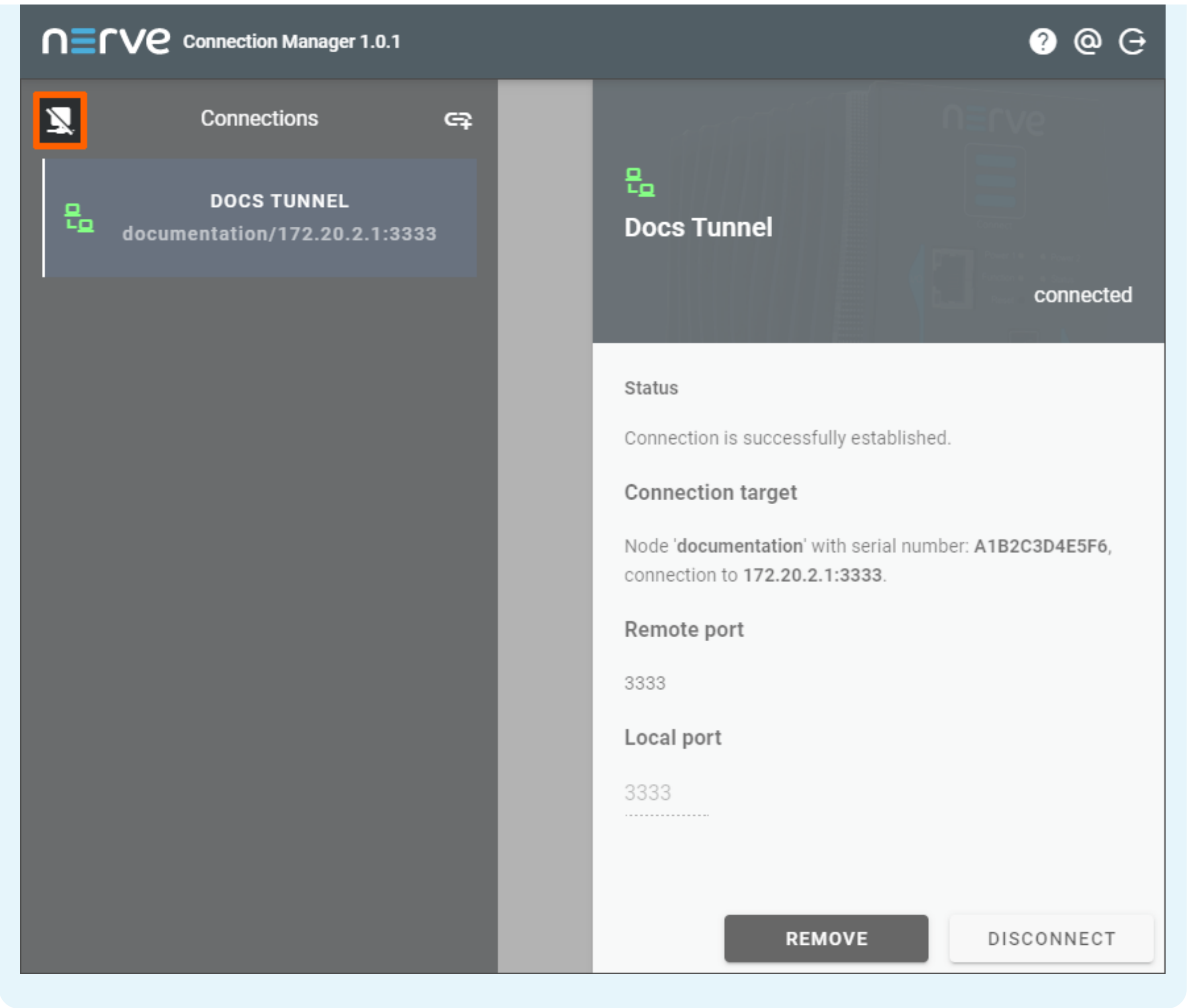
3333

Local port

3333

NOTE

Alternatively, all remote connections can be disconnected at once by clicking the **Disconnect all** symbol on the left side next to **Connections**.



Status

Connection is successfully established.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

Remote port

3333

Local port

3333

Note that disconnecting does not terminate the connection. The connection will stay established until it is terminated in the Nerve Connection Manager, the Local UI or the Management System.

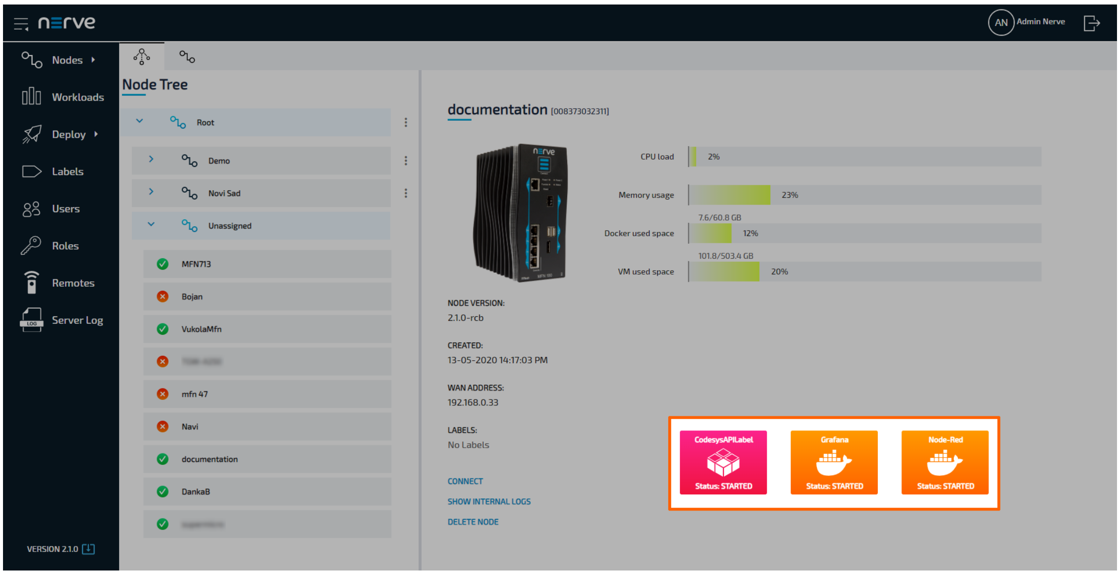
Using a remote tunnel to a workload

Note that the Nerve Connection Manager is required to use a remote tunnel. Download the Nerve Connection Manager from the Nerve Software Center and install it first.

Established remote tunnels are listed under **Remotes** in the navigation on the left until they are terminated.

1. Select **Nodes** in the navigation on the left.
2. Select the node tree tab

   on the right to display registered nodes in the node tree.
3. Select the node that has a deployed workload with a remote connection.
4. Select the workload.



documentation [008373032311]

CPU load 2%

Memory usage 23%

7.6/60.8 GB

Docker used space 12%

101.8/503.4 GB

VM used space 20%

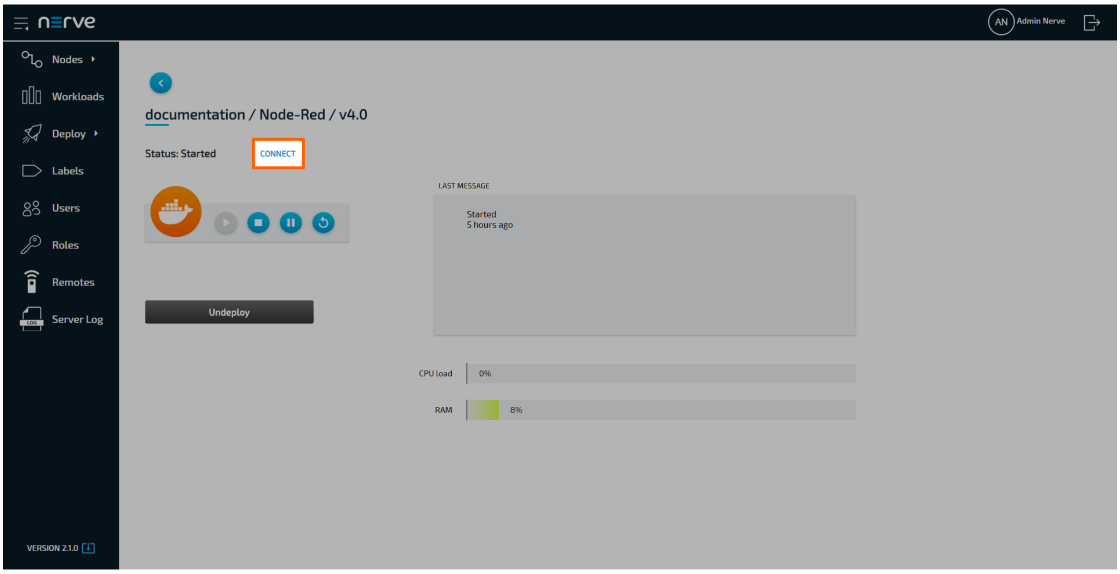
NODE VERSION:
2.1.0-rcb

CREATED:
13-05-2020 14:17:03 PM

WAN ADDRESS:
192.168.0.33

LABELS:
No Labels

5. Click **CONNECT** next to the workload status.



documentation / Node-Red / v4.0

Status: Started

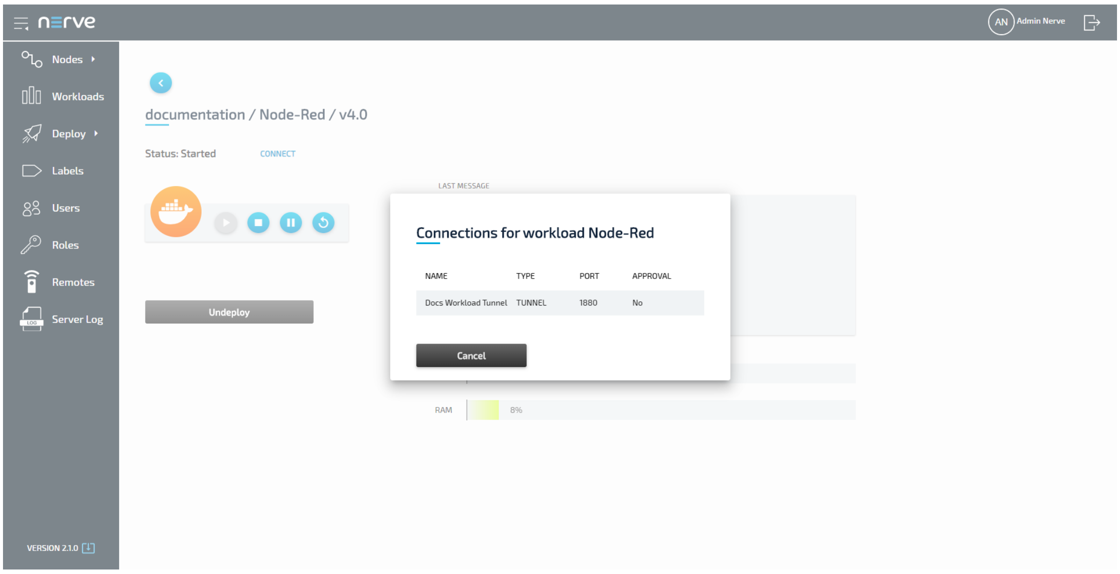
LAST MESSAGE

Started
5 hours ago

CPU load 0%

RAM 8%

6. Select the remote connection from the list in the new window. Note that remote tunnels have the type **TUNNEL**.



documentation / Node-Red / v4.0

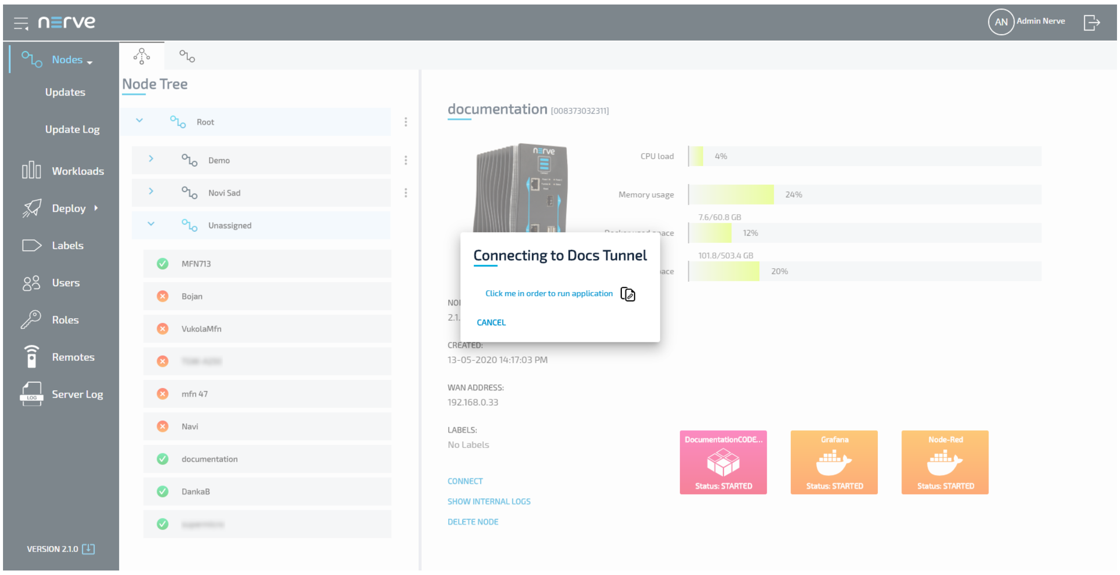
Status: Started

Connections for workload Node-Red

NAME	TYPE	PORT	APPROVAL
Docs Workload Tunnel	TUNNEL	1880	No

Cancel

7. Select **Click me in order to run application** in the new window.



documentation [008373032311]

Connecting to Docs Tunnel

Click me in order to run application

CANCEL

CREATED:
13-05-2020 14:17:03 PM

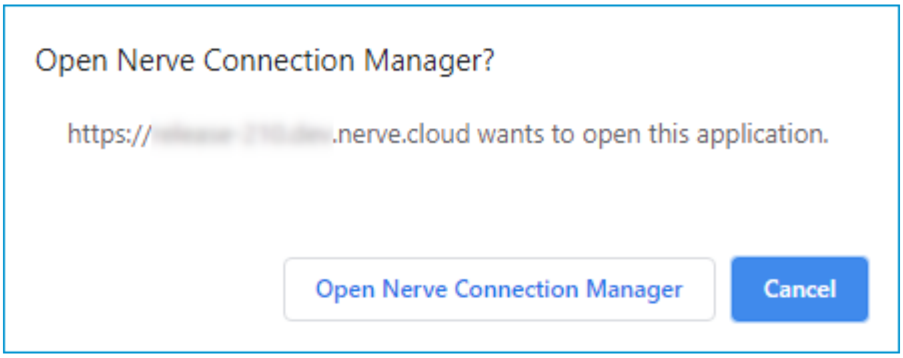
WAN ADDRESS:
192.168.0.33

LABELS:
No Labels

NOTE

If **Local acknowledgment** is set to **Yes**, the Management System will wait for approval until the remote connection has been locally approved before displaying the window above. Refer to Approving a remote connection for more information.

8. If the Nerve Connection Manager installed correctly, confirm the browser message that the Nerve Connection Manager shall be opened.
   Depending on the browser that is used, this message will differ. The Nerve Connection Manager will start automatically once the message is confirmed.

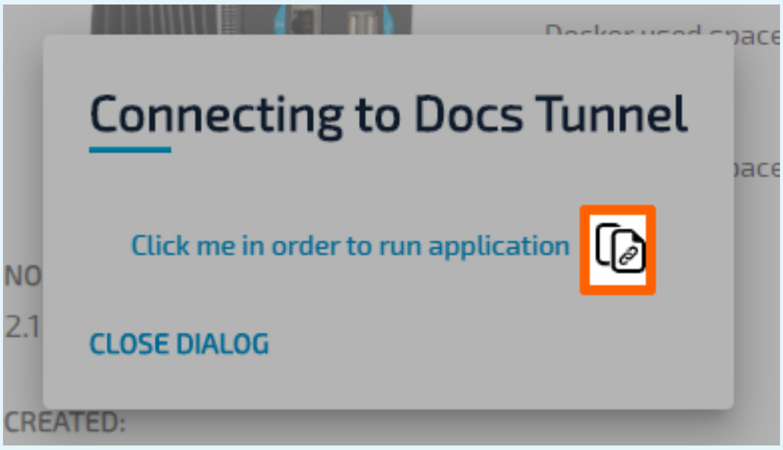


Open Nerve Connection Manager?

https://[redacted].nerve.cloud wants to open this application.

NOTE

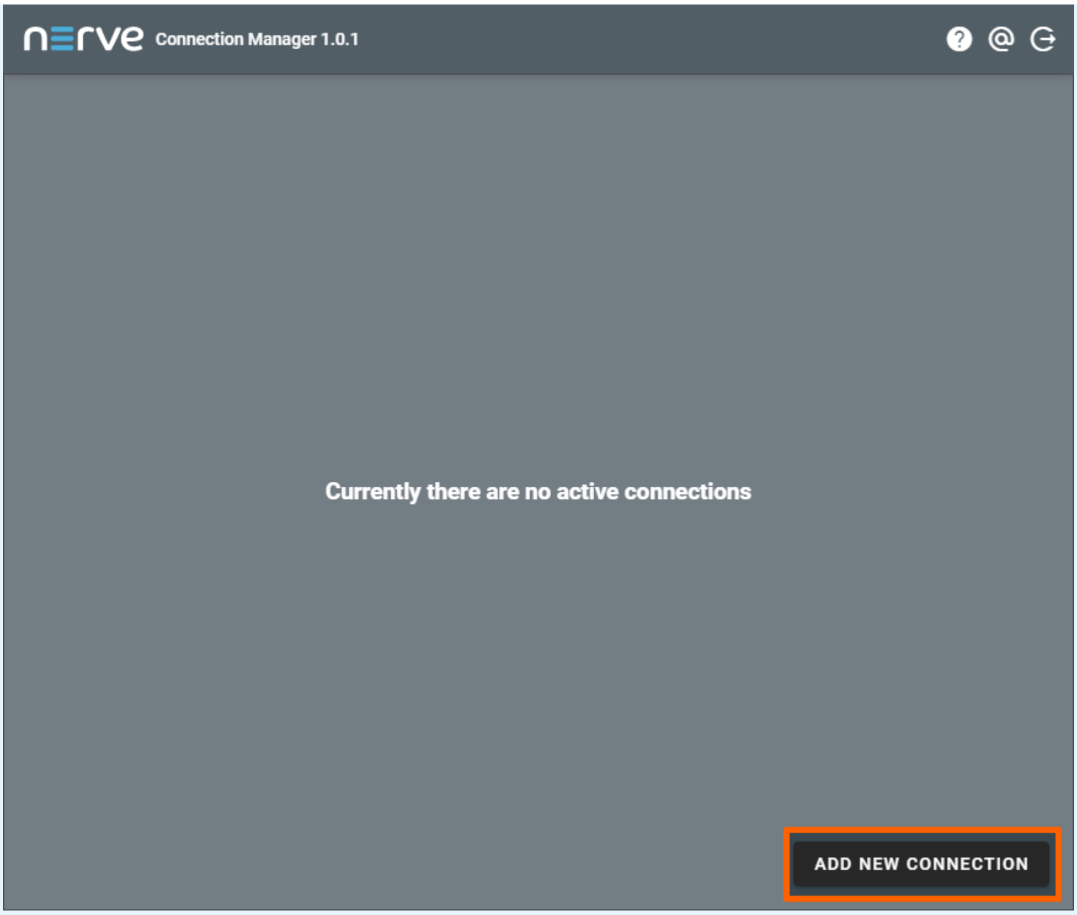
If the Nerve Connection Manager does not start automatically, select the copy to clipboard symbol next to **Click me in order to run application** in the Management System. This copies the remote connection URL.



Connecting to Docs Tunnel

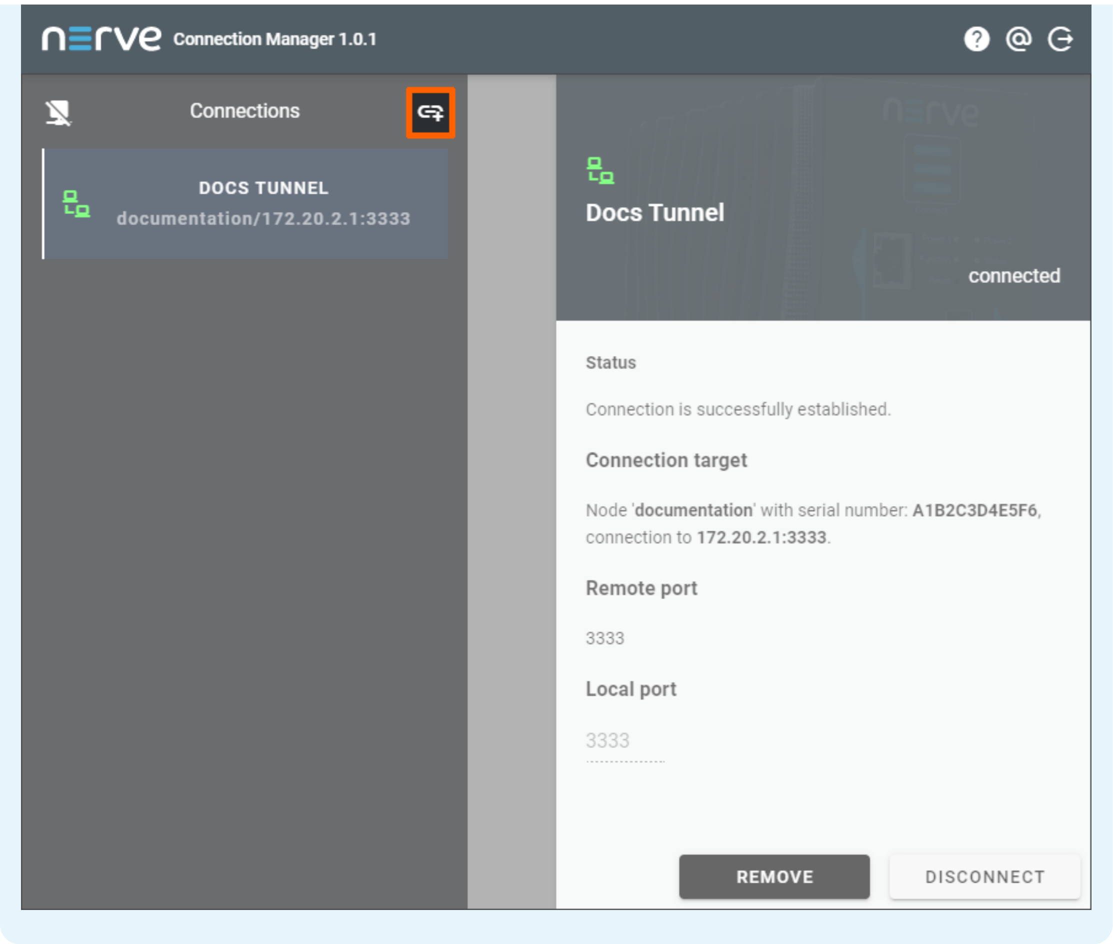
Click me in order to run application

Start the Nerve Connection Manager manually and add the new connection by clicking **ADD NEW CONNECTION** in the lower right and pasting the URL.



nerve Connection Manager 1.0.1
Currently there are no active connections
ADD NEW CONNECTION

If an established connection already exists in the Nerve Connection Manager, select the **Add new connection** symbol next to **Connections** on the left side of the window.



Status

Connection is successfully established.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

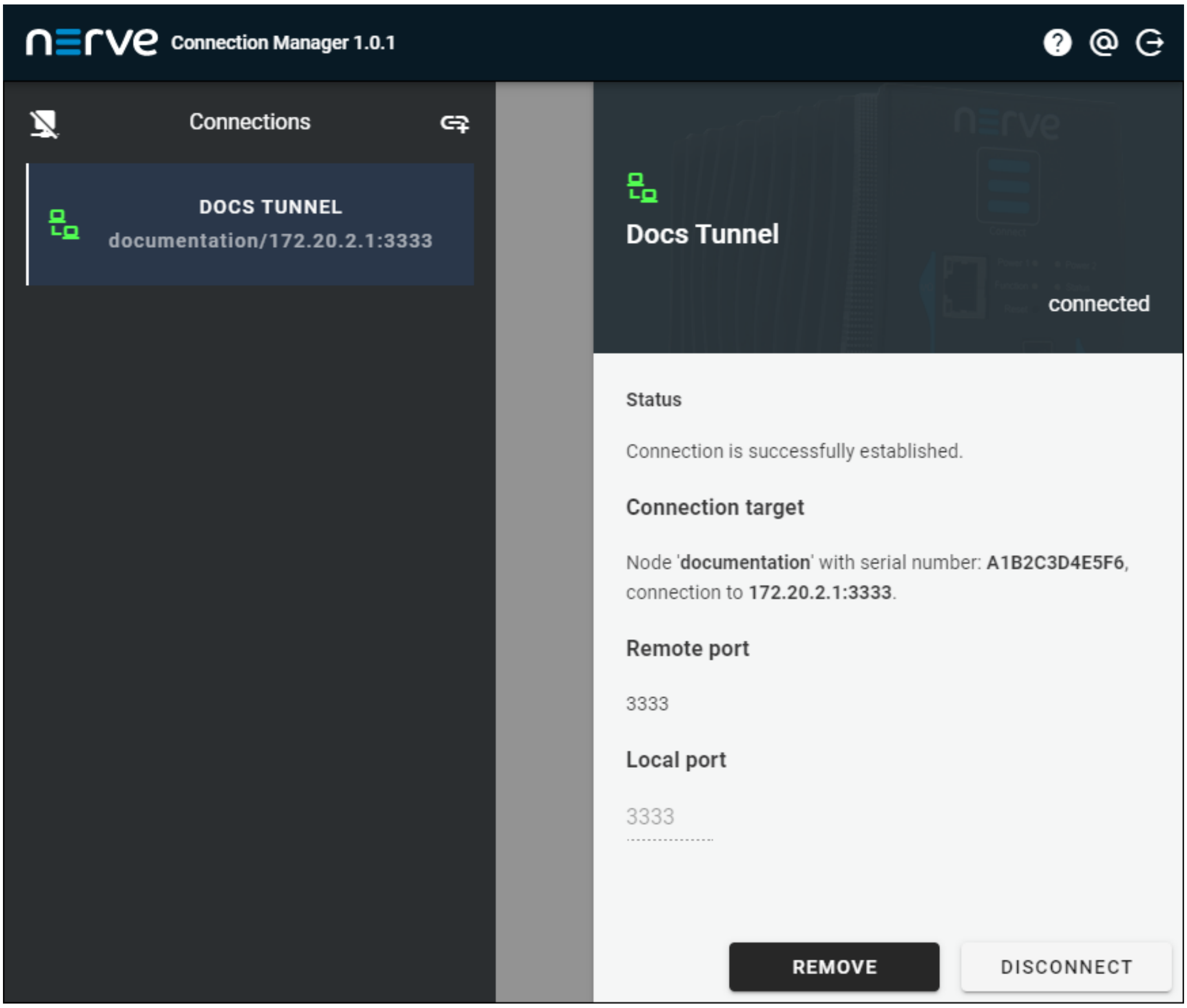
Remote port

3333

Local port

3333

The remote connection will be established once the Nerve Connection Manager starts.



Status

Connection is successfully established.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

Remote port

3333

Local port

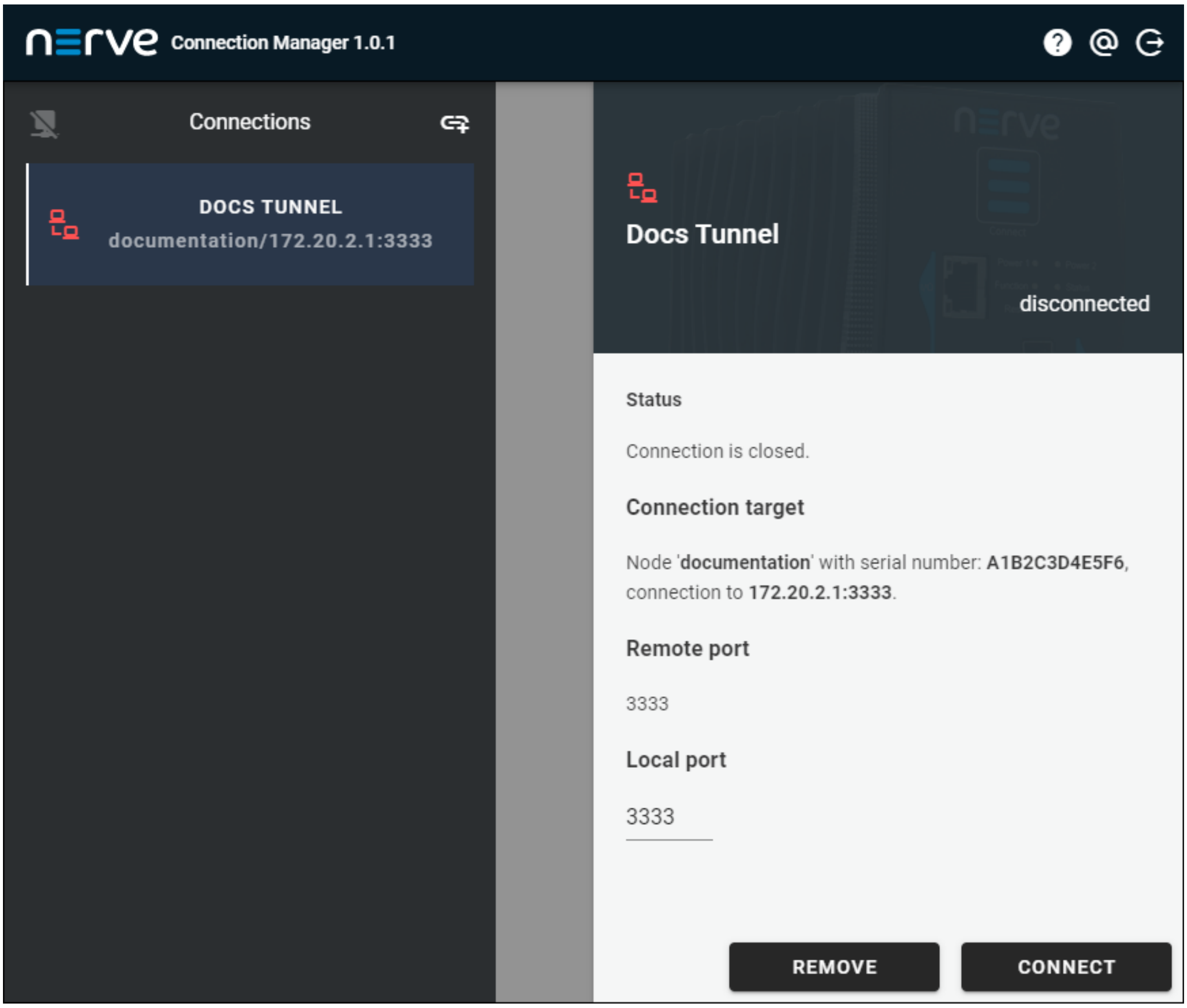
3333

Data about the establish remote tunnel is displayed on the right half of the Nerve Connection Manager window, showing the **Status**, **Connection target**, **Remote port** and **Local port** with a summary on the left side under the remote tunnel name.

NOTE

If the local port on the local workstation is already in use or occupied by the system, the Nerve Connection Manager will not establish a connection. **Local port** will be marked on the right. Enter a different port in this field that is not used on the workstation in order to establish the remote tunnel.

The connection can now be used from the local workstation by using `localhost:<localport>` through PuTTY in order to establish an SSH connection or in a web browser. Disconnect by clicking **DISCONNECT** in the lower right corner.



Status

Connection is closed.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

Remote port

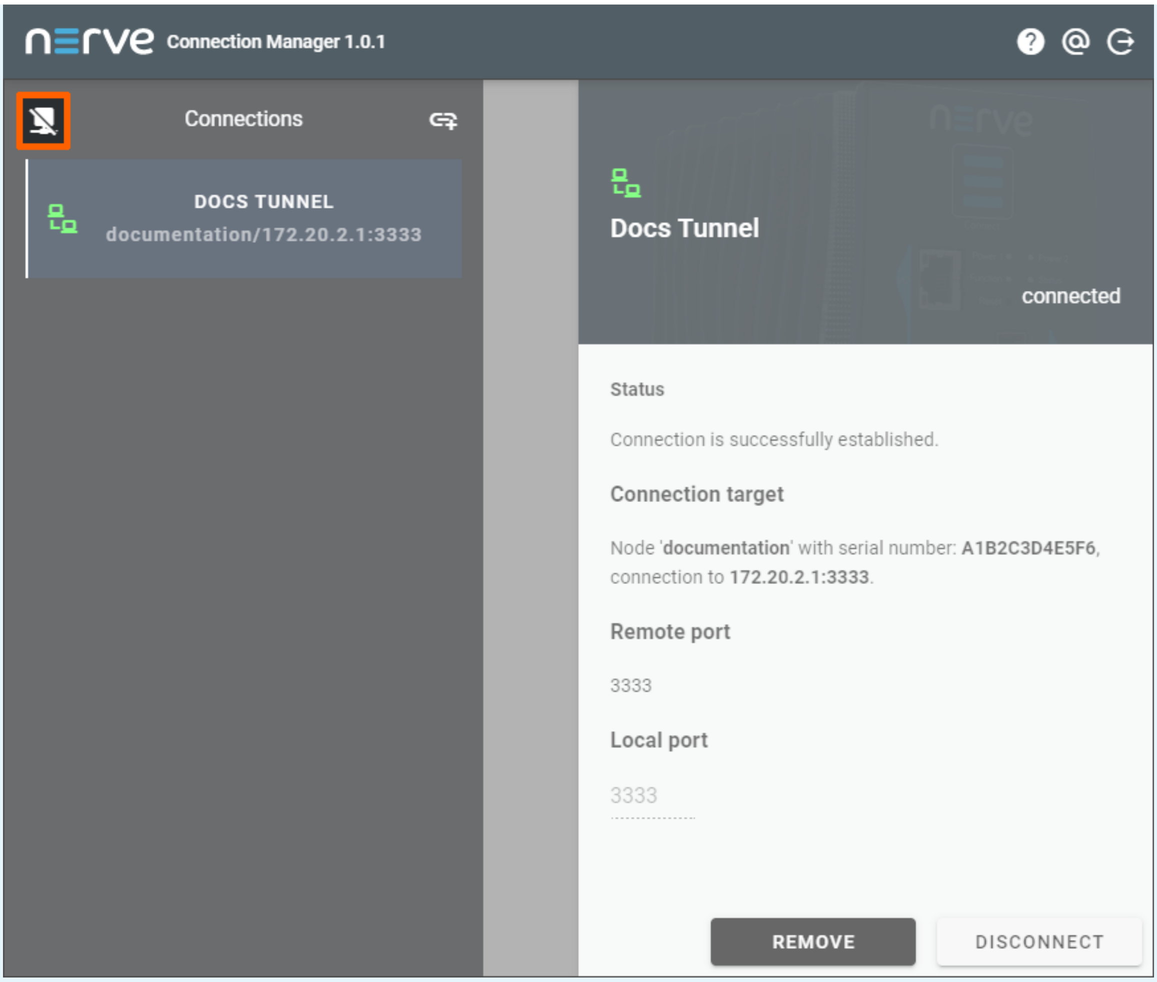
3333

Local port

3333

NOTE

Alternatively, all remote connections can be disconnected at once by clicking the **Disconnect all** symbol on the left side next to **Connections**.



Status

Connection is successfully established.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

Remote port

3333

Local port

3333

Note that disconnecting does not terminate the connection. The connection will stay established until it is terminated in the Nerve Connection Manager, the Local UI or the Management System.

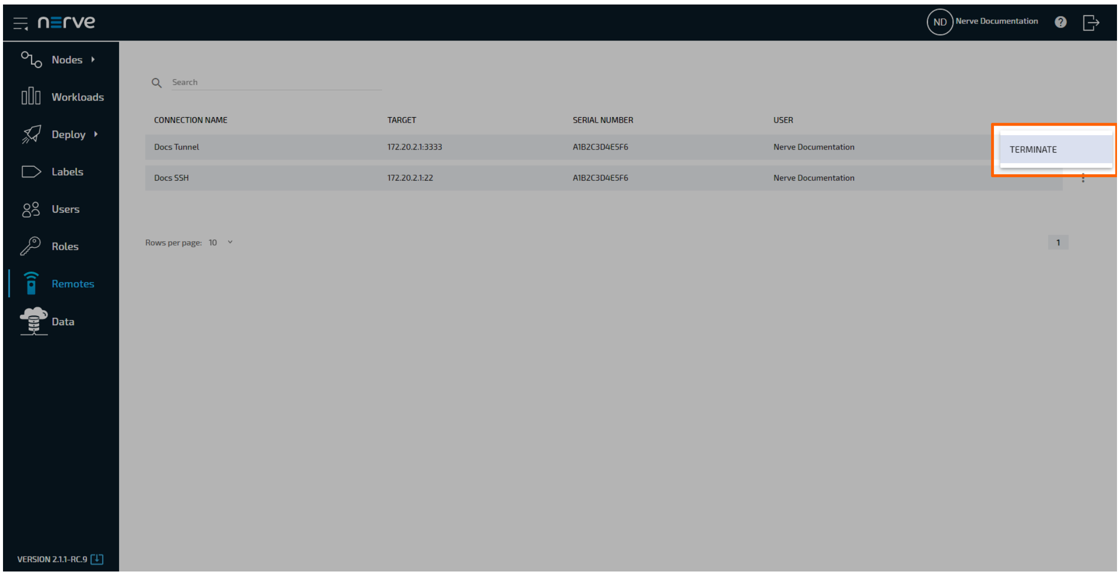
Terminating remote connections

Remote connections are open and can be used as long as they are not terminated. A remote connection can be terminated from the Management System, in the Local UI or in the Nerve Connection Manager. Also, remote connections terminate automatically after 30 minutes of inactivity. Once a connection has been terminated, it has to be established again.

Terminating an active remote connection in the Management System

Note that terminating an open remote connection does not remove the configuration of the remote connection from the node or workload. If a remote connection is terminated, it has to be re-established in the Management System to be used again.

1. Connect to the Management System.
2. Select **Remotes** in the navigation on the left.
3. Select the ellipsis menu to the right of an active remote connection.
4. Select **TERMINATE** in the overlay that appeared.



CONNECTION NAME	TARGET	SERIAL NUMBER	USER
Docs Tunnel	172.20.2.1:3333	A1B2C3D4E5F6	Nerve Documentation
Docs SSH	172.20.2.1:22	A1B2C3D4E5F6	Nerve Documentation

5. Select **OK** in the new window.

Terminating a connection in the Management System automatically removes the connection in the Management System, Local UI and Nerve Connection Manager.

NOTE

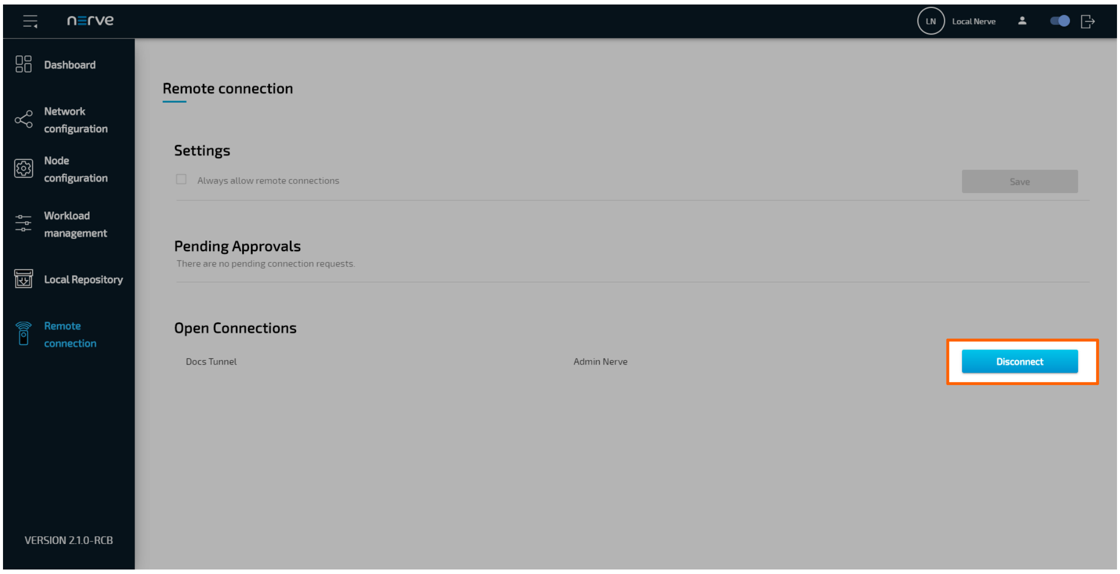
Once a remote screen has been terminated while the browser tab is still open, a pop-up window will appear that offers the option to reconnect. Clicking **Reconnect** in the pop-up window has no effect. Close the window and re-established the connection in the Management System.

Terminating an active remote connection in the Local UI

Note that terminating an open remote connection does not remove the configuration of the remote connection from the node or workload. If a remote connection is terminated, it has to be re-established in the Management System to be used again.

1. Connect to the Local UI.
2. Select **Remote Connections** in the navigation on the left.
3. Choose the remote connections that will be terminated.

4. Select **Disconnect**.



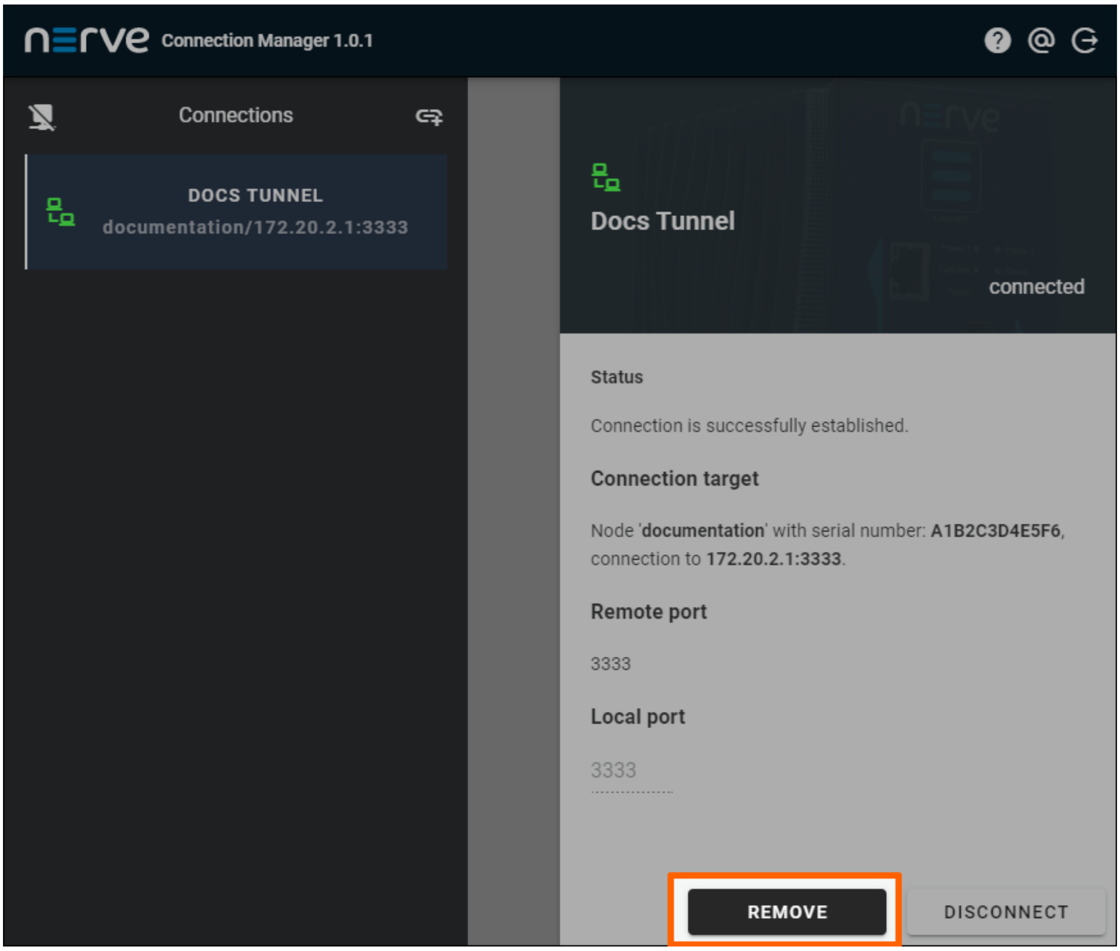
5. Select **YES** in the new window.

Terminating a connection in the Local UI automatically removes the connection in the Management System, Local UI and Nerve Connection Manager.

Terminating a remote connection in the Nerve Connection Manager

Note that terminating an open remote connection does not remove the configuration of the remote connection from the node or workload. If a remote connection is terminated, it has to be re-established in the Management System to be used again.

1. Open the Nerve Connection Manager. Note that the Nerve Connection Manager will already be open if a remote tunnel has been established.
2. Select a remote connection that will be terminated in the list on the left.
3. Select **REMOVE** in the lower-right.



Status

Connection is successfully established.

Connection target

Node '**documentation**' with serial number: **A1B2C3D4E5F6**, connection to **172.20.2.1:3333**.

Remote port

3333

Local port

3333

4. Select **YES** in the overlay that appeared.

Terminating a connection in the Nerve Connection Manager automatically removes the connection in the Management System, Local UI and Nerve Connection Manager. Exiting the Nerve Connection Manager terminates all remote tunnels

Common error cases and known issues

Below is a list of most common error cases and known limitations. Hints how to avoid them or solve them the easiest way are given where applicable.

Remote screens

- When trying to connect to a suspended workload, long loading times might occur. The connection can also seem established but the user will not be able to act in the remote screen window. The reason might be that a remote screen to a suspended workload was attempted. This is not supported.

  Close the browser tab and terminate the connection in the Management System in that case. Make sure the workload is in the started state and re-establish the remote screen. If the behavior persists, investigate the workload settings or the node.

- Remote screens to workloads will be shown as active under **Remotes** if the workload is undeployed while the remote screen is being used.

Remote tunnels

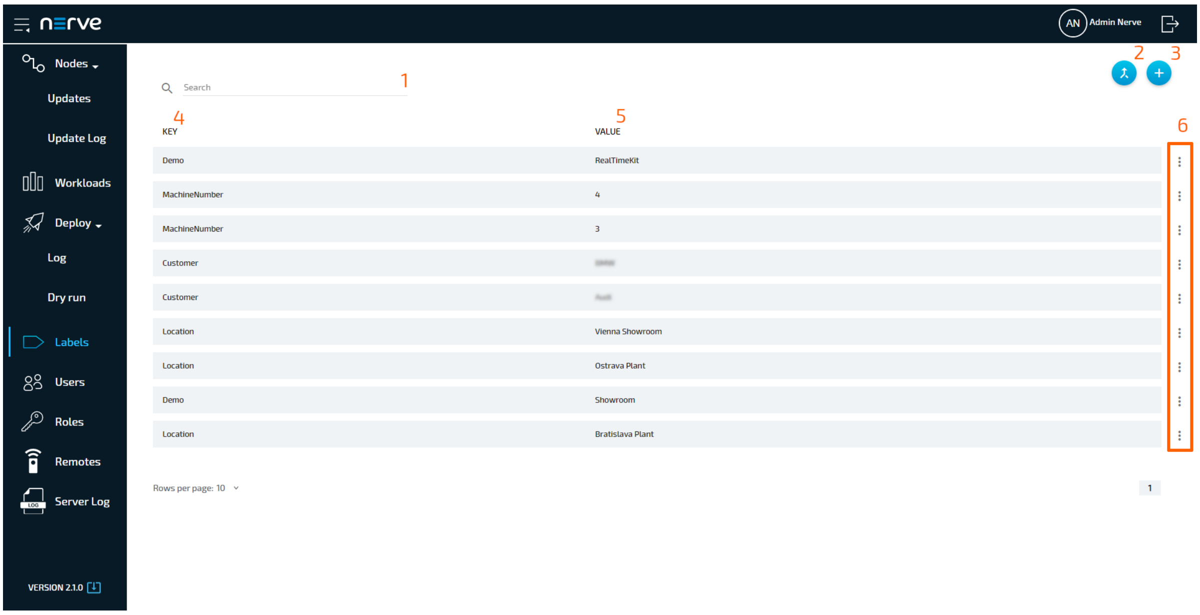
- Using two remote tunnels to two nodes, accessing the Local UI of each node at the same time is not possible. This is due to authentication conflicts.

Use the incognito mode of the current browser for the second tab or a second browser if both Local UIs have to be operated at the same time.

- Some systems might restrict usage of local ports lower than 1024. This is true for Linux systems especially. Enter ports higher than 1024 under **Port on PC** when configuring a remote tunnel to avoid port conflicts.

Labels

Labels are a useful feature that help with the organization of nodes and workloads. They can be defined and used freely. All labels that have been created are listed in the labels menu.



KEY	VALUE
Demo	RealTimeKit
MachineNumber	4
MachineNumber	3
Customer	[redacted]
Customer	[redacted]
Location	Vienna Showroom
Location	Ostrava Plant
Demo	Showroom
Location	Bratislava Plant

Rows per page: 10

1

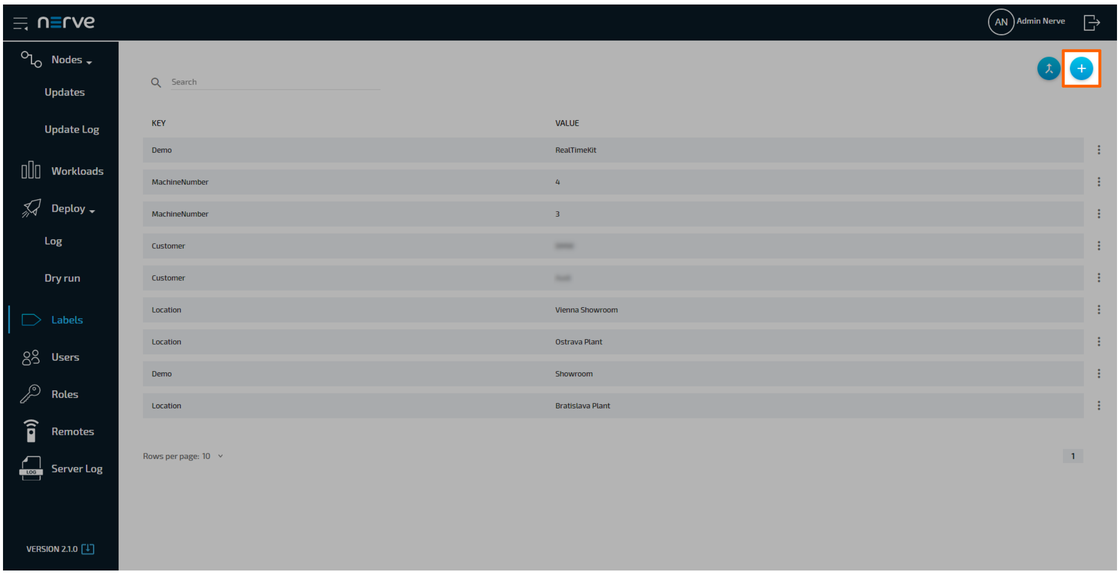
Item	Description
Search bar (1)	Use the search bar to filter labels by key.
Merge labels (2)	Clicking here leads to a page that allows to merge existing labels.
Add new label (3)	Click here to add a new label consisting of label key and label value.
KEY (4)	This is the key of the label. It can be understood as the "category" of the label. Examples of label keys are location, machine number or hardware.
VALUE (5)	This is the value of the label. It corresponds to the key of the label. Examples are Vienna, Machine 1 or MFN 100.
Ellipsis menu (6)	Clicking here opens an overlay that allows the deletion of labels.

Adding a new label

Labels can be defined with any key and value. Note that labels are displayed in the `<key>:<value>` format in the Management System. The key of a label can be understood as the category of a label with the value being an item in that category. Example: `Location:Vienna`.

1. Select **Labels** in the menu on the left side.

2. Click the **Add new label** icon in the upper-right corner.



KEY	VALUE
Demo	RealTimeKit
MachineNumber	4
MachineNumber	3
Customer	[redacted]
Customer	[redacted]
Location	Vienna Showroom
Location	Ostrava Plant
Demo	Showroom
Location	Bratislava Plant

Rows per page: 10

1

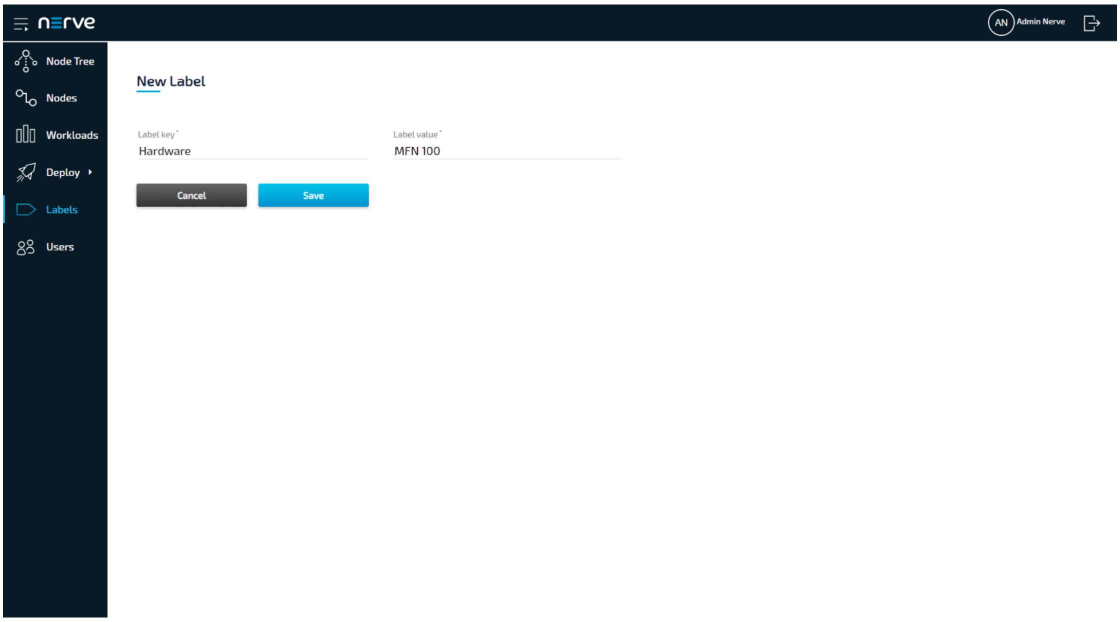
3. Enter the following information:

Item	Description
Label key	Enter the "category" of the label. Example: **Hardware**.
Label value	Enter the value for the label here. Example: **MFN 100**.

NOTE

Label keys must consist of one word only. Use – and _ as separators.
Also, only use alphanumeric characters (a–z, A–Z, 0–9) and underscore (_). Any other special characters are not allowed.

4. Click **Save** to add the new label.



New Label

Label key * Hardware

Label value * MFN 100

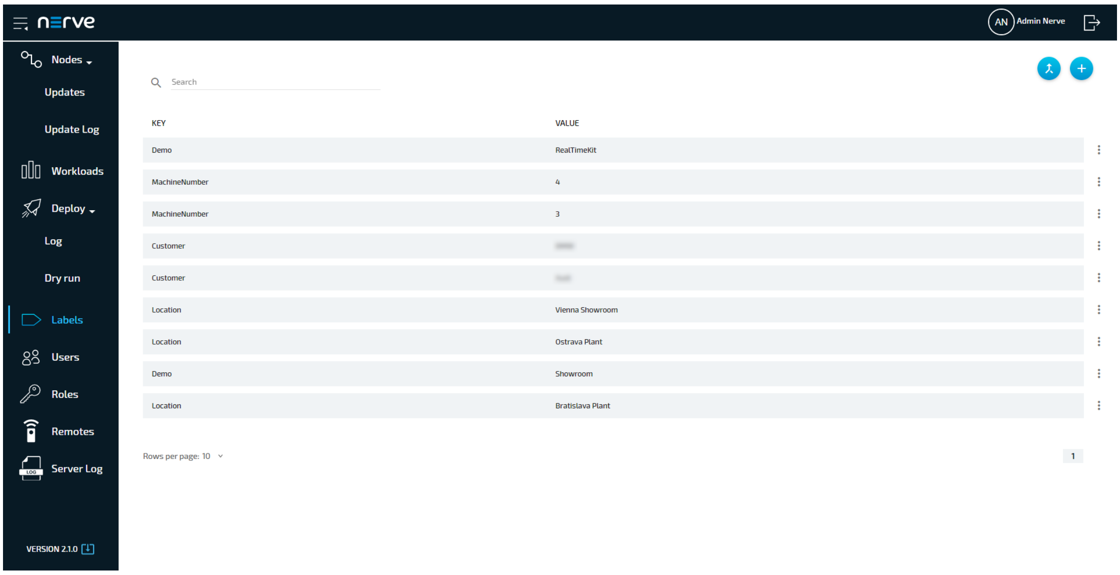
The label now appears in the label list and can be used in the Management System. Labels can be assigned to nodes when node details are edited. In the workload provisioning process they can be chosen as selectors.

Editing a label

Editing a label is virtually identical to the process of creating a new label. Note that labels will be edited even if they are currently assigned to nodes or used as selectors by workloads.

1. Select **Labels** in the menu on the left side.

2. Select a label to edit.

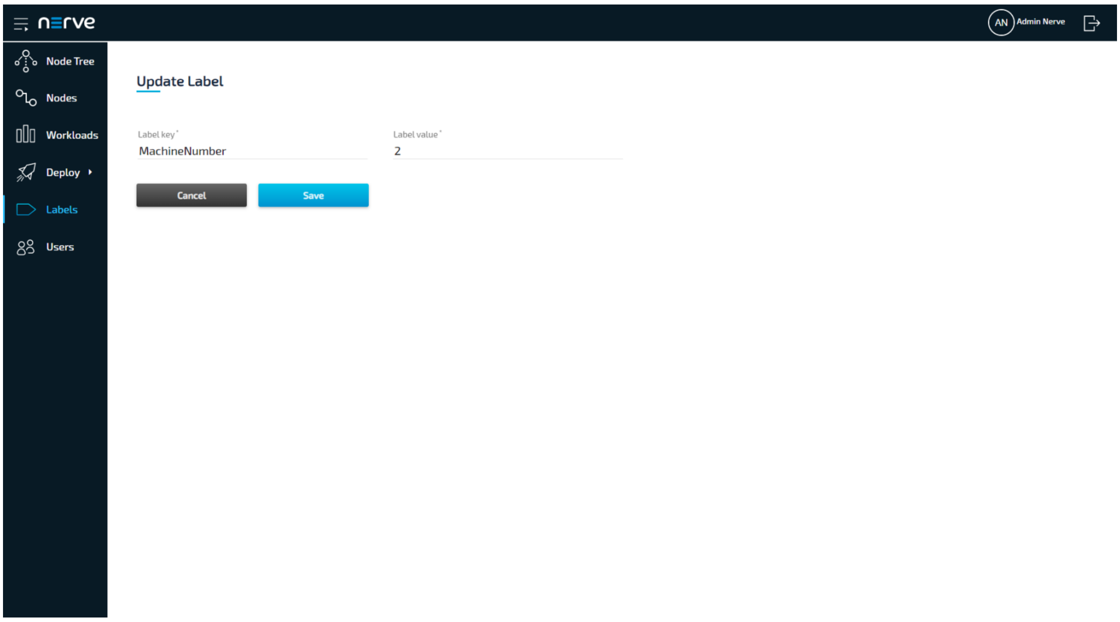


KEY	VALUE
Demo	RealTimeKit
MachineNumber	4
MachineNumber	3
Customer	[redacted]
Customer	[redacted]
Location	Vienna Showroom
Location	Ostrava Plant
Demo	Showroom
Location	Bratislava Plant

Rows per page: 10

1

3. Edit **Label key** and **Label value**.



Update Label

Label key* MachineNumber

Label value* 2

NOTE

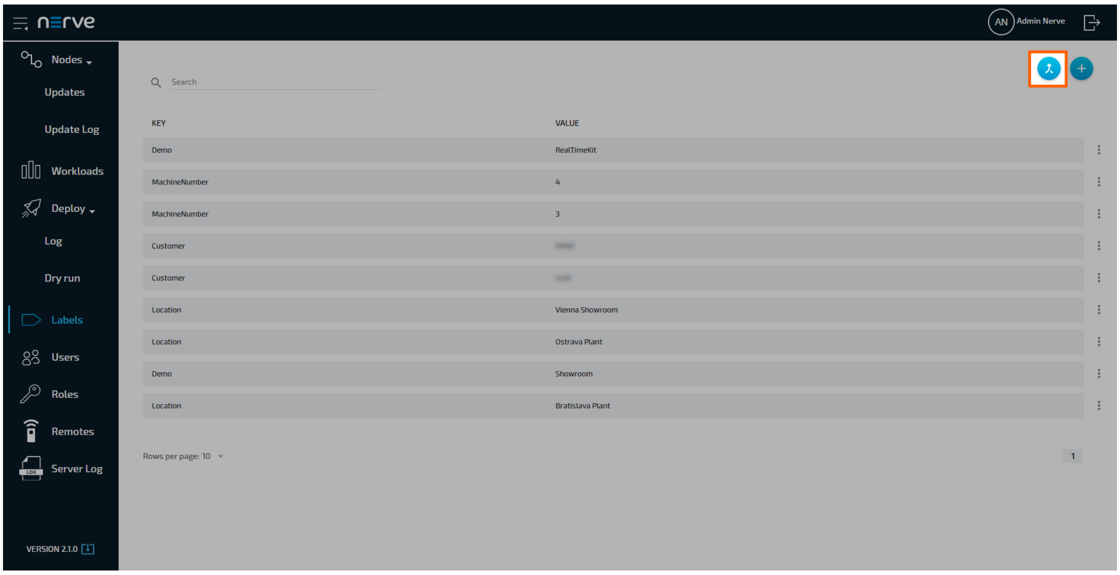
Label keys must consist of one word only. Use - and _ as separators.
Also, only use alphanumeric characters (`a-z`, `A-Z`, `0-9`) and underscore (_). Any other special characters are not allowed.

4. Click **Save** to update the label.

Merging labels

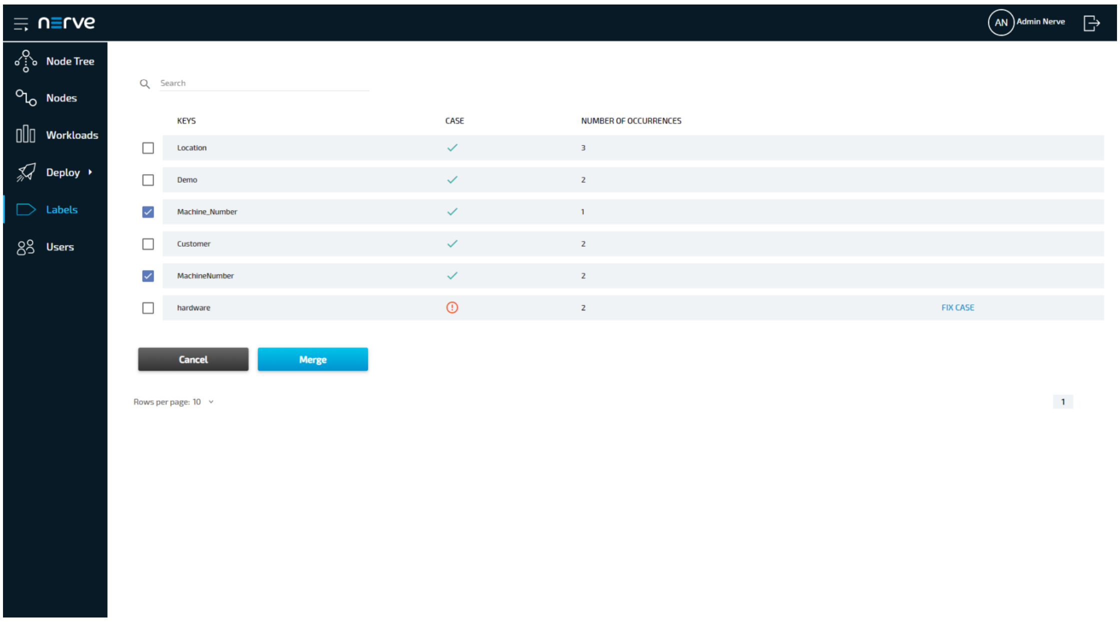
Labels with overlapping information or typos can be merged into one. However, only label keys are merged. The label values are left untouched and assigned to the new label key.

1. Select **Labels** in the menu on the left side.
2. Click the **Merge labels** icon in the upper-right corner.



KEY	VALUE
Demo	RealTimeKit
MachineNumber	4
MachineNumber	3
Customer	[redacted]
Customer	[redacted]
Location	Vienna Showroom
Location	Ostrava Plant
Demo	Showroom
Location	Bratislava Plant

3. Tick the checkboxes left of the labels that will be merged. Multiple labels can be selected.
4. Select **Merge**.



	KEYS	CASE	NUMBER OF OCCURRENCES	
□	Location	✓	3	
□	Demo	✓	2	
☑	Machine_Number	✓	1	
□	Customer	✓	2	
☑	MachineNumber	✓	2	
□	hardware	!	2	FIX CASE

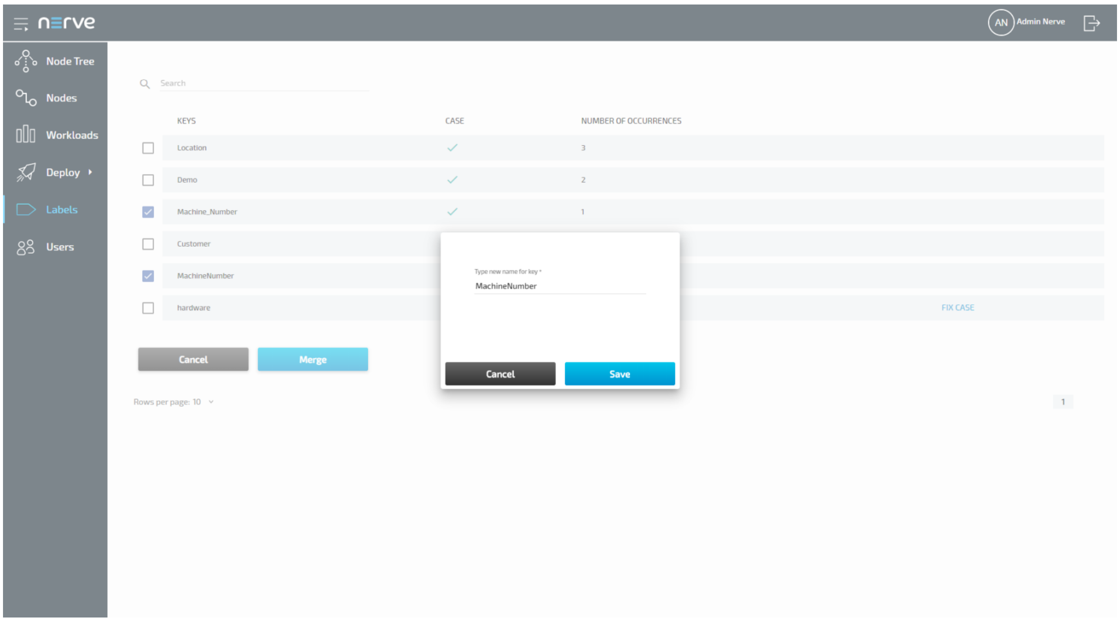
NOTE

The table here gives the following information:

- **CASE**
  The system checks if labels that use the same characters are also written in the same case. A green check mark indicates that the case matches. A circled red exclamation mark appears if the case does not match.

- **NUMBER OF OCCURRENCES**
  The number here indicates how many labels have been defined with the same label key.

5. Type in the new name for the label key.



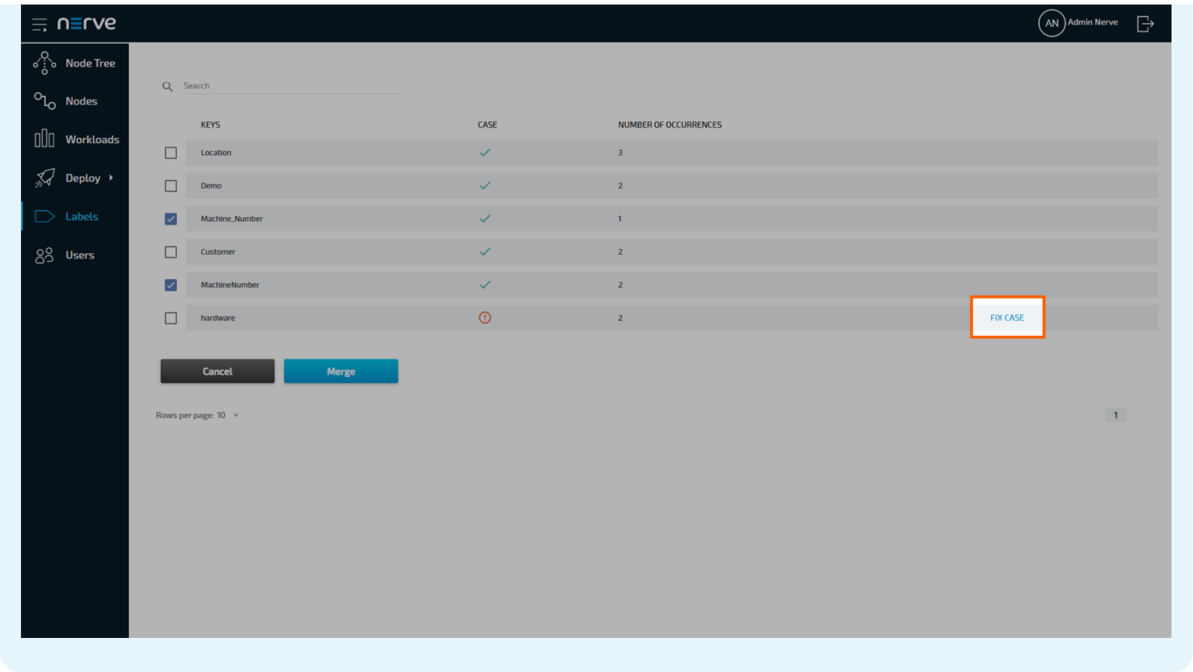
	KEYS	CASE	NUMBER OF OCCURRENCES	
□	Location	✓	3	
□	Demo	✓	2	
☑	Machine_Number	✓	1	
□	Customer			
☑	MachineNumber			
□	hardware			FIX CASE

Type new name for key *
MachineNumber

6. Click **Save** to save the new label.

NOTE

If labels have the same label key but are not written in the same case, the Management System will recognize it. In this case, select **FIX CASE** in the list, type in the new name for the label key and select **Save**.



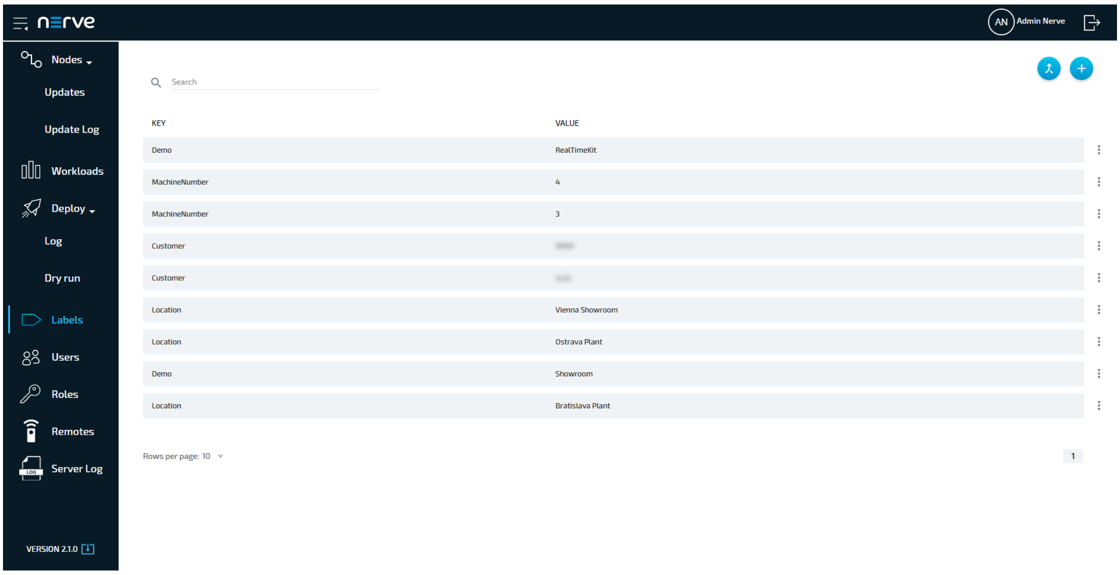
	KEYS	CASE	NUMBER OF OCCURRENCES
□	Location	✓	3
□	Demo	✓	2
☑	Machine_Number	✓	1
□	Customer	✓	2
☑	MachineNumber	✓	2
□	hardware	!	2

Deleting a label

Note that when a label is deleted from the list, it will also be deleted from any node or workload that had the label assigned.

1. Select **Labels** in the menu on the left side.

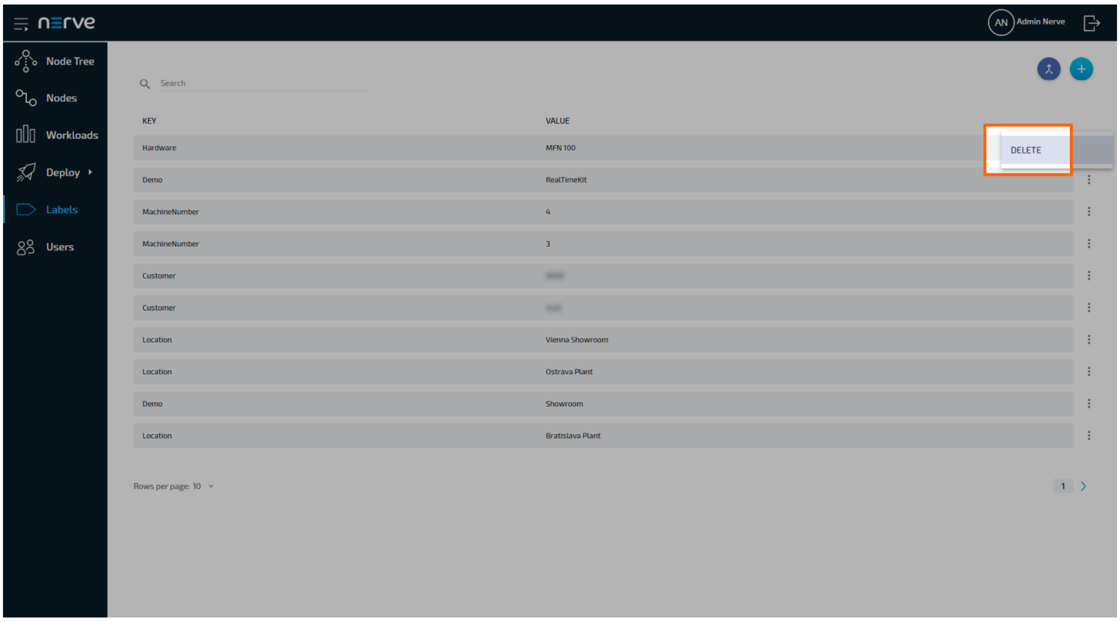
2. Choose a label to delete.



KEY	VALUE
Demo	RealTimeKit
MachineNumber	4
MachineNumber	3
Customer	[redacted]
Customer	[redacted]
Location	Vienna Showroom
Location	Ostrava Plant
Demo	Showroom
Location	Bratislava Plant

3. Click the ellipsis menu on the right side of the label.

4. Select **Delete** in the overlay that appeared.



KEY	VALUE
Hardware	MFN 100
Demo	RealTimeKit
MachineNumber	4
MachineNumber	3
Customer	[redacted]
Customer	[redacted]
Location	Vienna Showroom
Location	Ostrava Plant
Demo	Showroom
Location	Bratislava Plant

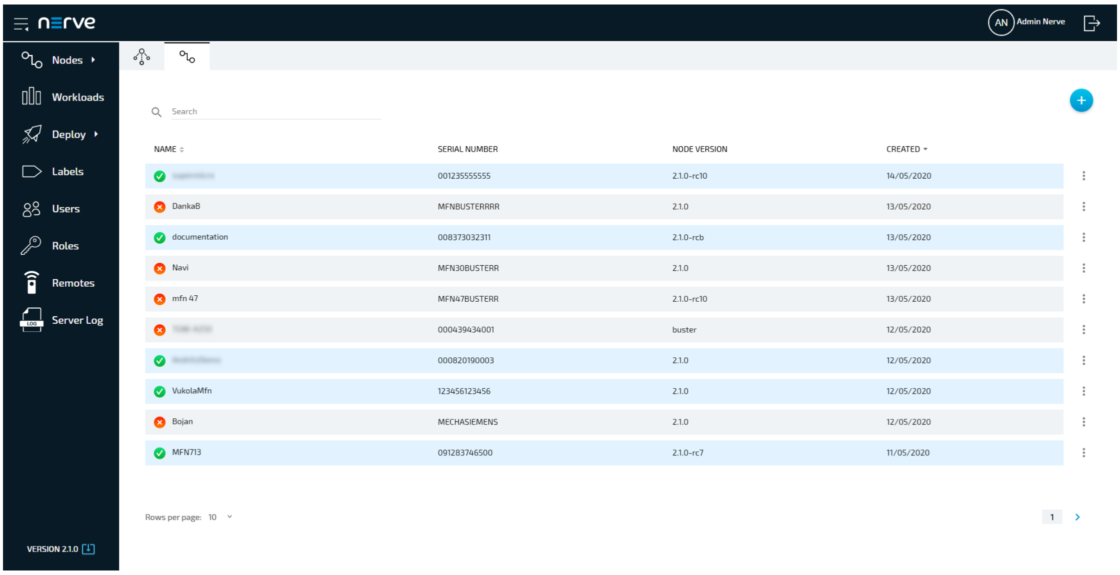
5. Click **OK** to delete the label.

Adding a label to a node

Adding a label to a node is done in the node menu. It is recommended to create labels before they are added to nodes.

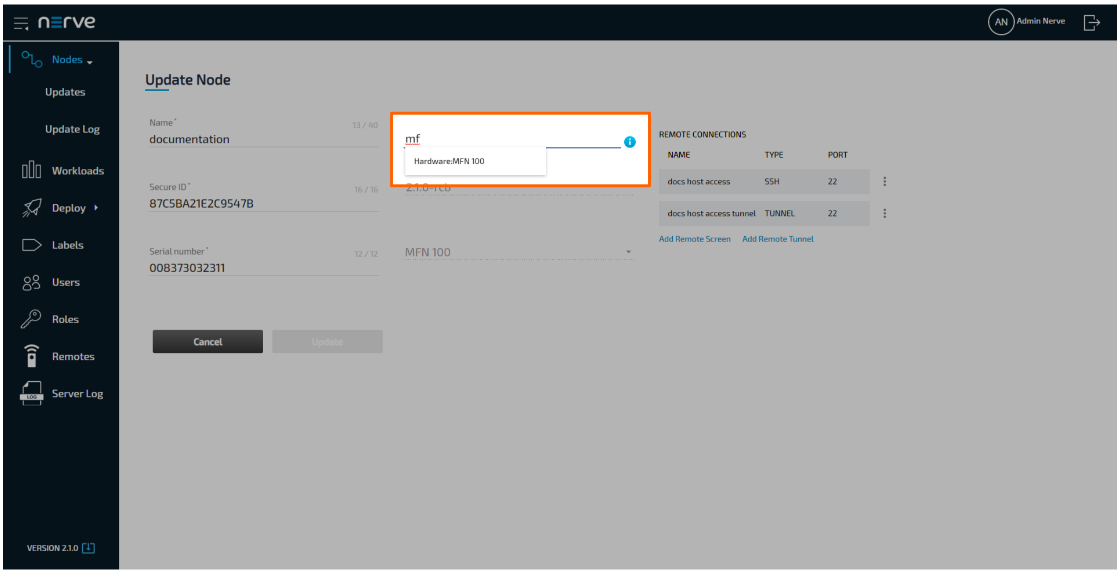
1. Select **Nodes** in the navigation on the left.
2. Select the nodes tab

   on the right to display the list of registered nodes.
3. Select a node to which a label will be assigned.



NAME	SERIAL NUMBER	NODE VERSION	CREATED
✓ [redacted]	001235555555	2.1.0-rc10	14/05/2020
✕ DankaB	MFNBUSTERRRR	2.1.0	13/05/2020
✓ documentation	008373032311	2.1.0-rcb	13/05/2020
✕ Navi	MFN30BUSTERR	2.1.0	13/05/2020
✕ mfn 47	MFN47BUSTERR	2.1.0-rc10	13/05/2020
✕ [redacted]	000439434001	buster	12/05/2020
✓ [redacted]	000820190003	2.1.0	12/05/2020
✓ VukolaMfn	123456123456	2.1.0	12/05/2020
✕ Bojan	MECHASIEMENS	2.1.0	12/05/2020
✓ MFN713	091283746500	2.1.0-rc7	11/05/2020

4. Start typing in the **Insert label** field. Available labels will be displayed if they match the typed string.



Update Node

Name* documentation 13 / 40

Secure ID* 87C5BA21E2C9547B 16 / 16

Serial number* 008373032311 12 / 12

mf

Hardware:MFN 100

MFN 100

REMOTE CONNECTIONS

NAME	TYPE	PORT
docs host access	SSH	22
docs host access tunnel	TUNNEL	22

Add Remote Screen Add Remote Tunnel

5. Select a label from the suggestions.

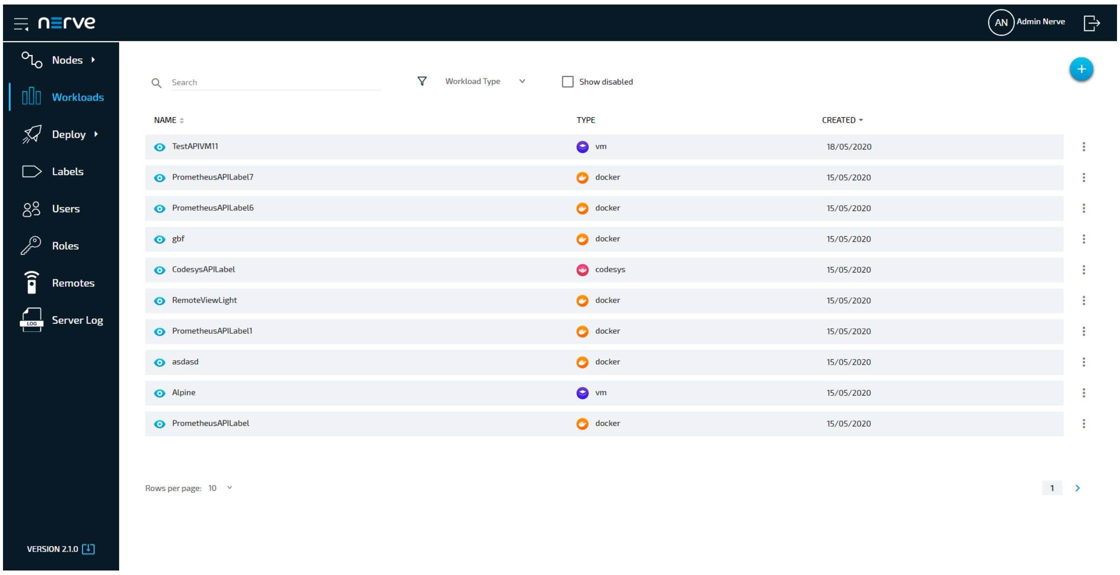
6. Select **Update** to save the changes to the node.

Adding a label to a workload

Adding a label to a workload is done in the workloads menu. It is recommended to create labels before they are added to workloads. Note that only pre-defined labels can be added to workloads.

1. Select **Workloads** in the navigation on the left.

2. Select a workload to which a label will be assigned.

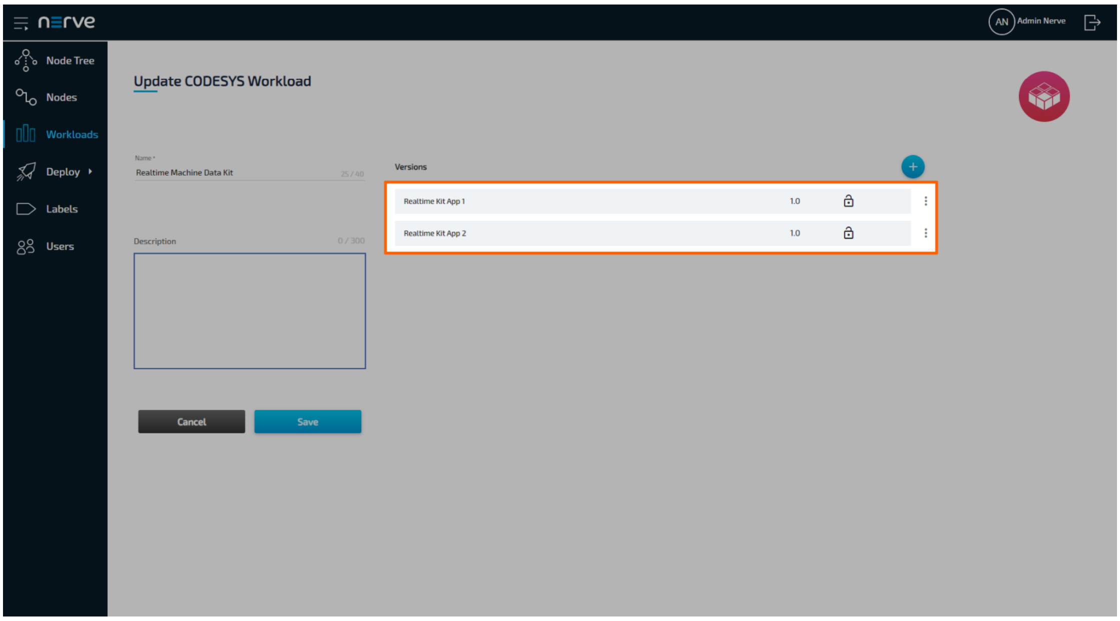


NAME	TYPE	CREATED
TestAPIVM11	vm	18/05/2020
PrometheusAPILabel7	docker	15/05/2020
PrometheusAPILabel6	docker	15/05/2020
gbf	docker	15/05/2020
CodesysAPILabel	codesys	15/05/2020
RemoteViewLight	docker	15/05/2020
PrometheusAPILabel1	docker	15/05/2020
asdasd	docker	15/05/2020
Alpine	vm	15/05/2020
PrometheusAPILabel	docker	15/05/2020

Rows per page: 10

1

3. Select the workload version to which a label will be assigned.



Update CODESYS Workload

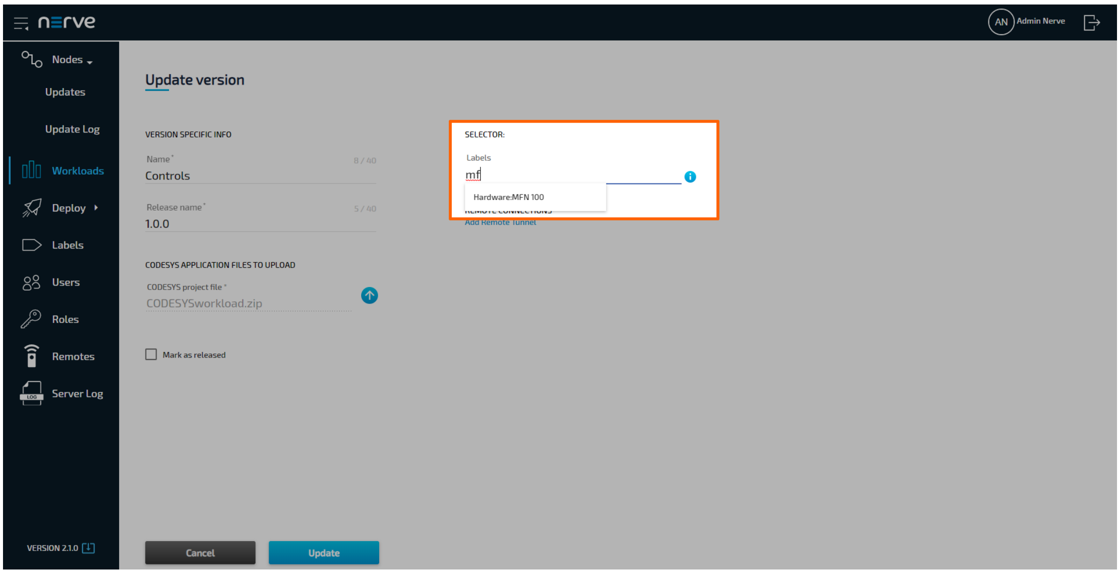
Name*
Realtime Machine Data Kit 25 / 40

Description 0 / 300

Versions

Realtime Kit App 1	1.0	
Realtime Kit App 2	1.0	

4. Start typing in the **Insert label** field. Available labels will be displayed if they match the typed string.



Update version

VERSION SPECIFIC INFO

Name* 8 / 40
Controls

Release name* 5 / 40
1.0.0

CODESYS APPLICATION FILES TO UPLOAD

CODESYS project file*
CODESYSworkload.zip

☐ Mark as released

SELECTOR:

Labels
mf
Hardware:MFN 100

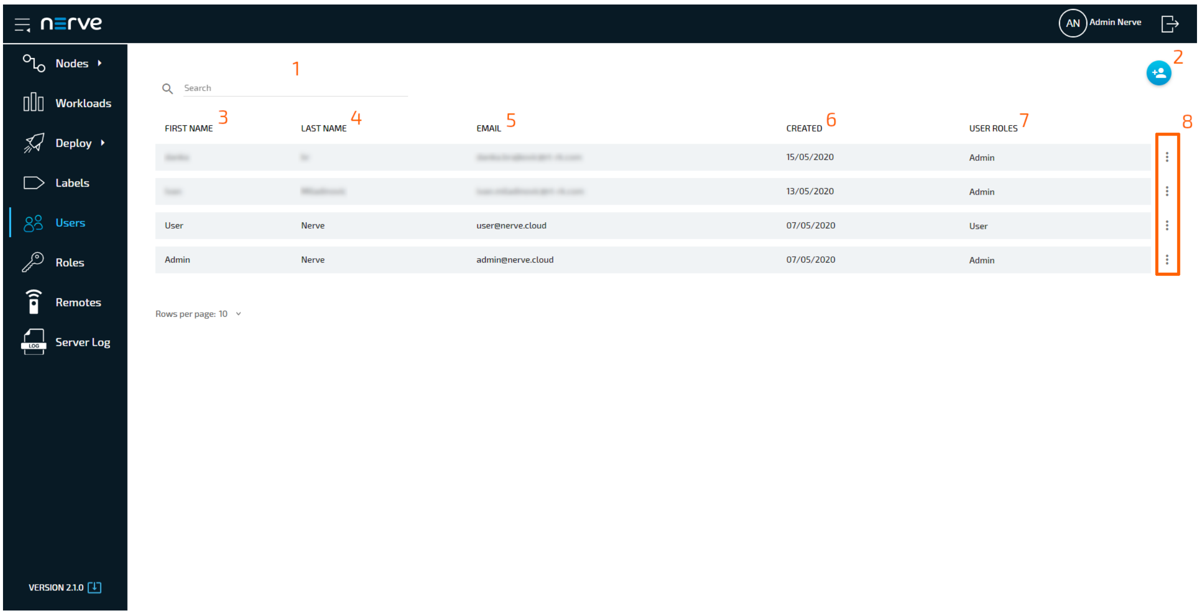
REMOTE CONNECTIONS
Add Remote Tunnel

5. Select a label from the suggestions.

6. Select **Update** to save the changes to the node.

Users

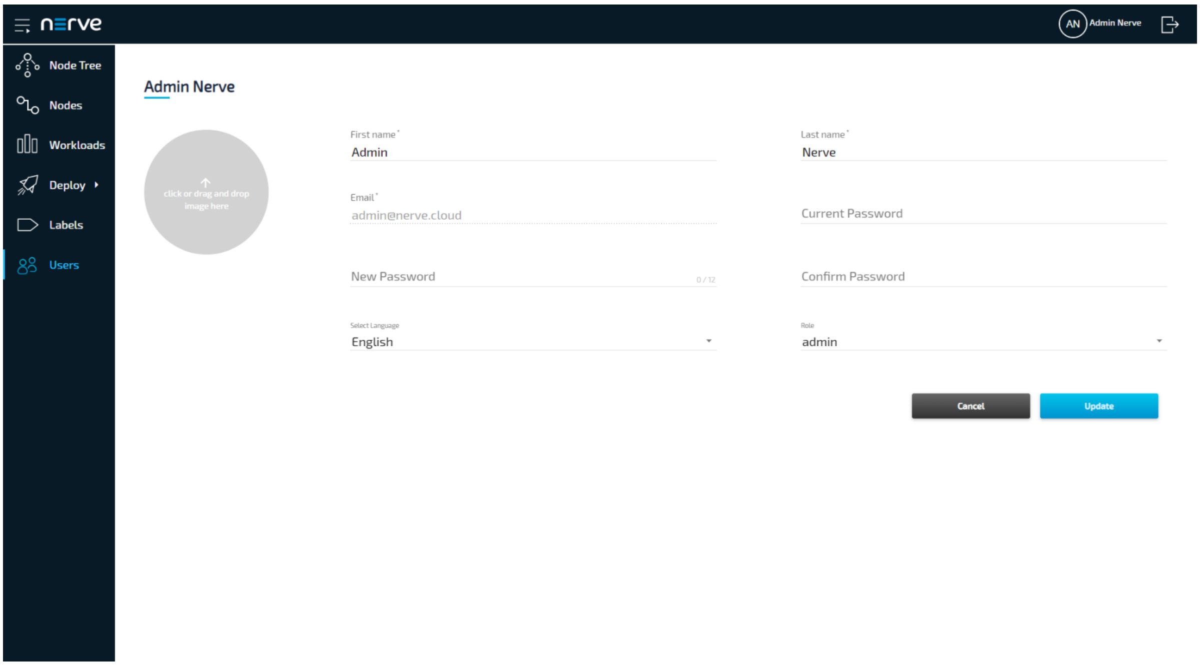
This menu offers a list of all registered users. Every user has their own user profile with details about the user. There is one default user that is created with the Management System. The credentials for this user can be found in the customer profile.



FIRST NAME	LAST NAME	EMAIL	CREATED	USER ROLES
[redacted]	[redacted]	[redacted]	15/05/2020	Admin
[redacted]	[redacted]	[redacted]	13/05/2020	Admin
User	Nerve	user@nerve.cloud	07/05/2020	User
Admin	Nerve	admin@nerve.cloud	07/05/2020	Admin

Item	Description
Search bar (1)	Use the search bar to filter the list of users. The columns **FIRST NAME**, **LAST NAME** and **EMAIL** are the targets of the search.
Add new user (2)	Select this icon to add a new user.
FIRST NAME (3)	This is the first name of the registered user. It is displayed in the upper-right corner when the user is logged in.
LAST NAME (4)	This is the last name of the registered user. It is displayed in the upper-right corner when the user is logged in.
EMAIL (5)	This is the e-mail address of the user. It is used as the username for logging in to the Management System. The system also uses this e-mail for sending the activation link and the instructions on how to reset the login password.
CREATED (6)	This is the date the user was created. The date format is DD/MM/YYYY.
USER ROLES (7)	The roles that are assigned to a user are displayed in this column.
Ellipsis menu (8)	Clicking here opens an overlay that allows the deletion of users.

Clicking any of the users leads to their user profile. An admin can edit the details of any user profile. This includes **First name**, **Last name**, **Select language** and **Role**. The e-mail address of a user cannot be changed. The password for an account can only be changed by the respective user.



Admin Nerve

First name*: Admin

Last name*: Nerve

Email*: admin@nerve.cloud

Current Password

New Password 0 / 12

Confirm Password

Select Language: English

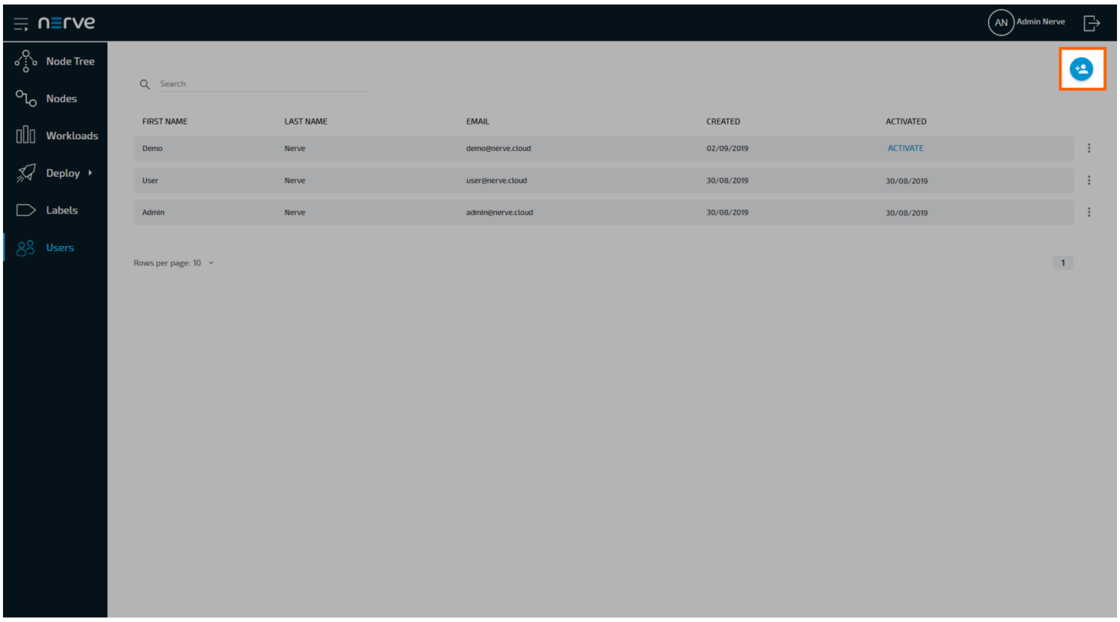
Role: admin

Item	Description
Profile Picture	Click here or drag and drop the image into the gray circle to upload a profile picture for the user. It is displayed in the upper-right corner when the user is logged in.
FIRST NAME	This is the first name of the registered user. It is displayed in the upper-right corner when the user is logged in.
LAST NAME	This is the last name of the registered user. It is displayed in the upper-right corner when the user is logged in.
EMAIL	This is the e-mail address of the user. It is used as the username for logging in to the Management System. The system also uses this e-mail for sending the activation link and the instructions on how to reset the login password.
Current Password	Enter the current password here when changing the password. This can only be done by the active user.
New Password	Enter the new password here when changing the password. This can only be done by the active user.
Confirm Password	Enter the new password again when changing the password. This can only be done by the active user.
Select Language	Select the Management System language from the drop-down menu.
Role	This defines the role of the user. Refer to Roles and Permissions for more information.

Adding a new user

Admin users or users having the **UI_USER:CREATE** permission assigned in a role need to create new users before these can set a password and log in to the Management System.

1. Select **Users** in the menu on the left side.
2. Click the **Add new user** symbol in the upper-right corner.



FIRST NAME	LAST NAME	EMAIL	CREATED	ACTIVATED
Demo	Nerve	demo@nerve.cloud	02/09/2019	ACTIVATE
User	Nerve	user@nerve.cloud	30/08/2019	30/08/2019
Admin	Nerve	admin@nerve.cloud	30/08/2019	30/08/2019

3. Enter the required information: **First name**, **Last name** and **Email**.

4. Select one or more roles from the drop-down menu under **Role**.
5. Select **Save** to create the user.

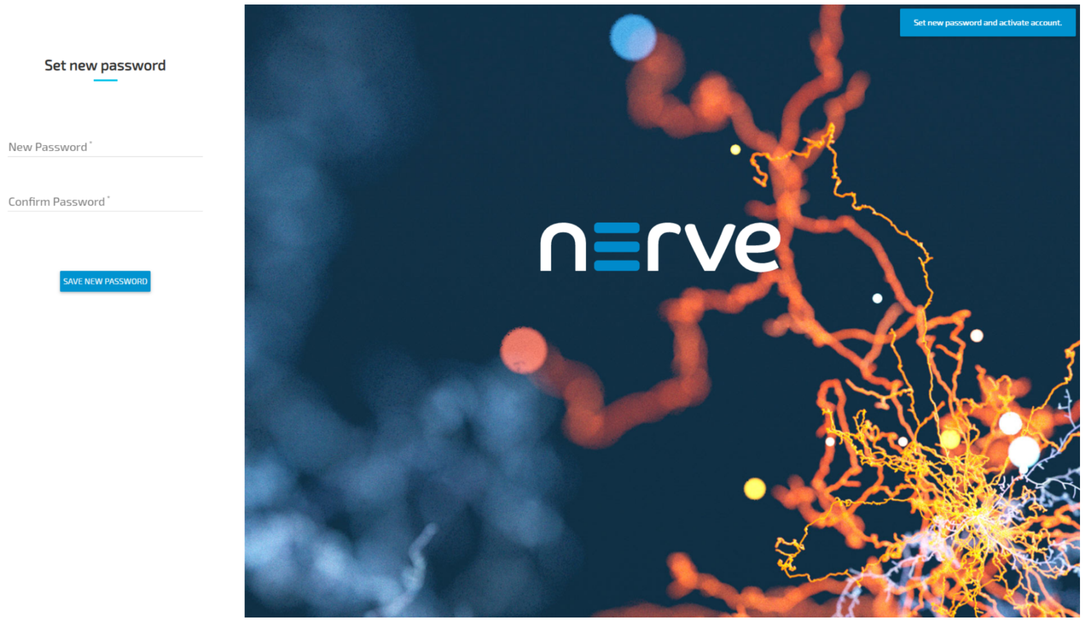
The user account needs to be activated after it has been created by clicking an activation link. The activation link is sent to the e-mail address that was specified during the creation of the user. The activation e-mail is sent automatically by the Management System shortly after the user is created.

Activating a user

Newly created users are sent an activation link to the specified e-mail address when the user was created. Following the link, the user will need to set their password in order to be activated and to be able to log in to the Management System.

1. Follow the activation link that was sent to the specified e-mail address.

2. Enter the new password to use for Management System access under **New Password** and **Confirm Password**.



Set new password

New Password *

Confirm Password *

SAVE NEW PASSWORD

> **NOTE**
>
> The password must contain at least one uppercase letter, one lowercase letter and one number. It must be at least 7 characters.

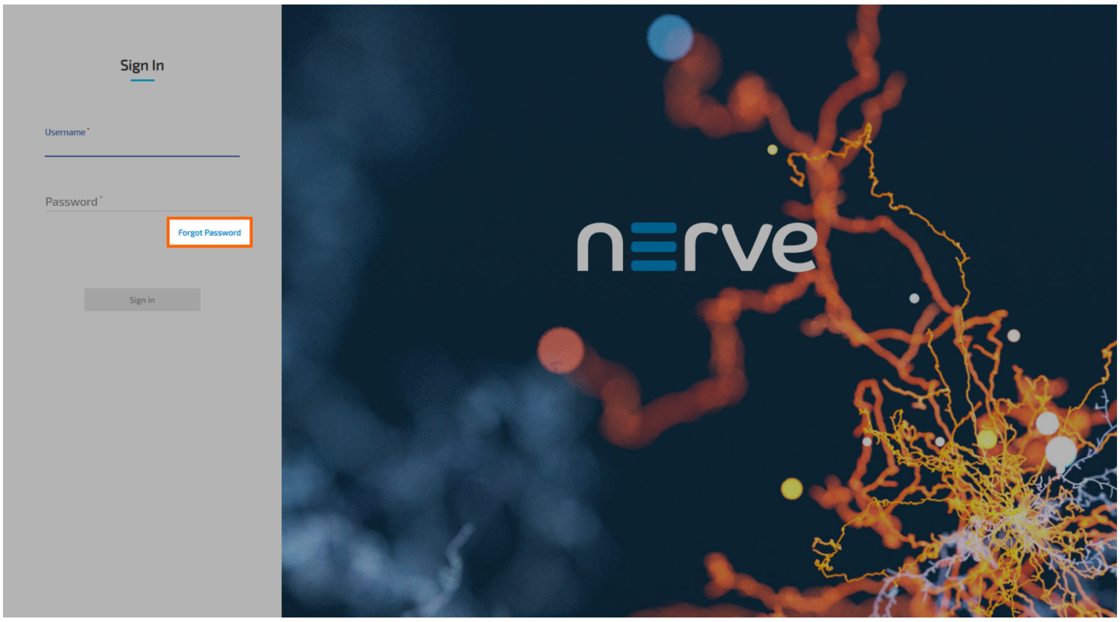
3. Select **SAVE NEW PASSWORD**.

The new user is now activated and can log in immediately with their e-mail address and the password they have defined.

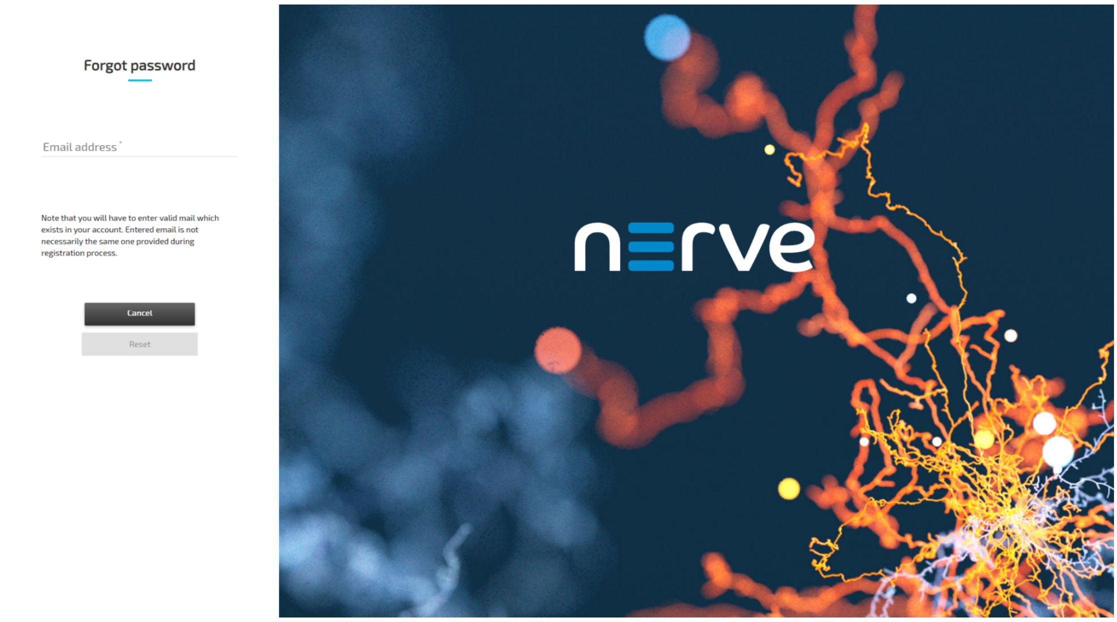
Resetting the password

In case a user does not remember their password, a request to reset the password can be sent from the login page of the Management System.

1. Enter the URL of the Management System in a browser.
2. Select **Forgot Password**.



3. Enter the e-mail address of the account that needs to reset the password.

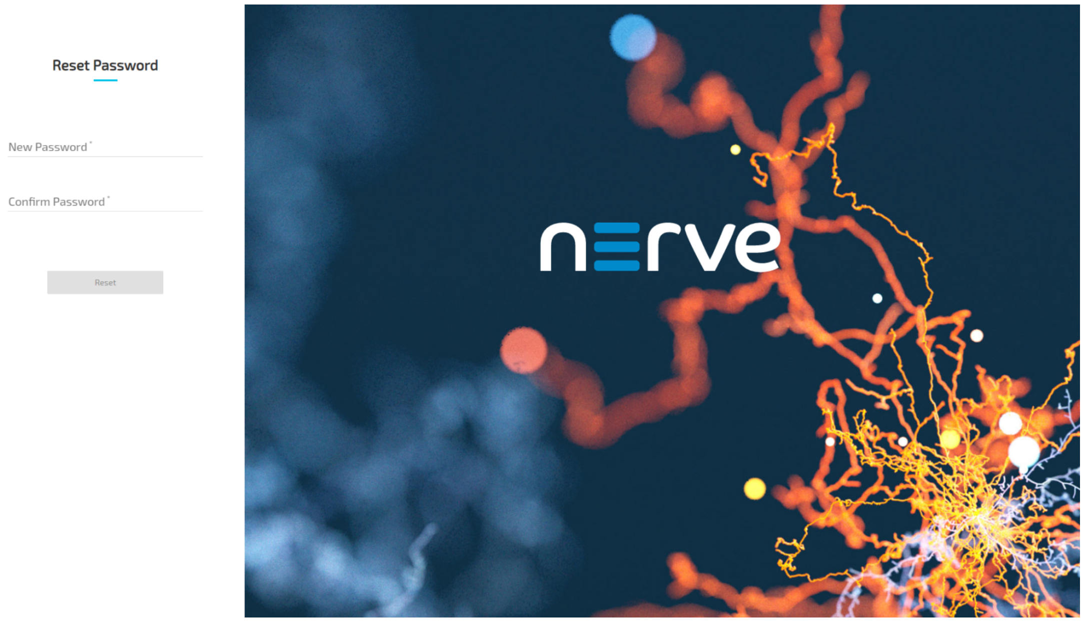


Note that you will have to enter valid mail which exists in your account. Entered email is not necessarily the same one provided during registration process.

4. Select **Reset**. An e-mail with instructions is sent to the e-mail address of the account.

5. Follow the link from the e-mail to reset the password.

6. Enter the new password for Management System access under **New Password** and **Confirm Password**.



7. Select **Reset** to save the new password.

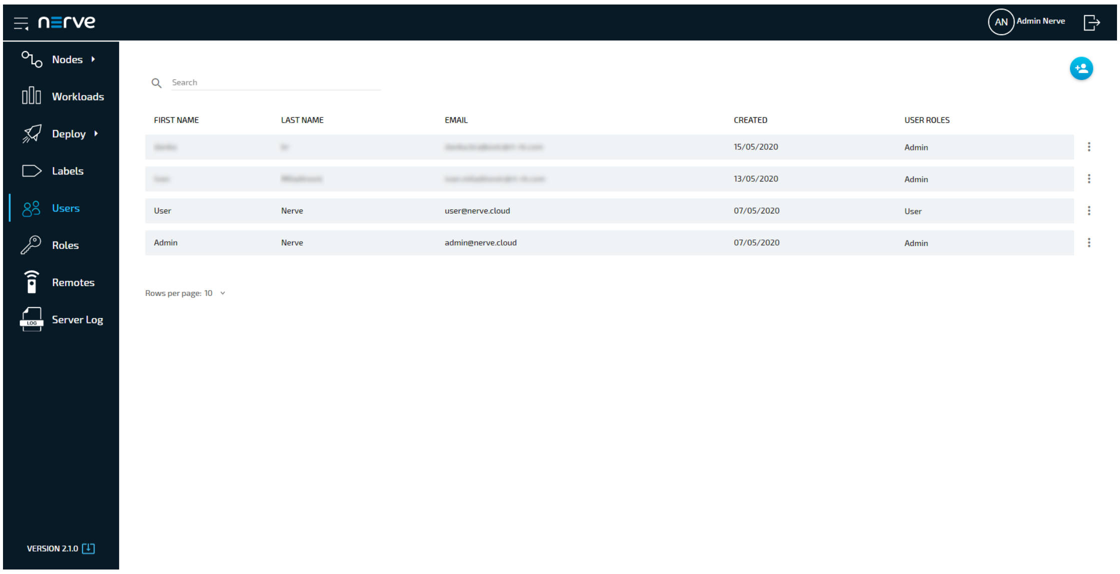
The password has been changed and the user can log in immediately with their e-mail address and the new password.

Deleting a user

Admin users or users having the **UI_USER:DELETE** permission assigned in a role can delete any user that is registered in the Management System.

1. Select **Users** in the menu on the left side.

2. Choose the user to delete.

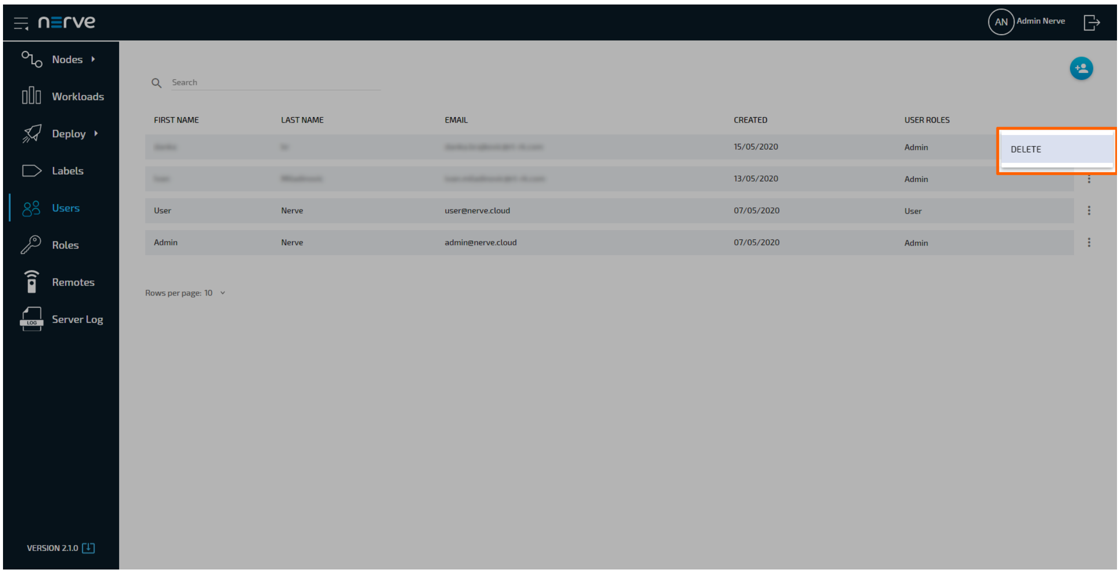


FIRST NAME	LAST NAME	EMAIL	CREATED	USER ROLES
[redacted]	[redacted]	[redacted]	15/05/2020	Admin
[redacted]	[redacted]	[redacted]	13/05/2020	Admin
User	Nerve	user@nerve.cloud	07/05/2020	User
Admin	Nerve	admin@nerve.cloud	07/05/2020	Admin

Rows per page: 10

3. Select the ellipsis menu to the right of the user in the list.

4. Select **DELETE** from the overlay that appeared.

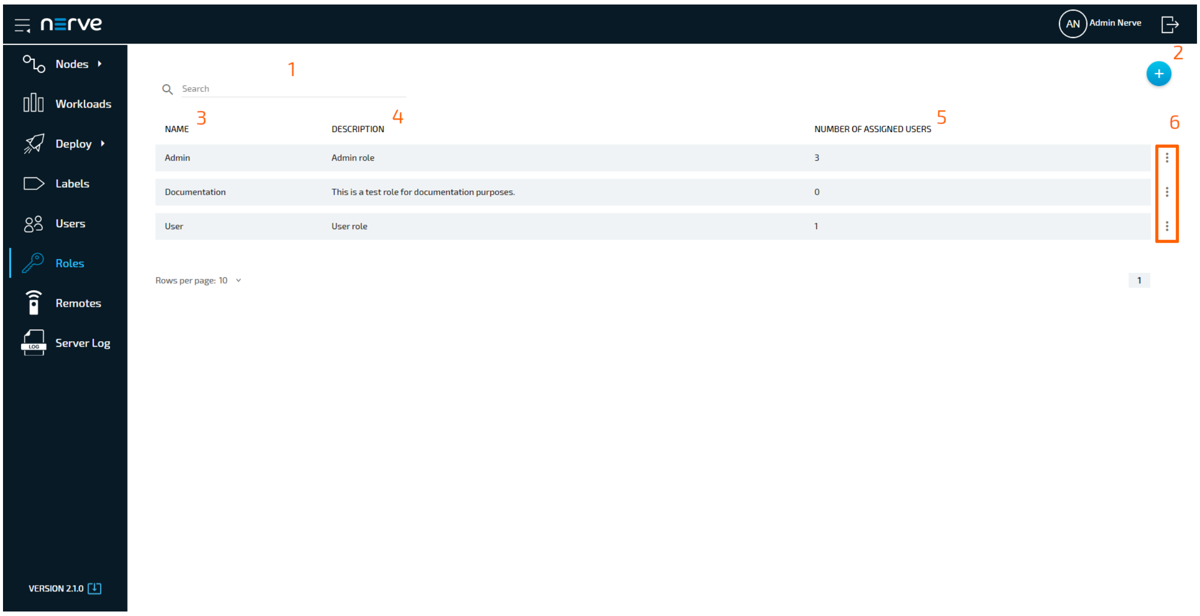


FIRST NAME	LAST NAME	EMAIL	CREATED	USER ROLES
[redacted]	[redacted]	[redacted]	15/05/2020	Admin
[redacted]	[redacted]	[redacted]	13/05/2020	Admin
User	Nerve	user@nerve.cloud	07/05/2020	User
Admin	Nerve	admin@nerve.cloud	07/05/2020	Admin

5. Click **OK** to confirm the deletion of the user.

Roles and permissions

Usage of the Management System is restricted by role-based access control (RBAC), meaning that users in the Management System are assigned roles. These roles are assigned a set of UI permissions and API permissions. Four user roles — **Admin**, **User**, **Logs** and **Datapath** — are available by default. Multiple roles can be assigned to one user. A user that is assigned multiple roles is granted the combined permissions of each role. Select **Roles** in the navigation on the left to reach a list of all available roles:



NAME	DESCRIPTION	NUMBER OF ASSIGNED USERS
Admin	Admin role	3
Documentation	This is a test role for documentation purposes.	0
User	User role	1

Item	Description
Search bar (1)	Use the search bar to filter roles by name.
Add new role (2)	Click here to add a new role.

Item	Description
NAME (3)	This is the name of the role that was defined when it was created. This name is also used when the role is assigned to users.
DESCRIPTION (4)	This is a description that gives more information about each user role.
NUMBER OF ASSIGNED USERS (5)	The number of users that this role has been assigned to is displayed here.
Ellipsis menu (6)	Clicking here opens an overlay that allows deleting roles.

The **Admin** role has all permissions assigned and cannot be edited. The **User** role has limited permissions. A user that has the **User** role assigned is not allowed to perform changes to the system such as adding or removing nodes, creating workloads or establishing remote connections among others. Users with the **User** role can work with the node tree and deploy workloads to nodes. The **Logs** and **Datapath** roles have permissions assigned to grant users access to previews of the logging and Data Services features.

UI permissions

UI permissions reflect the permissions of the frontend. They are relevant for a users interaction with the Management System. Below is the list of available UI permissions with descriptions.
As some actions depend on other actions, the system automatically selects and deselects permissions, including API permissions that are required for these actions.
When creating a new role for a regular user that will operate the Management System, use the UI permissions as a starting point and change API permissions only if necessary. However, note that changes to API permissions should only be done by users with expert knowledge.

The tables below are separated by the part of the Management System the permissions affect.

Workload deployment

Permission	Name	Description
Deploy workload	**UI_DEPLOY:DEPLOY**	Permission that grants the user the rights to deploy a workload to a node.
Force stop campaign task	**UI_DEPLOY:FORCE_CANCEL_ONE**	Permission that grants the user access to see the force stop button for a deployment campaign.
Delete workload deploy log	**UI_DEPLOY:LOG_DELETE**	Permission that grants the user the rights to delete a log entry in the workload deployment log.
Reset deployment task	**UI_DEPLOY:LOG_RESET**	Permission that grants the user the rights to reset a task in the workload deployment log.
Reset all deployment tasks	**UI_DEPLOY:LOG_RESET_ALL**	Permission that grants the user the rights to reset all tasks in the workload deployment log.
Preview workload deploy log	**UI_DEPLOY:LOG_VIEW**	Permission that grants the user the rights to view a log entry in the workload deployment log.

Permission	Name	Description
Access "Deploy" -> "Dry run"	**UI_SUBNAV_DEPLOY_DRY_RUN:VIEW**	Permission that grants the user the rights to see the dry run entry in the navigation menu and the rights to perform the dry run action with workloads to nodes.
Access "Deploy" -> "Log"	**UI_SUBNAV_DEPLOY_LOG:VIEW**	Permission that grants the user the rights to see the deployment log entry in the navigation menu and the rights to view the workload deployment log.

Labels

Permission	Name	Description
Create new label	**UI_LABEL:CREATE**	Permission that grants the user the rights to create new labels.
Delete label	**UI_LABEL:DELETE**	Permission that grants the user the rights to delete labels.
Edit existing label	**UI_LABEL:EDIT**	Permission that grants the user the rights to edit labels.
Group labels by key	**UI_LABEL:GROUP**	Permission that grants the user the rights to group labels by key.
Merge labels to one	**UI_LABEL:MERGE**	Permission that grants the user the rights to merge multiple labels.
Preview list of labels	**UI_LABEL:VIEW**	Permission that grants the user the rights to view the details of a label.

Navigation menu

Permission	Name	Description
Access Datapath Feature Preview	**UI_NAV_DATAPATH:VIEW**	Permission that grants the user access to the preview version of the Data Services feature.
"Deploy" section	**UI_NAV_DEPLOY:VIEW**	Permission that grants the user the rights to view the deployment menu.
"Labels" section	**UI_NAV_LABELS:VIEW**	Permission that grants the user the rights to see the labels entry in the navigation menu and the rights to list labels.
"Nodes" section	**UI_NAV_NODES:VIEW**	Permission that grants the user the rights to see the nodes entry in the navigation menu and the rights to list nodes.

Permission	Name	Description
"Remotes" section	**UI_NAV_REMOTE_CONNECTIONS:VIEW**	Permission that grants the user the rights to view active remote connections.
"Roles" section	**UI_NAV_ROLES:VIEW**	Permission that grants the user the rights to see the roles entry in the navigation menu and the rights to list roles.
Access "Server Log"	**UI_NAV_SERVER_LOGS:VIEW**	Permission that grants the user the rights to list internal server logs.
"Users" section	**UI_NAV_USERS:VIEW**	Permission that grants the user the rights to see the users entry in the navigation menu and the rights to list users.
"Workloads" section	**UI_NAV_WORKLOADS:VIEW**	Permission that grants the user the rights to see the workloads entry in the navigation menu and the rights to list workloads.

Nodes

Permission	Name	Description
Create new node	**UI_NODE:CREATE**	Permission that grants the user the rights to create new nodes.
Delete node	**UI_NODE:DELETE**	Permission that grants the user the rights to delete nodes.
Edit existing node	**UI_NODE:EDIT**	Permission that grants the user the rights to edit nodes.
Show logs of node	**UI_NODE:SHOW_LOGS**	Permission that grants the user the rights to view internal node logs.
Preview node	**UI_NODE:VIEW**	Permission that grants the user the rights to view details of a node.
Change a Node logging level configuration	**UI_NODE_LOG_LEVEL:MANAGE_LOG_LEVELS**	Permission that grants the user the rights to change the logging level settings of a Node.

Permission	Name	Description
Delete node update log	**UI_NODE_UPDATE:LOG_DELETE**	Permission that grants the user the rights to delete a log entry in the node update log.
Show node update log	**UI_NODE_UPDATE:LOG_VIEW**	Permission that grants the user the rights to view the details of a node update log entry.
Update node	**UI_NODE_UPDATE:UPDATE**	Permission that grants the user the rights to update a node.
Access "Nodes" -> "Updates"	**UI_SUBNAV_NODE_UPDATE:VIEW**	Permission that grants the user the rights to see the updates sub-entry in the navigation menu.
Access "Nodes" -> "Log"	**UI_SUBNAV_NODE_UPDATE_LOG:VIEW**	Permission that grants the user the rights to see the log sub-entry in the navigation menu.

Node tree

Permission	Name	Description
Add new tree item	**UI_NODE_TREE:ADD**	Permission that grants the user the rights to add new elements in the node tree.
Delete tree item	**UI_NODE_TREE:DELETE**	Permission that grants the user the rights to delete an element of the node tree.
Edit tree item	**UI_NODE_TREE:EDIT**	Permission that grants the user the rights to edit an element of the node tree.
Node tree manipulation	**UI_NODE_TREE:MANIPULATE**	Permission that grants the user the rights to manipulate the elements of the node tree structure, i.e perform changes to position and order.
Preview node details in Node tree	**UI_NODE_TREE:NODE_DETAILS**	Permission that grants the user the rights to view the node details of a node in the node tree.

Remote connections

Permission	Name	Description
Connect over remote connection	**UI_REMOTE_CONN:CONNECT**	Permission that grants the user the rights to connect to a host through a remote connection.
Create remote connection	**UI_REMOTE_CONN:CREATE**	Permission that grants the user the rights to establish a new remote connection.
Delete remote connection	**UI_REMOTE_CONN:DELETE**	Permission that grants the user the rights to delete remote connections.
Edit remote connection	**UI_REMOTE_CONN:EDIT**	Permission that grants the user the rights to edit existing remote connections.
List all remote connections	**UI_REMOTE_CONN:LIST**	Permission that grants the user the rights to list all remote connections in the node and workload details.
Preview remote connection	**UI_REMOTE_CONN:VIEW**	Permission that grants the user the rights to view the details of a remote connection.
Terminate remote connection	**UI_REMOTE_CONNECTIONS:TERMINATE**	Permission that grants the user the rights to terminate active remote connections.

Roles

Permission	Name	Description
Create new role	**UI_ROLE:CREATE**	Permission that grants the user the rights to create new roles.
Delete role	**UI_ROLE:DELETE**	Permission that grants the user the rights to delete roles.
Edit role	**UI_ROLE:EDIT**	Permission that grants the user the rights to edit roles.
Preview role	**UI_ROLE:VIEW**	Permission that grants the user the rights to view the details of a role.

Server log

Permission	Name	Description
Preview server logs	**UI_SERVER_LOGS:VIEW**	Permission that grants the user the rights to list internal server logs.

User menu

Permission	Name	Description
Create new user profile	**UI_USER:CREATE**	Permission that grants the user the rights to create new users.
Delete user profile	**UI_USER:DELETE**	Permission that grants the user the rights to delete users.
Edit user profile	**UI_USER:EDIT**	Permission that grants the user the rights to edit the profiles of other users.
Preview user profile	**UI_USER:VIEW**	Permission that grants the user the rights to view the details of a user.
Edit user settings	**UI_USER_SETTINGS:UPDATE**	Permission that grants the user the rights to update their user settings.
Preview user settings	**UI_USER_SETTINGS:VIEW**	Permission that grants the user the rights to view their user settings.

Management System update

Permission	Name	Description
List available Cloud app versions	**UI_VERSION:LIST**	Permission that grants the user the rights to list all available version of the Management System.
Upload Cloud app versions	**UI_VERSION:UPDATE**	Permission that grants the user the rights to update the Management System.

Workload management

Permission	Name	Description
Create new workload	**UI_WORKLOAD:CREATE**	Permission that grants the user the rights to create new workloads.
Delete workload	**UI_WORKLOAD:DELETE**	Permission that grants the user the rights to delete workloads.
Disable workload	**UI_WORKLOAD:DISABLE**	Permission that grants the user the rights to disable workloads.
Edit workload	**UI_WORKLOAD:EDIT**	Permission that grants the user the rights to edit workload details (name and description).
Preview workload	**UI_WORKLOAD:VIEW**	Permission that grants the user the rights to view the details of a workload.
Create workload version	**UI_WORKLOAD:VERSION_CREATE**	Permission that grants the user the rights to create new workload versions.
Delete workload version	**UI_WORKLOAD:VERSION_DELETE**	Permission that grants the user the rights to delete workload versions.

Permission	Name	Description
Edit workload version	**UI_WORKLOAD:VERSION_EDIT**	Permission that grants the user the rights to edit workload versions.
Preview workload version	**UI_WORKLOAD:VERSION_VIEW**	Permission that grants the user the rights to view workload versions.
Control deployed workload	**UI_WORKLOAD_CONTROL:CONTROL**	Permission that grants the user full control over status and life cycle of workloads deployed to nodes.
List deployed workloads	**UI_WORKLOAD_CONTROL:LIST**	Permission that grants the user the rights to list workloads that are deployed to a node.
Preview deployed workloads	**UI_WORKLOAD_CONTROL:VIEW**	Permission that grants the user the rights to view the details of a workload deployed to a node.

API permissions

API permissions reflect the permissions of the server backend. They are primarily relevant for automating the Management System through API calls. When creating a role in the Management System for a program, they can be selected without selecting UI permissions beforehand. When creating a new role for a regular user that will operate the Management System, use the UI permissions as a starting point and change API permissions only if necessary. Note that API permissions should only be handled by persons with expert knowledge.

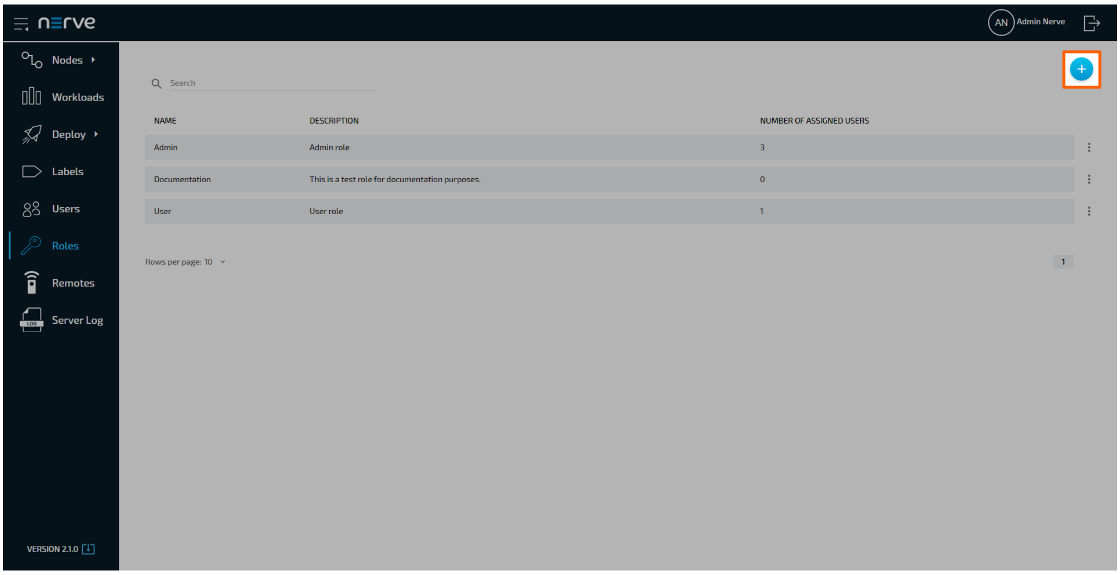
Adding a new role

When adding a new role, it depends whether the role is going to be created for regular users or programs. When creating a new role for a regular user that will operate the Management System, use the UI permissions as a starting point and change API permissions only if necessary. When creating a role for a program, API permissions can be selected without selecting UI permissions beforehand. Note that API permissions should only be handled by persons with expert knowledge.

Selecting one permission might automatically select other permissions, which are needed to perform the task indicated by the selected permission. An example: if a user is permitted to deploy a workload, then the same user is also permitted to view the list of workloads. Associated API permissions will also be selected. Note that deselecting a permission might also deselect linked permissions.

1. Select **Roles** in the navigation on the left.

2. Click the plus symbol (**Add new role**) in the upper-right corner.

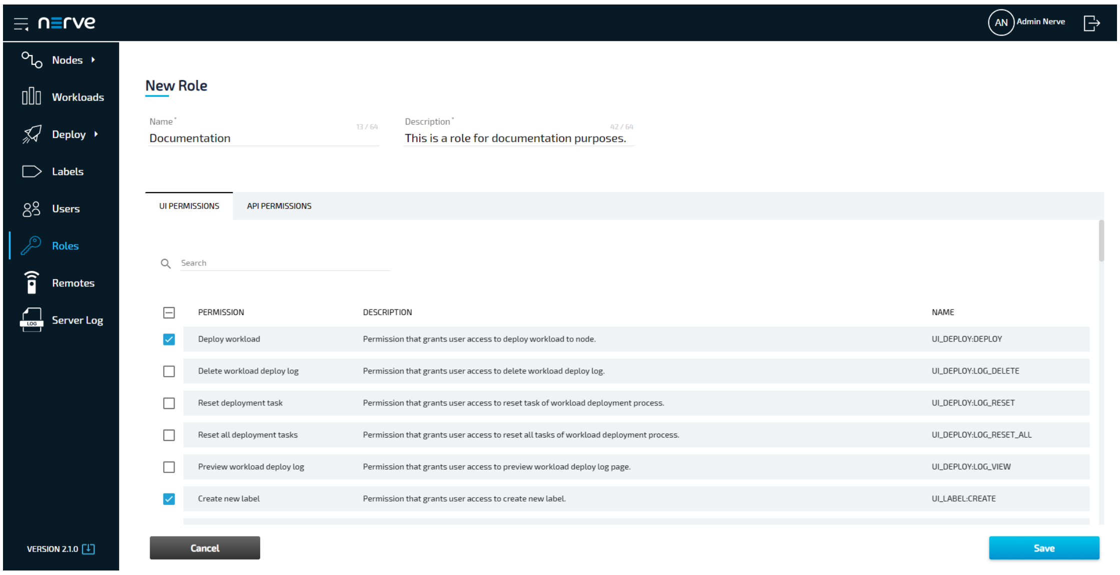


NAME	DESCRIPTION	NUMBER OF ASSIGNED USERS
Admin	Admin role	3
Documentation	This is a test role for documentation purposes.	0
User	User role	1

3. Enter a **Name** and a **Description** at the top.

4. Select the **UI PERMISSIONS** tab.

5. Tick the checkboxes next to the desired permissions.



New Role

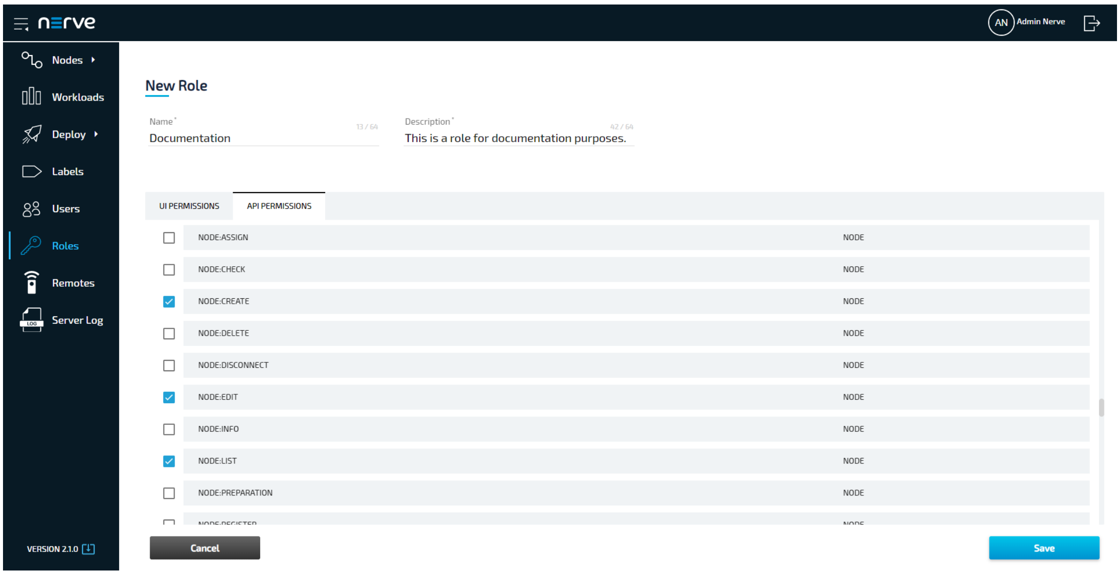
Name* 13 / 64
Documentation

Description* 42 / 64
This is a role for documentation purposes.

⊟	PERMISSION	DESCRIPTION	NAME
☑	Deploy workload	Permission that grants user access to deploy workload to node.	UI_DEPLOY:DEPLOY
□	Delete workload deploy log	Permission that grants user access to delete workload deploy log.	UI_DEPLOY:LOG_DELETE
□	Reset deployment task	Permission that grants user access to reset task of workload deployment process.	UI_DEPLOY:LOG_RESET
□	Reset all deployment tasks	Permission that grants user access to reset all tasks of workload deployment process.	UI_DEPLOY:LOG_RESET_ALL
□	Preview workload deploy log	Permission that grants user access to preview workload deploy log page.	UI_DEPLOY:LOG_VIEW
☑	Create new label	Permission that grants user access to create new label.	UI_LABEL:CREATE

6. Select the **API PERMISSIONS** tab.

7. Tick or untick the permissions that need to be changed.



New Role

Name* Documentation 13 / 64

Description* This is a role for documentation purposes. 42 / 64

UI PERMISSIONS | API PERMISSIONS

□	NODE:ASSIGN	NODE
□	NODE:CHECK	NODE
☑	NODE:CREATE	NODE
□	NODE:DELETE	NODE
□	NODE:DISCONNECT	NODE
☑	NODE:EDIT	NODE
□	NODE:INFO	NODE
☑	NODE:LIST	NODE
□	NODE:PREPARATION	NODE
□	NODE:REGISTER	NODE

NOTE

Make sure to review the selected permissions for completeness before saving the role. The system automatically selects and deselects permissions that are linked and might have added or removed desired permissions when permissions where selected or deselected.

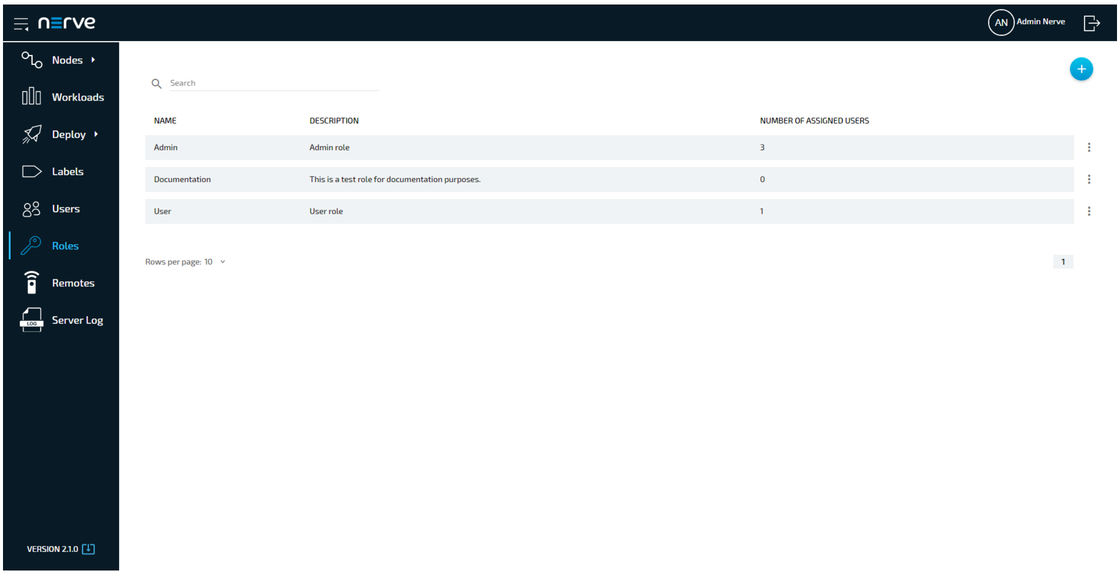
8. Click **Save**.

Editing a role

Note that editing the permissions of a role changes the permissions for users who are already assigned this role.

1. Select **Roles** in the navigation on the left.

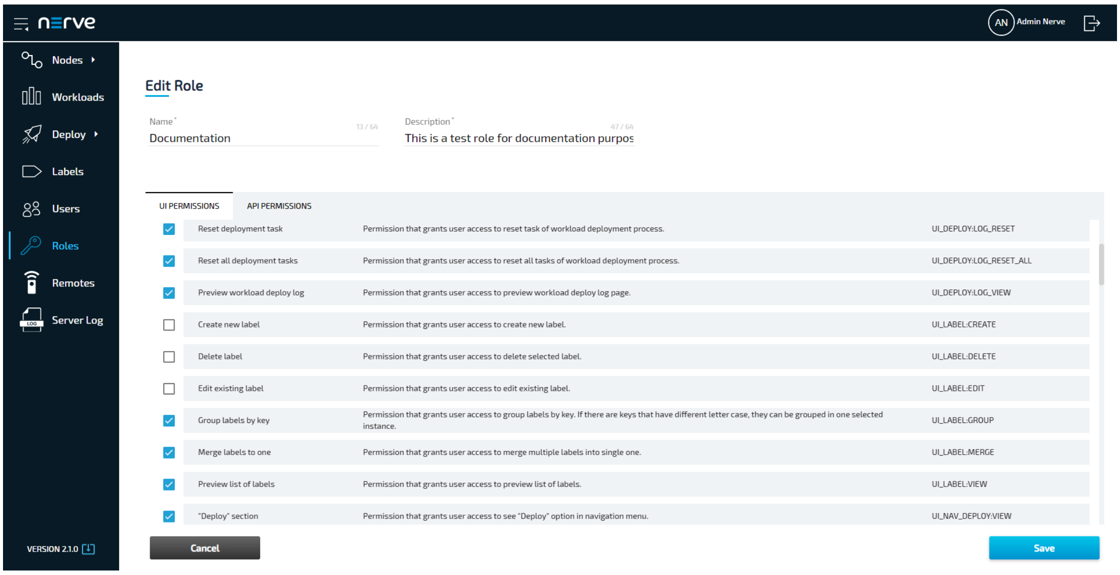
2. Select a role from the list.



NAME	DESCRIPTION	NUMBER OF ASSIGNED USERS
Admin	Admin role	3
Documentation	This is a test role for documentation purposes.	0
User	User role	1

Rows per page: 10 1

3. Edit **Name** and **Description** at the top.
4. Select the **UI PERMISSIONS** tab.
5. Tick or untick the permissions that need to be changed.



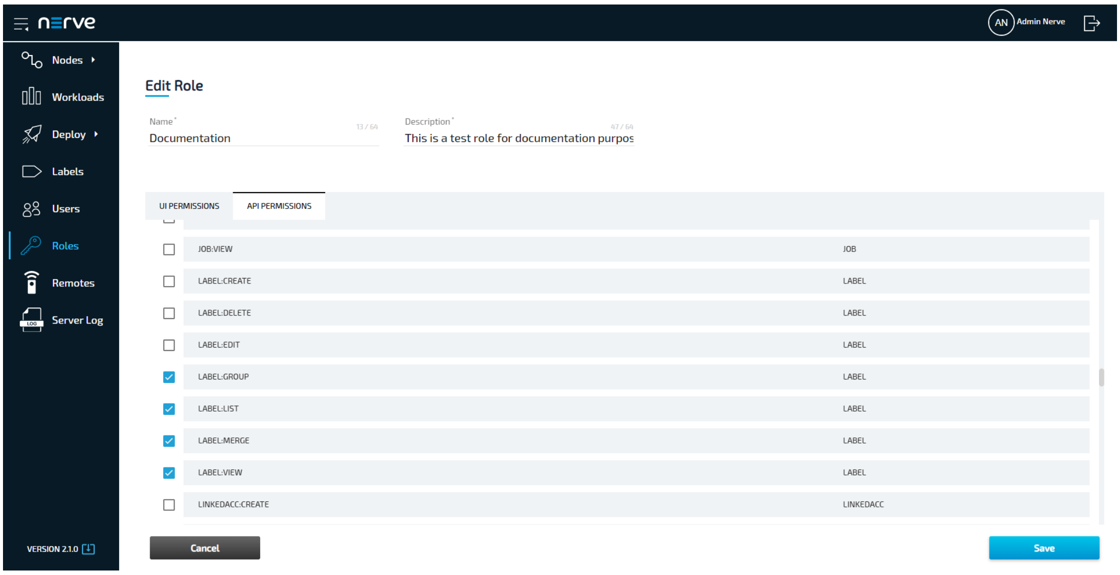
Edit Role

Name* 13 / 64
Documentation

Description* 47 / 64
This is a test role for documentation purpos

☑	Reset deployment task	Permission that grants user access to reset task of workload deployment process.	UI_DEPLOY:LOG_RESET
☑	Reset all deployment tasks	Permission that grants user access to reset all tasks of workload deployment process.	UI_DEPLOY:LOG_RESET_ALL
☑	Preview workload deploy log	Permission that grants user access to preview workload deploy log page.	UI_DEPLOY:LOG_VIEW
□	Create new label	Permission that grants user access to create new label.	UI_LABEL:CREATE
□	Delete label	Permission that grants user access to delete selected label.	UI_LABEL:DELETE
□	Edit existing label	Permission that grants user access to edit existing label.	UI_LABEL:EDIT
☑	Group labels by key	Permission that grants user access to group labels by key. If there are keys that have different letter case, they can be grouped in one selected instance.	UI_LABEL:GROUP
☑	Merge labels to one	Permission that grants user access to merge multiple labels into single one.	UI_LABEL:MERGE
☑	Preview list of labels	Permission that grants user access to preview list of labels.	UI_LABEL:VIEW
☑	"Deploy" section	Permission that grants user access to see "Deploy" option in navigation menu.	UI_NAV_DEPLOY:VIEW

6. Select the **API PERMISSIONS** tab.
7. Tick or untick the permissions that need to be changed.



Edit Role

Name* 13 / 64
Documentation

Description* 47 / 64
This is a test role for documentation purpos

□	JOB:VIEW	JOB
□	LABEL:CREATE	LABEL
□	LABEL:DELETE	LABEL
□	LABEL:EDIT	LABEL
☑	LABEL:GROUP	LABEL
☑	LABEL:LIST	LABEL
☑	LABEL:MERGE	LABEL
☑	LABEL:VIEW	LABEL
□	LINKEDACC:CREATE	LINKEDACC

NOTE

Make sure to review the selected permissions for completeness before saving the role. The system automatically selects and deselects permissions that are linked and might have added or removed desired permissions when permissions where selected or deselected.

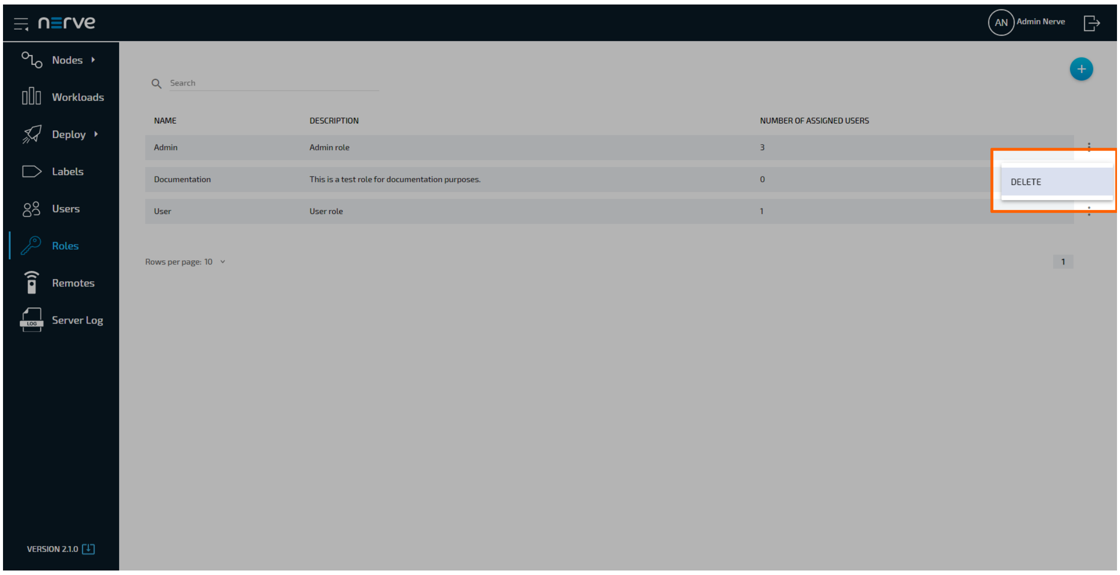
8. Click **Save**.

Deleting a role

Note that a role cannot be deleted if it is assigned to a user.

1. Select **Roles** in the navigation on the left.
2. Choose a role from the list.
3. Click the ellipsis menu next to the role.

4. Select **DELETE** in the overlay that appeared.



NAME	DESCRIPTION	NUMBER OF ASSIGNED USERS
Admin	Admin role	3
Documentation	This is a test role for documentation purposes.	0
User	User role	1

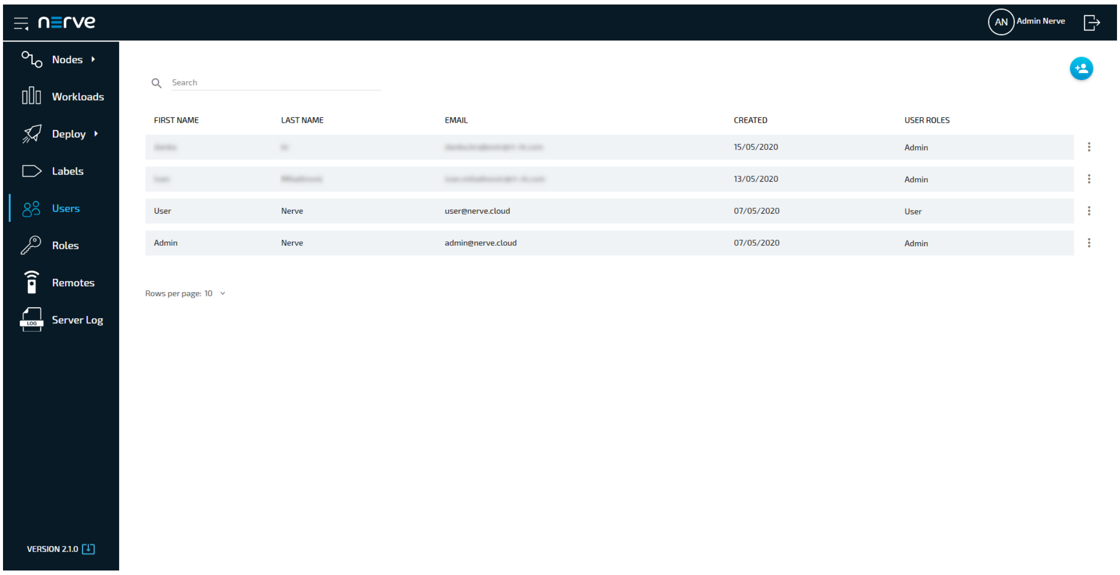
5. Select **OK** to delete the role.

Assigning a role to a user

Assigning a role is done in the users menu. Users can be assigned multiple roles. A user that is assigned multiple roles is granted the combined permissions of each role.

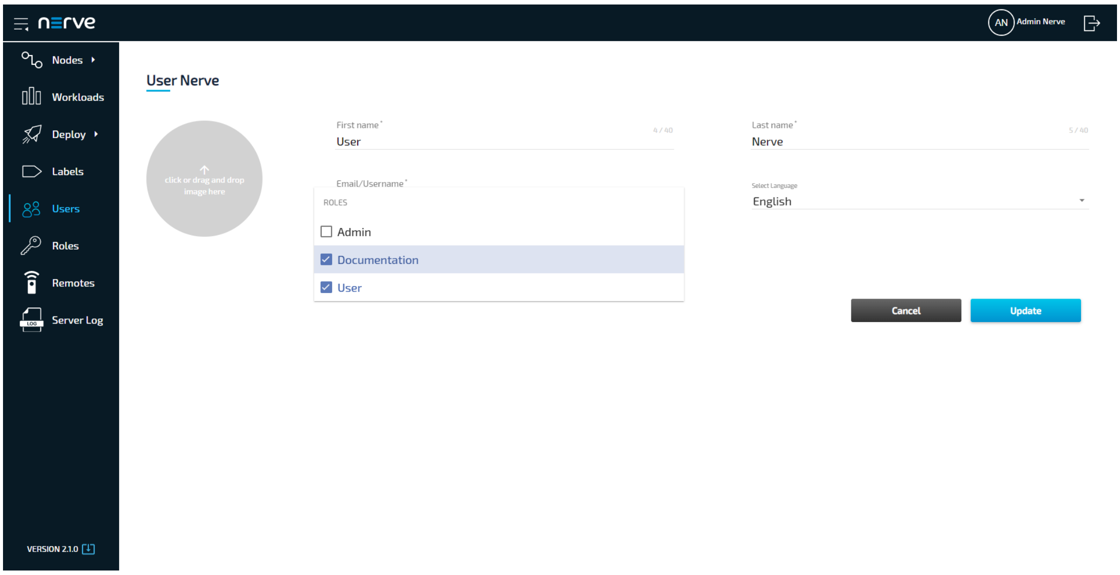
1. Select **Users** in the navigation on the left.

2. Select a user from the list.



FIRST NAME	LAST NAME	EMAIL	CREATED	USER ROLES
[redacted]	[redacted]	[redacted]	15/05/2020	Admin
[redacted]	[redacted]	[redacted]	13/05/2020	Admin
User	Nerve	user@nerve.cloud	07/05/2020	User
Admin	Nerve	admin@nerve.cloud	07/05/2020	Admin

3. Click the field under **Role** to open a drop-down menu.
4. Tick one or more roles that will be assigned to this user. Note that at least one role must be selected.



User Nerve

First name* User 4 / 40

Last name* Nerve 5 / 40

Email/Username*

ROLES

☐ Admin

☑ Documentation

☑ User

Select Language English

5. Select **Update** in the lower-right.

Device Guide

Device Guide

The device guide is an extension of the user guide. It gives an overview of supported Nerve Devices and the device specific information that is required for operating Nerve Blue software.

Each device chapter includes the following information:

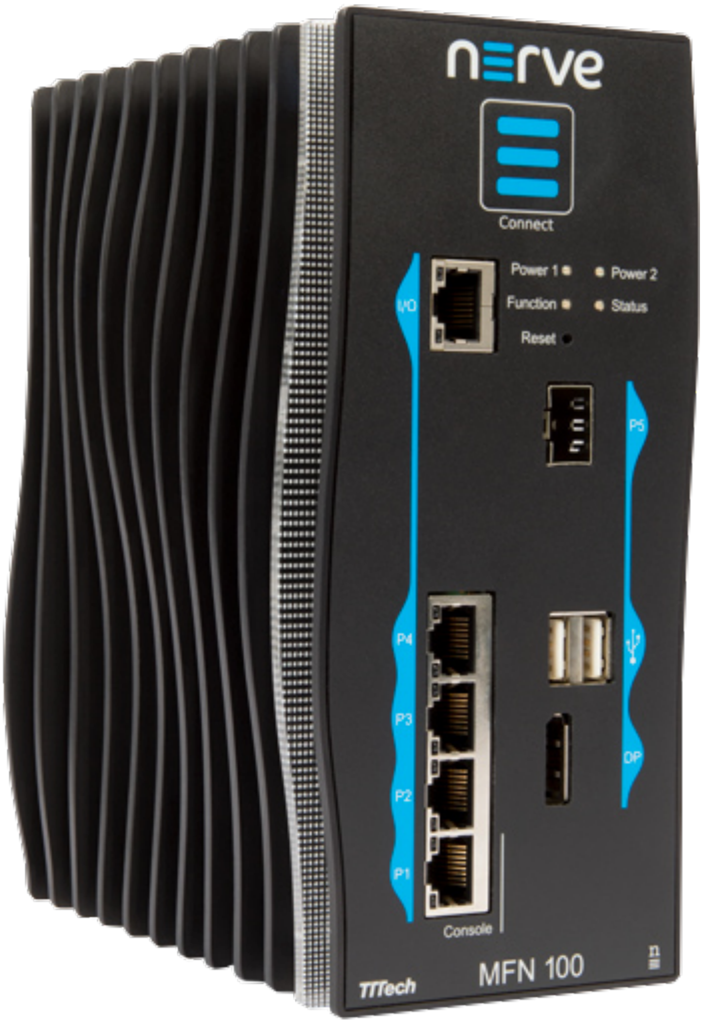
- Links to documentation material from the manufacturer
- Hardware setup for getting the device Nerve Blue ready
- A guide for installing Nerve Blue on the device
- First steps after the installation
- Overview of physical ports and node internal networking

All devices share the base functionality of Nerve Blue but differ in the extent of functionality.

NOTE

The device guide chapter for the MFN 100 contains more information as it is the flagship device for Nerve Blue. Nerve Blue can be used to its full extent when operating on the MFN 100.

MFN 100



The MFN 100 is a qualified Nerve Device that is optimized and tested for use with Nerve software. The device is designed for use in harsh industrial environments (-40°C to +70°C). It is based on an Intel Atom x5-E3940/50 CPU and offers 4 GB/8 GB RAM and up to 512 GB SSD storage. The MFN 100 offers one I/O port for Ethernet-based fieldbus connectivity, four GbE switch ports and one SFP port. Additional interfaces include two USB 2.0 ports and one Display Port.

Technical data

CPU	**Intel E3940** 4 cores, 1.8 GHz, 4 GB RAM **Intel E3950** 4 cores, 2.0 GHz, 8 GB RAM
Storage	64 GB SSD MLC 256 GB SSD MLC 512 GB SSD MLC
Performance	1 ms control cycle time achievable with Nerve Blue

Interfaces	• 4 x< RJ 45 Ethernet (1000/100/10 Mbit/s) • 1 x SFP (1000 Mbit/s) Optical transceivers / OFCS modules may be used which are in compliance with Class I device acc. 21 CFR 1040 and IEC/EN 60825-1 • 1 x DP++ • 2 x USB 2.0 1 A combined current
Mounting	DIN rail or wall mount
Dimensions	(h x w x d): 179 x 87 x 143 mm
Weight	2.1 kg
Power	2 x 24 V redundant input, Average power consumption 12 W
Environmental Parameters	• Operating Temperature Range: -40°C to 70°C • Shock and Vibration: ISO 60068-2-27: 15 g peak, 11 ms ISO 60068-2-6: 5 Hz <= f < 8.4 Hz: 3.5 mm, 8.4 Hz <= f <= 150 Hz: 1.0 g • IP 40 according to IEC 60529 • Indoor use only, intended for use in control cabinets • Use up to pollution degree 3 • Use only in environments where no condensation will occur • Maximum relative humidity: 80% for temperatures up to 31°C decreasing linearly to 50% relative humidity at 40°C • Maximum altitude: 2000 msl
Certificates	CE and UL certified (EN 61000-6-2/4, IEC/UL 61010, CSA C22.2 NO. 61010-1-12)

Identifying the MFN 100

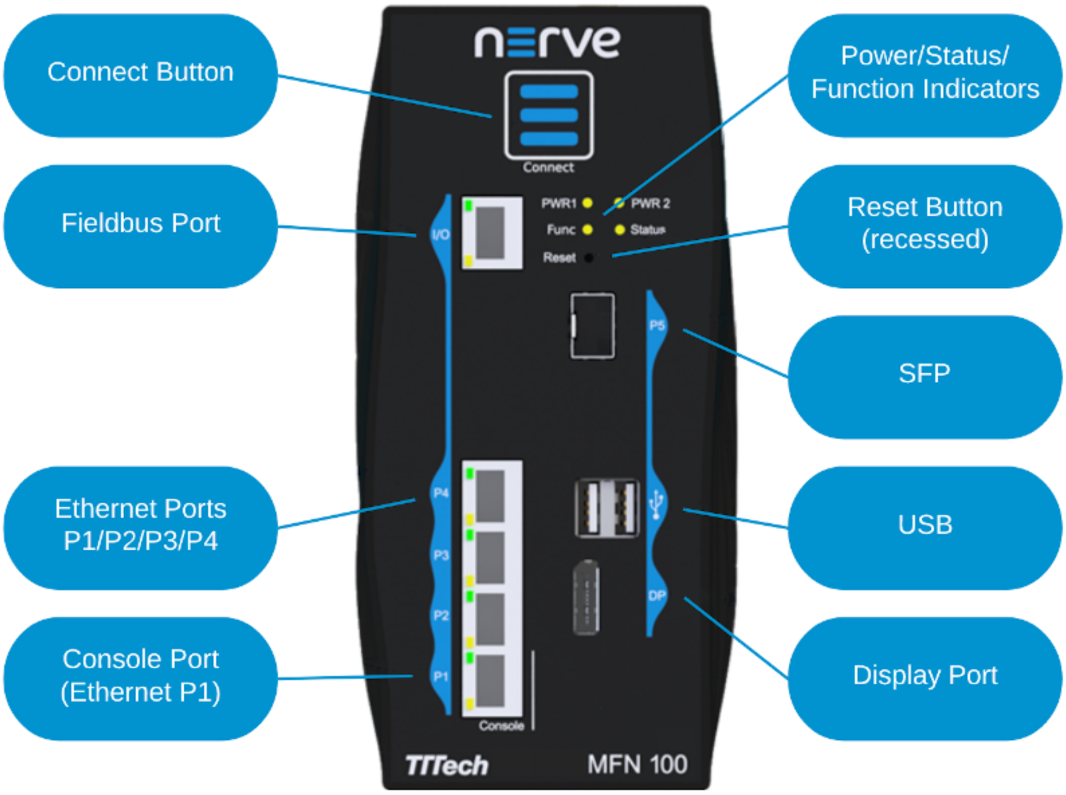
The label of the MFN 100 can be found on the back of the device, close to the DIN rail clip. Exact identification is possible through the combination of product number (P/N), serial number (S/N) and version number (V/N) that are printed on the label. The model number of the MFN 100 details the variant of the MFN 100:

Letter or Number	Description
CODESYS indicator	This letter indicates whether the device has a CODESYS runtime pre-configured: • **C** — The CODESYS runtime is pre-configured • **X** — The CODESYS runtime is not pre-configured
SSD size	This number indicates the size of the SSD: • **6** — 64 GB SSD • **2** — 256 GB SSD • **5** — 512 GB SSD

Letter or Number	Description
CPU variant	This indicates the CPU variant of the device: • **4** — Intel E3940 (4 GB RAM) • **5** — Intel E3950 (8 GB RAM)

Front panel controls and indicators

Below is an overview of the front panel of the MFN 100, describing physical interfaces, indicators and their labels.

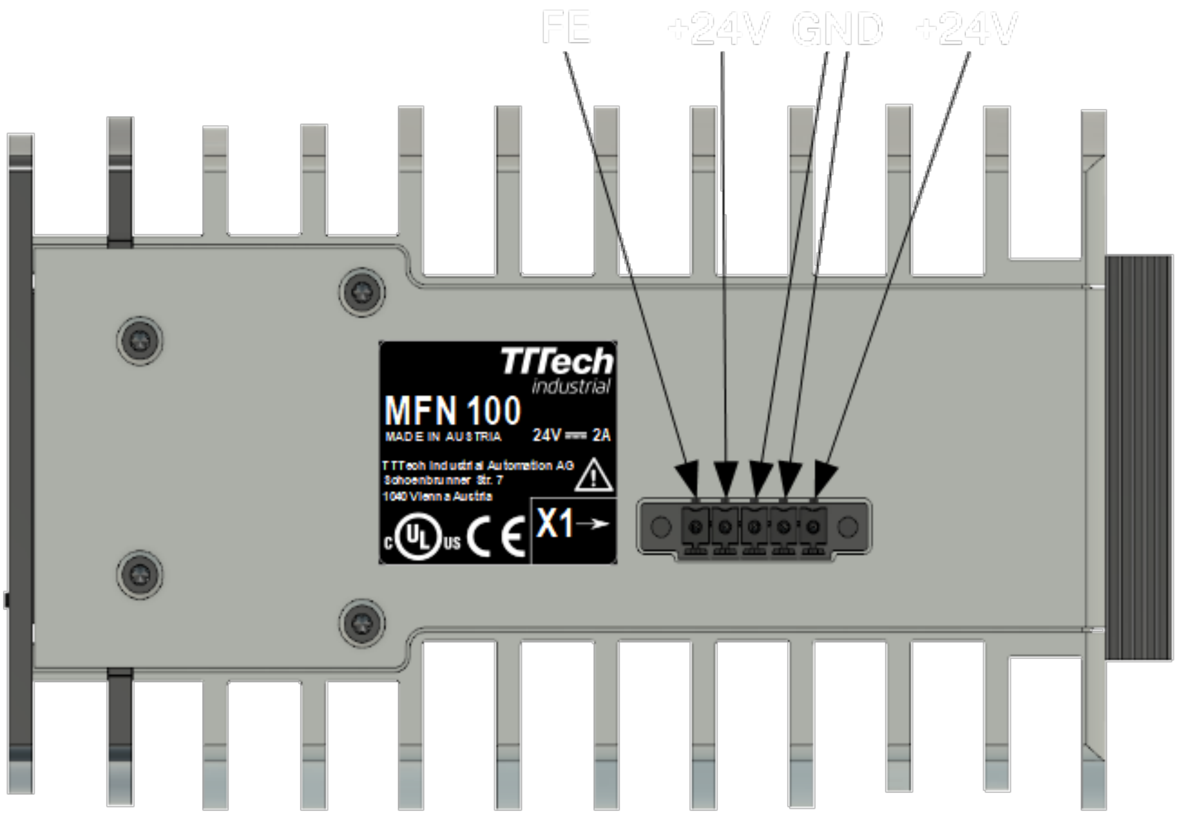


Label	Description
Connect Button	The connect button interrupts the connection on ports P2 to P5 of the MFN 100. This is the behavior in the standard configuration. The function is configurable on request. The button may be configured to change the network configuration.
Connection Indicator	The connection indicator is the first fin in the MFN 100 housing. It lights up blue when all required services are initiated and the connection to the Management System is configured.
Reset	Holding the button for 4-8 seconds initiates a power cycle. Use a tool with a rounded tip to press the button.
Power 1 **Power 2**	Indicators showing power active on the power supply.

Label	Description
Status	LED indicating system status • Green: All device functions are ready. • Not lit: Device functions are not ready or the device is booting.
Function	LED indicating CODESYS runtime status • Green: CODESYS runtime is operational. • Not lit: CODESYS runtime is not operational.
P1 Console	Ethernet port/console port. This port is typically used to connect a workstation to configure the MFN 100.
P2/P3/P4	Ethernet ports
P5	SFP port
I/O	Fieldbus interface
USB	Two USB 2.0 ports with 1.1 A maximum output current for both ports combined.
DP	Display Port supporting the DP++ standard.

Power connectors overview

The power connectors are located at the bottom of the MFN 100 next to the label. There are two separate 24 V inputs, two GND inputs and one Functional Earth (FE) input. The inputs are fused internally. The fuse cannot be replaced by the user. The power supply inputs are protected against reverse polarity.



Pin	Description
1	Power supply line 2
2	GND
3	GND
4	Power supply line 1
5	Functional Earth (FE)

NOTE

The GND and FE pins (pins 2, 3, and 5) are electrically connected to the housing.

Power supply details

Parameter	Value
Operating voltage	18 - 30 VDC
Start-up current	7 A max.
Consumption	1.4 A continuous 2.1 A peak
Dissipated power	33.6 W at 24 VDC

Installation and removal on a DIN rail

The MFN 100 is intended for mounting on a DIN rail inside a closed cabinet. Due to its weight it should be installed on a strong DIN rail. No tool is required to install or remove the MFN 100.

Follow these steps to install the MFN 100 on a DIN rail:

1. Engage the DIN rail mounting clip of the MFN 100 with the upper edge of the DIN rail.
2. Push the MFN 100 down into the DIN rail.
3. Place the MFN 100 in a vertical position so that the mounting clip engages the lower edge of the DIN rail.

Follow these steps to remove the MFN 100 from a DIN rail:

1. Push the MFN 100 down.
2. Rotate the MFN 100 upwards so that the lower edge of the DIN rail disengages.
3. Lift the MFN 100 slightly to remove it.

Setting up the MFN 100

When delivered, Nerve Blue is already installed on the MFN 100. Two network cables and a +24 V DC power supply are required to finish the setup and use Nerve Blue on the MFN 100. This includes connecting the power supply to the mating connector which is delivered with the MFN 100.

1. Connect pin 1 of the mating connector to +24 V DC.
2. Connect pin 2 of the mating connector to GND.
3. Plug the mating connector into the bottom side of the MFN 100.

4. Connect port 2 of the MFN 100 to a DHCP-enabled network.

> **NOTE**
>
> Port 2 is used for communication with the Management System. Make sure to connect the MFN 100 to the correct network, depending on whether the Management System is hosted on premise or by TTTech Industrial.

5. Plug in the power supply.

The MFN 100 will start after a few minutes and light up blue when all necessary services are initiated.

> **NOTE**
>
> - Contact the IT administrator for help on how to allow external devices to connect to the network.
> - To connect the MFN 100 to a fieldbus, connect a network cable to the I/O port of the MFN 100 and to a fieldbus interface.
> - A second power supply can also be connected to the MFN 100 as a backup. To do so, connect pin 3 of the mating connector to GND and connect pin 4 of the mating connector to +24 V DC.

Activating the Nerve Blue license

The product license needs to be activated so that Nerve Blue can be used on the device. Connect a workstation to port **P1** and configure the network adapter of the workstation. The IP address has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask.

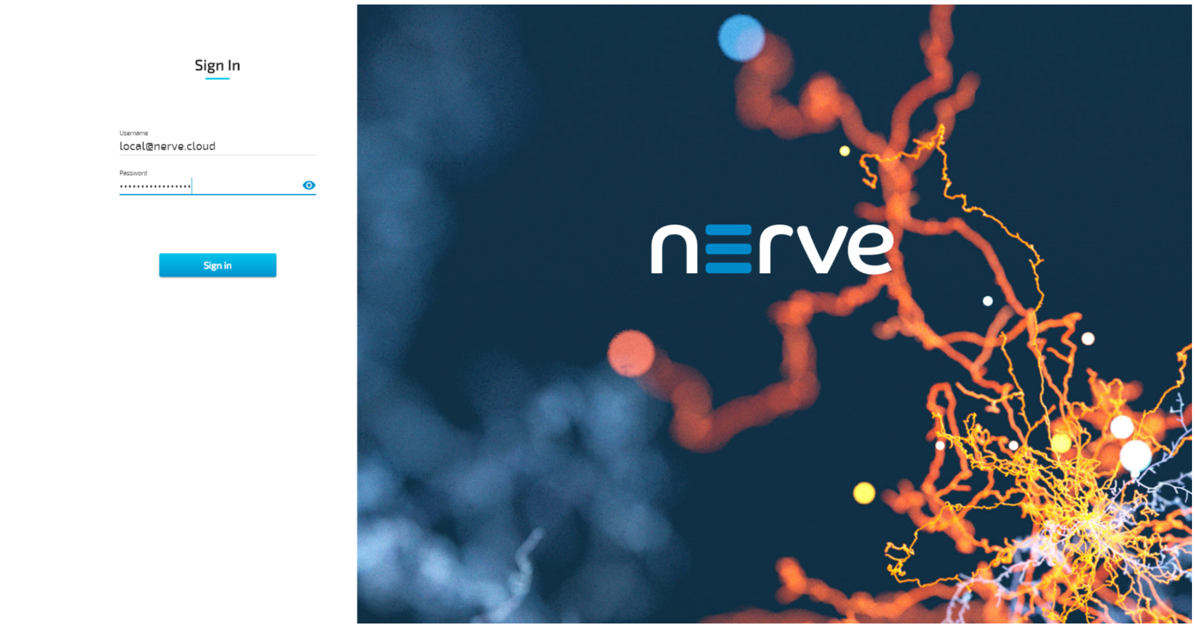
Access the license activation UI at http://172.20.2.1:3333/ and refer to License activation in the user guide for more information.

> **NOTE**
>
> It takes around 30 seconds until the Local UI can be accessed after license activation. Refresh the page until the Local UI login page is displayed.

Accessing the Local UI and registering the device

With the license activated, the node needs to be registered for use in the Management System through the Local UI. To access the Local UI, first connect a workstation to port **P1** and configure the network adapter of the workstation. The IP address has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask. The credentials for the Local UI found in the customer profile are also required.

1. Follow this link to connect to the Local UI: http://172.20.2.1:3333/
2. Log in with the credentials from the customer profile.



Sign In

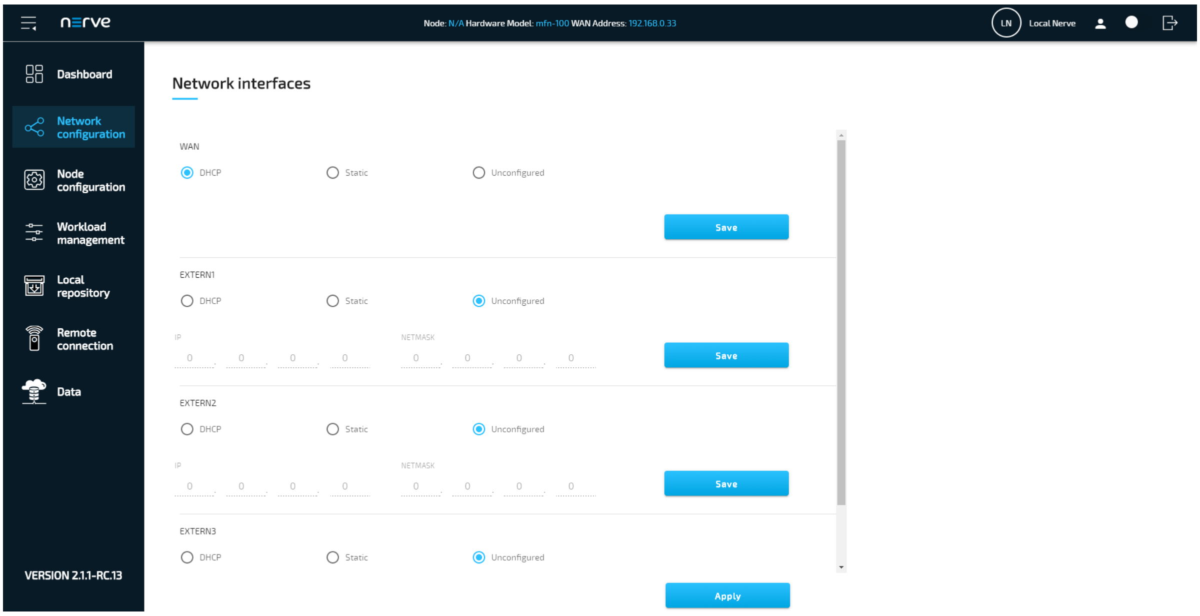
Username: local@nerve.cloud

Password: ••••••••••••••••••

Continue with Node configuration for information on how to start registering the device in the Management System.

Network configuration

The Ethernet ports of the Nerve Devices can be configured from the Local UI. For the MFN 100, the interfaces in the Local UI represent the physical ports 2, 3, 4 and 5. The console port **P1** and the I/O port of the MFN 100 are reserved and cannot be modified. The console port is used solely for configuration purposes. The I/O port is connected to the CODESYS runtime and used for fieldbus communication. Select **Network configuration** in the navigation on the left to reach this menu.



Node: N/A Hardware Model: mfn-100 WAN Address: 192.168.0.33

VERSION 2.1.1-RC.13

Network interfaces

WAN

● DHCP ○ Static ○ Unconfigured

EXTERN1

○ DHCP ○ Static ● Unconfigured

IP 0 . 0 . 0 . 0 NETMASK 0 . 0 . 0 . 0

EXTERN2

○ DHCP ○ Static ● Unconfigured

IP 0 . 0 . 0 . 0 NETMASK 0 . 0 . 0 . 0

EXTERN3

○ DHCP ○ Static ● Unconfigured

CODESYS related information

For working with the CODESYS Development System, a device description for Nerve Devices is required. The device description can be downloaded from the Nerve Software Center.

The MFN 100 has an Ethernet port that is reserved for machine data acquisition. Connect a network cable to the I/O port of the MFN 100 and to a fieldbus interface to acquire machine data. The CODESYS runtime can be reached at `172.20.2.2`.

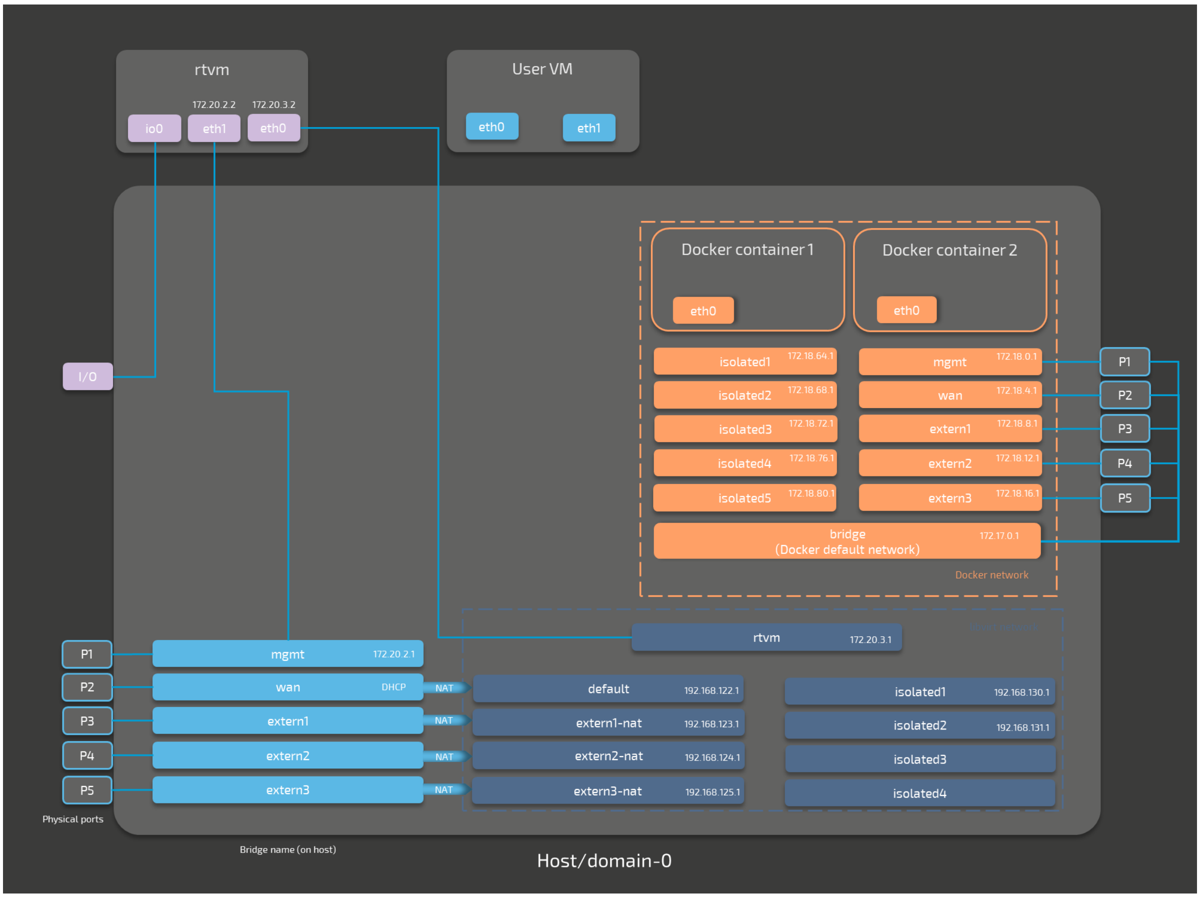
Physical ports and network interfaces

Below is a depiction of the node internal networking for the MFN 100. Refer to Node internal networking for more information. The table offers a quick overview of the network interfaces that can be reached through the physical ports of the MFN 100.

Physical port	Network name
I/O	io0
P1	mgmt
P2	wan
P3	extern1
P4	extern2
P5	extern3

Below is a graphic that details the available interfaces of the MFN 100 for use with Nerve Blue. Pictured is how the physical interfaces translate to the Host and the CODESYS runtime.

The image shows an example node and how the physical interfaces translate to the Host and the CODESYS runtime. The node consists of the **host/domain-0** and the real-time VM running the CODESYS runtime (labeled **rtvm**). It also has one Virtual Machine workload and two Docker workloads deployed. The virtual machine is located outside of the host and the Docker containers are located in the Docker network inside of the host. However, the workloads are not yet connected.



Notable IP Adresses	
Host access	`172.20.2.1`
CODESYS runtime access	`172.20.2.2`

Refer to Node internal networking for more information on networking in the Nerve Blue system.

Updating Nerve Blue from version 2.0 to 2.1

As updating nodes to newer versions through the Management System is a feature introduced in version 2.1, updates from version 2.0 to version 2.1 have to be performed manually. However, before updating the node, contact the sales representative or customer support for information on how to backup current data.

Requirements for updating Nerve Blue on the MFN 100:

- a monitor with a display port input
- a keyboard
- a USB drive
- the `Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img.tar.gz` which can be downloaded from the Nerve Software Center
- a tool for creating bootable USB drives like Rufus on Windows

In addition, a workstation is required to prepare the bootable USB drive. Connect the monitor and the keyboard to the MFN 100.

Before beginning with the installation, make sure that the device will boot from the USB drive. Press F7 when the device is booting to enter the boot menu.

On Windows

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img.tar.gz` from the Nerve Software Center to a workstation.
2. Extract the `Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img.tar.gz` file to retrieve the `Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img`. Depending on the program used, the file might need to be extracted more than once.
3. Transfer the extracted `Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img` file to the USB drive using Rufus.
4. Plug the USB drive into a USB port of the Nerve Device.
5. Power on the device.
6. Press F7 to enter the boot menu. Make sure that the device will boot from the USB drive.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

On Linux

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img.tar.gz` file from the Nerve Software Center.

2. Enter the following commands to extract the `Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img.tar.gz` file and transfer the extracted file to the USB drive:

   ```
   tar xf Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img.tar.gz
   sudo dd if=Nerve_Blue_USB-installer_2.1.2_for_mfn-100.img bs=4M of=/dev/sd<drivena
   sync
   ```

   NOTE

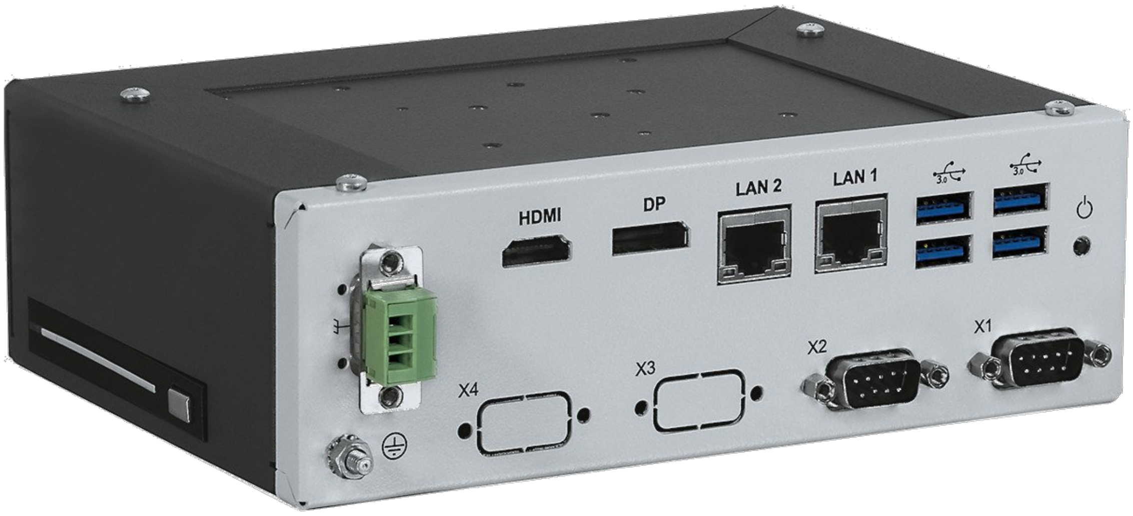
   Make sure to replace `<drivename>` with the system name of the USB drive.

3. Plug the USB drive into a USB port of the Nerve Device.

4. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

Kontron KBox A-150-APL



The KBox A-150-APL is an industrial computer platform for process control and optimization with the Intel Atom series processors. It offers DIN Rail mounting positions in limited space.

For more information refer to the information materials provided by the manufacturer:

- Product page
- User manual

Device specifications

The table below contains the key specifications of the specific hardware model that has been certified for Nerve Blue usage. Use the article number listed here when ordering the device from the manufacturer only. Note that other device variants are not supported as Nerve Devices.

If required, contact sales@tttech-industrial.com for help with ordering Nerve Devices.

Item	Description
Article number	EN00-03002-01
CPU	Intel Atom E3950
Cores	4
RAM	4 GB DDR3
Storage	128 GB 2.5" SATA SSD
TPM	TPM 2.0 included
Interfaces	• 2x GB LAN • 2x RS232/422/485 • 4x USB • 1x Display Port • 1x HDMI

Setting up the device for Nerve Blue usage

Requirements for the instructions below:

- a monitor with a display port input
- a keyboard
- a USB drive formatted to FAT32

Refer to the user manual of the manufacturer to set up the hardware. Connect a keyboard and a monitor to the device and make sure that the device is ready to be powered on. Also, prepare a USB drive in case the BIOS version of the device needs to be updated.

BIOS update

To avoid possible issues and complications, update the BIOS on the KBox A-150-APL to the latest version. Refer to the user manual of the manufacturer on information how to update the BIOS version.

Required BIOS settings for Nerve Blue

Certain BIOS settings need to be changed to ensure the desired performance of the Nerve Blue system.

1. Power on the device.
2. Press Del while the device is booting to enter the BIOS menu.

3. Change the following settings:

Path	Setting
Advanced > CPU Chipset Configuration > EIST	**Disabled**
Advanced > CPU Chipset Configuration > Active Processor Cores	**Enabled**
Advanced > CPU Chipset Configuration > Intel Virtualization Technology	**Enabled**
Advanced > CPU Chipset Configuration > VT-d	**Enabled**
Advanced > CPU Chipset Configuration > C-States	**Disabled**
Advanced > Network Stack > Network Stack	**Enabled**
Security > Secure Boot > Attempt Secure Boot	**Disabled**

4. Save the changes and exit BIOS.

Installing Nerve Blue

Requirements for installing Nerve Blue on the device:

- a monitor with a display port input
- a keyboard
- a USB drive
- the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img.tar.gz` which can be downloaded from the Nerve Software Center
- a tool for creating bootable USB drives like Rufus on Windows

In addition, a workstation is required to prepare the bootable USB drive. Connect the monitor and the keyboard to the KBox A-150-APL.

Before beginning with the installation, make sure that the device will boot from the USB drive. Press Del when the device is booting to enter BIOS and change the boot device settings.

On Windows

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img.tar.gz` from the Nerve Software Center to a workstation.
2. Extract the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img.tar.gz` file to retrieve the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img` file. Depending on the program used, the file might need to be extracted more than once.
3. Transfer the extracted `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img` file to the USB drive using Rufus.
4. Plug the USB drive into a USB port of the Nerve Device.
5. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

On Linux

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img.tar.gz` file from the Nerve Software Center.

2. Enter the following commands to extract the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img.tar.gz` file and transfer the extracted file to the USB drive:

```
tar xf Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img.tar.gz
sudo dd if=Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-150-apl.img bs=4M of=
sync
```

NOTE

Make sure to replace `<drivename>` with the system name of the USB drive.

3. Plug the USB drive into a USB port of the Nerve Device.

4. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

Finding out the IP address of the device

Due to the limited availability of ethernet ports, Nerve Blue does not offer a designated port and interface for host access and management purposes on the Kontron KBox A-150-APL. The IP address of the **wan** interface that is mapped to physical port **LAN 1** is required to start using Nerve Blue. Depending on the network access the node has, this needs to be done differently.

The node has network access

If the node has network access, an IP address will be assigned to the **wan** interface by a DHCP server. Follow the instructions below to find out the IP address of the **wan** interface.

1. Connect a keyboard and a monitor to the device.
2. Power up the device once Nerve Blue is installed.
3. Enter the login credentials for host access once the system asks for host login. The login credentials can be found in the customer profile.
4. Enter the following command to display the IP address of the **wan** interface:

   ```
   ip a s wan
   ```

The IP address is displayed next to **inet** in the output the system gives. This IP address is required to access the license activation UI and the Local UI in the instructions below.

The node does not have network access

In case of the node not having network access, the IP address of the **wan** interface has to be set manually. For simplicity, the IP address of the **wan** interface will be set to `172.20.2.1` — the IP address of the host.

1. Connect a keyboard and a monitor to the device.
2. Power up the device once Nerve Blue is installed.
3. Enter the login credentials for host access once the system asks for host login. The login credentials can be found in the customer profile.
4. Enter the following commands to open the **wan** interface configuration:

   ```
   cd /etc/network/interfaces.d
   sudo nano wan
   ```

5. Enter the host access password if prompted.
6. Edit the configuration the following way:

   ```
   auto wan
   iface wan inet static
     bridge_ports eth0
     address 172.20.2.1
     netmask 255.255.255.0
   ```

7. Enter Ctrl+S to save the configuration.
8. Enter Ctrl+X to exit the Nano editor.
9. Enter the following command to apply the changes to the **wan** interface by restarting the networking services:

   ```
   /etc/init.d/networking restart
   ```

10. Enter the host access password if prompted.

With the **wan** interface IP address set to `172.20.2.1` the license activation UI can be reached at http://172.20.2.1:3333/.

Activating the Nerve Blue license

After the installation, the product license needs to be activated so that Nerve Blue can be used on the device. Connect a workstation to port **LAN 1** and configure the network adapter of the workstation. The IP address has to be in the same range as the IP address of the **wan** interface with a `255.255.255.0` subnet mask. Refer to the chapter above on how to find out this IP address.

Access the license activation UI at `<wanip>:3333` in a web browser and refer to License activation in the user guide for more information.

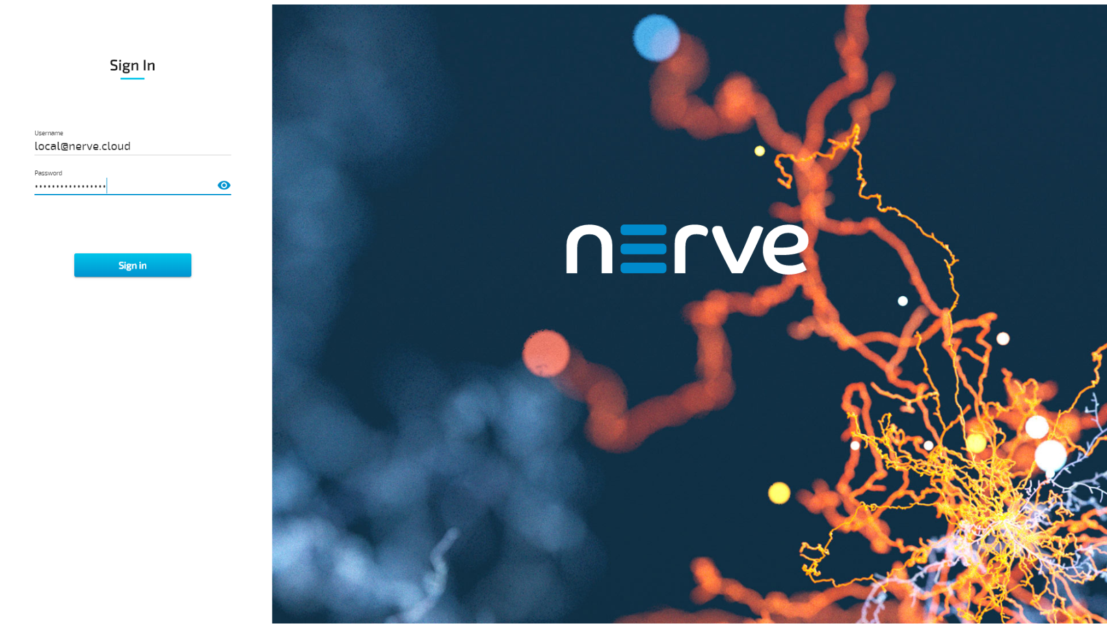
> **NOTE**
>
> It takes around 30 seconds until the Local UI can be accessed after license activation. Refresh the page until the Local UI login page is displayed.

Accessing the Local UI and registering the device

With the license activated, the node needs to be registered for use in the Management System through the Local UI. To access the Local UI, first connect a workstation to port **LAN 1** and configure the network adapter of the workstation. The IP address has to be in the same range as the IP address of the **wan** interface with a `255.255.255.0` subnet mask. Refer to the chapter above on how to find out this IP address. The credentials for the Local UI found in the customer profile are also required.

1. Access the Local UI at `<wanip>:3333` in a web browser.

2. Log in with the credentials from the customer profile.



Username: local@nerve.cloud

Password: ••••••••••••••••

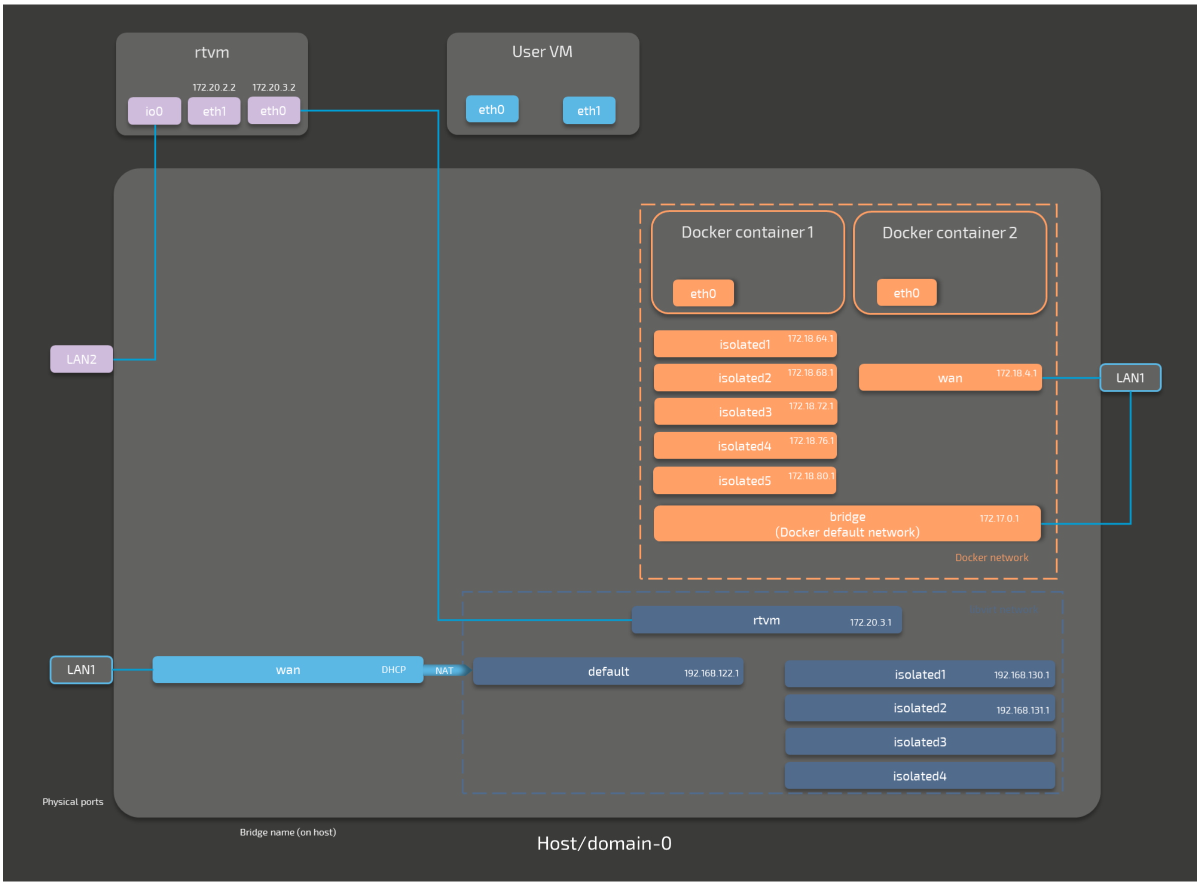
Continue with Node configuration for information on how to start registering the device in the Management System.

Physical ports and network interfaces

Below is a depiction of the node internal networking adapted to the KBox A-150-APL hardware. Refer to Node internal networking for more information. The table offers a quick overview of the network interfaces that can be reached through the physical ports of the KBox A-150-APL.

Physical port	Network name
LAN 1	wan

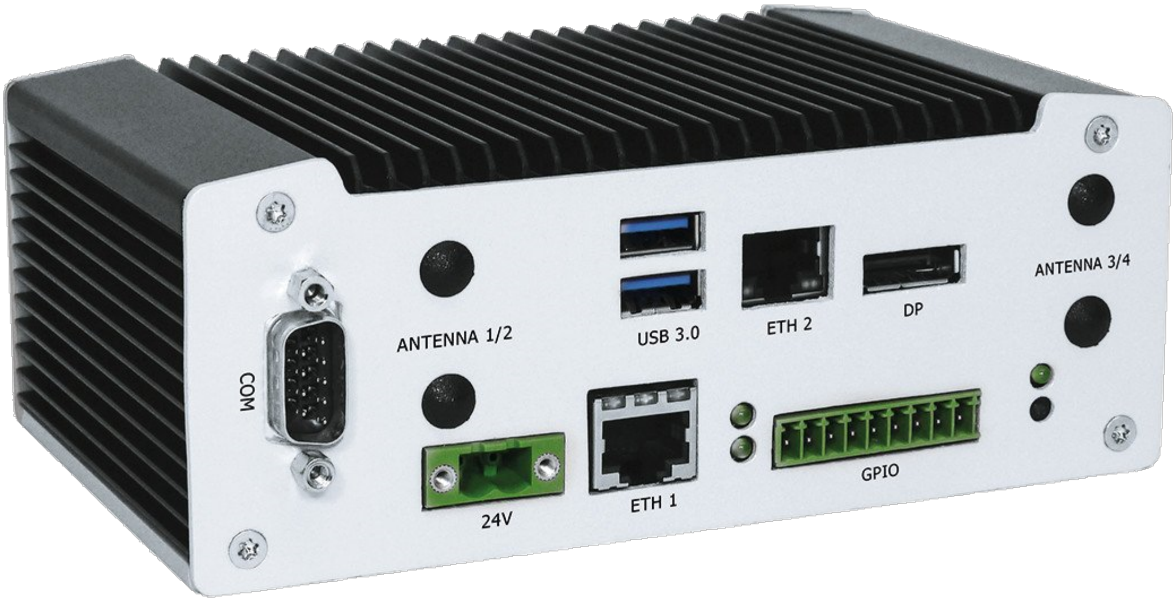
Physical port	Network name
LAN 2	io0



Known issues and limitations

Development of CODESYS applications is not possible with the KBox A-150-APL due to the number of physical ports. The `mgmt` interface is not accessible and can therefore not be used by the CODESYS Development System.

Kontron KBox A-250



Based on a pITX-2.5“ SBC with Intel Atom processors of the E3900 family, the fanless KBox A-250 has been designed as a gateway for IoT-Edge applications. Its fields of application are primarily found in industrial automation.

For more information refer to the information materials provided by the manufacturer:

- Product page
- User manual

Device specifications

The table below contains the key specifications of the specific hardware model that has been certified for Nerve Blue usage. Use the article number listed here when ordering the device from the manufacturer only. Note that other device variants are not supported as Nerve Devices.

If required, contact sales@tttech-industrial.com for help with ordering Nerve Devices.

Item	Description
Article number	2-A0DM-009
CPU	Intel Atom E3950
Cores	4
RAM	4 GB DDR3L
Storage	128 GB M.2 MLC
TPM	TPM 2.0 included
Interfaces	• 2x GB LAN • 1x RS232/422/485 • 2x USB • 1x Display Port

Setting up the device for Nerve Blue usage

Requirements for the instructions below:

- a monitor with a display port input
- a keyboard
- a USB drive formatted to FAT32

Refer to the user manual of the manufacturer to set up the hardware. Connect a keyboard and a monitor to the device and make sure that the device is ready to be powered on. Also, prepare a USB drive in case the BIOS version of the device needs to be updated.

BIOS update

To avoid possible issues and complications, update the BIOS on the KBox A-250 to the latest version. Refer to the user manual of the manufacturer on information how to update the BIOS version.

Required BIOS settings for Nerve Blue

Certain BIOS settings need to be changed to ensure the desired performance of the Nerve Blue system.

1. Power on the device.
2. Press Del while the device is booting to enter the BIOS menu.

3. Change the following settings:

Path	Setting
Advanced > ACPI Settings > Enable ACPI Auto Configuration	**Disabled**
Advanced > ACPI Settings > Enable Hibernation	**Disabled**
Advanced > ACPI Settings > ACPI Sleep State	**Suspend Disabled**
Advanced > CPU Configuration > Turbo Mode	**Disabled**
Advanced > CPU Configuration > Intel Virtualization Technology	**Enabled**
Advanced > CPU Configuration > VT-d	**Enabled**
Advanced > Network Stack Configuration > Network Stack	**Enabled**
Advanced > CSM Configuration > CSM Support	**Disabled**
Advanced > System Component > DDR SSC	**Disable**
Advanced > System Component > HighSpeed SerialIO SSC	**Disable**
Chipset > South Bridge > OS Selection	**Intel Linux**
Chipset > South Bridge > Real Time Option	**RT Enabled, Agent Disabled**
Security > Secure Boot > Attempt Secure Boot	**Disabled**

4. Save the changes and exit BIOS.

Installing Nerve Blue

Requirements for installing Nerve Blue on the device:

- a monitor with a display port input
- a keyboard
- a USB drive
- the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img.tar.gz` which can be downloaded from the Nerve Software Center
- a tool for creating bootable USB drives like Rufus on Windows

In addition, a workstation is required to prepare the bootable USB drive. Connect the monitor and the keyboard to the KBox A-250.

Before beginning with the installation, make sure that the device will boot from the USB drive. Press Del when the device is booting to enter BIOS and change the boot device settings.

On Windows

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img.tar.gz` from the Nerve Software Center to a workstation.
2. Extract the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img.tar.gz` file to retrieve the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img` file. Depending on the program used, the file might need to be extracted more than once.
3. Transfer the extracted `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img` file to the USB drive using Rufus.
4. Plug the USB drive into a USB port of the Nerve Device.
5. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

On Linux

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img.tar.gz` file from the Nerve Software Center.

2. Enter the following commands to extract the `Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img.tar.gz` file and transfer the extracted file to the USB drive:

```
tar xf Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img.tar.gz
sudo dd if=Nerve_Blue_USB-installer_2.1.2_for_kontron-kbox-a-250.img bs=4M of=/dev
sync
```

> **NOTE**
>
> Make sure to replace `<drivename>` with the system name of the USB drive.

3. Plug the USB drive into a USB port of the Nerve Device.

4. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

Finding out the IP address of the device

Due to the limited availability of ethernet ports, Nerve Blue does not offer a designated port and interface for host access and management purposes on the Kontron KBox A-250. The IP address of the **wan** interface that is mapped to physical port **ETH 2** is required to start using Nerve Blue. Depending on the network access the node has, this needs to be done differently.

The node has network access

If the node has network access, an IP address will be assigned to the **wan** interface by a DHCP server. Follow the instructions below to find out the IP address of the **wan** interface.

1. Connect a keyboard and a monitor to the device.
2. Power up the device once Nerve Blue is installed.
3. Enter the login credentials for host access once the system asks for host login. The login credentials can be found in the customer profile.

4. Enter the following command to display the IP address of the **wan** interface:

   ```
   ip a s wan
   ```

The IP address is displayed next to **inet** in the output the system gives. This IP address is required to access the license activation UI and the Local UI in the instructions below.

The node does not have network access

In case of the node not having network access, the IP address of the **wan** interface has to be set manually. For simplicity, the IP address of the **wan** interface will be set to `172.20.2.1` — the IP address of the host.

1. Connect a keyboard and a monitor to the device.
2. Power up the device once Nerve Blue is installed.
3. Enter the login credentials for host access once the system asks for host login. The login credentials can be found in the customer profile.

4. Enter the following commands to open the **wan** interface configuration:

   ```
   cd /etc/network/interfaces.d
   sudo nano wan
   ```

5. Enter the host access password if prompted.

6. Edit the configuration the following way:

   ```
   auto wan
   iface wan inet static
     bridge_ports eth0
     address 172.20.2.1
     netmask 255.255.255.0
   ```

7. Enter Ctrl+S to save the configuration.

8. Enter Ctrl+X to exit the Nano editor.

9. Enter the following command to apply the changes to the **wan** interface by restarting the networking services:

   ```
   /etc/init.d/networking restart
   ```

10. Enter the host access password if prompted.

With the **wan** interface IP address set to `172.20.2.1` the license activation UI can be reached at http://172.20.2.1:3333/.

Activating the Nerve Blue license

After the installation, the product license needs to be activated so that Nerve Blue can be used on the device. Connect a workstation to port **ETH 2** and configure the network adapter of the workstation. The IP address has to be in the same range as the IP address of the **wan** interface with a `255.255.255.0` subnet mask. Refer to the chapter above on how to find out this IP address.

Access the license activation UI at `<wanip>:3333` in a web browser and refer to License activation in the user guide for more information.

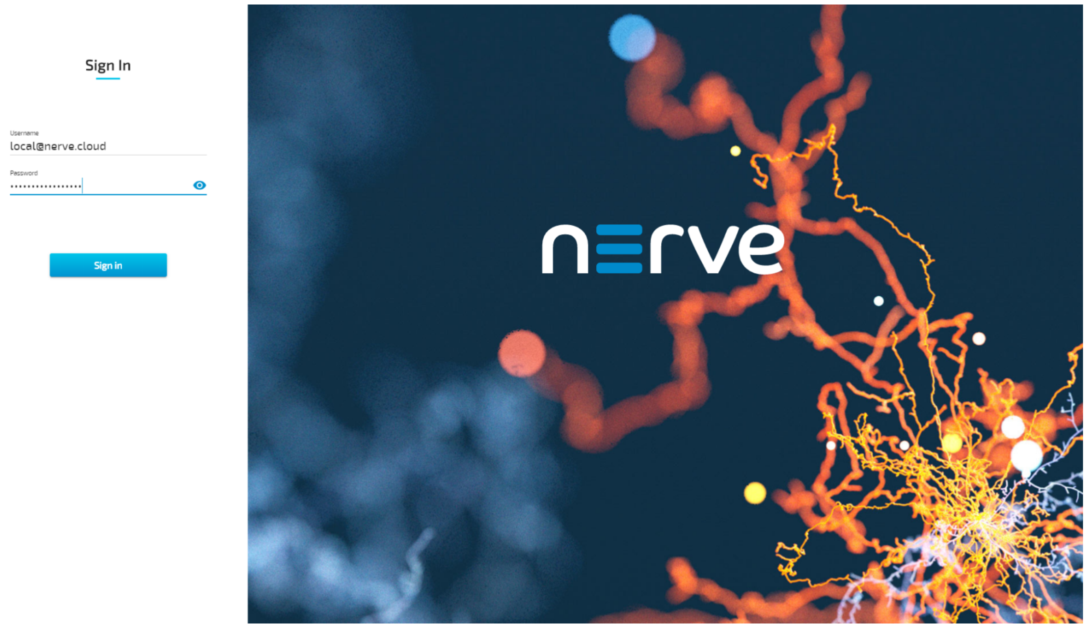
NOTE

It takes around 30 seconds until the Local UI can be accessed after license activation. Refresh the page until the Local UI login page is displayed.

Accessing the Local UI and registering the device

With the license activated, the node needs to be registered for use in the Management System through the Local UI. To access the Local UI, first connect a workstation to port **ETH 2** and configure the network adapter of the workstation. The IP address has to be in the same range as the IP address of the **wan** interface with a `255.255.255.0` subnet mask. Refer to the chapter above on how to find out this IP address. The credentials for the Local UI found in the customer profile are also required.

1. Access the Local UI at `<wanip>:3333` in a web browser.
2. Log in with the credentials from the customer profile.



Username: local@nerve.cloud

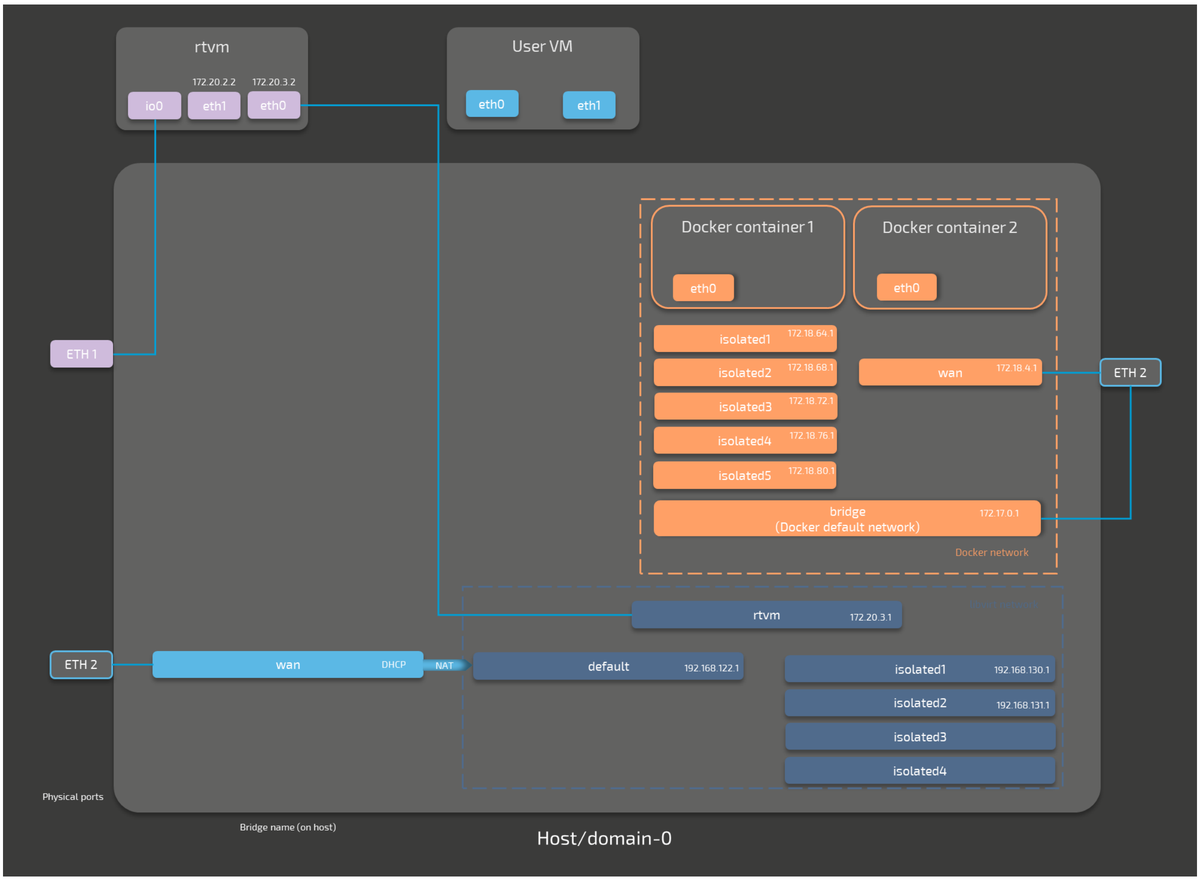
Password: ••••••••••••••••

Continue with Node configuration for information on how to start registering the device in the Management System.

Physical ports and network interfaces

Below is a depiction of the node internal networking adapted to the KBox A-250 hardware. Refer to Node internal networking for more information. The table offers a quick overview of the network interfaces that can be reached through the physical ports of the KBox A-250.

Physical port	Network name
ETH 2	wan
ETH 1	io0



Known issues and limitations

Development of CODESYS applications is not possible with the KBox A-250 due to the number of physical ports. The `mgmt` interface is not accessible and can therefore not be used by the CODESYS Development System.

Siemens SIMATIC IPC127E

The SIMATIC IPC127E can be integrated into a control cabinet or directly on the machine with minimal space requirements to record, collect, process, and transfer data directly in the production environment.

For more information refer to the information materials provided by the manufacturer:

- Product page
- User manual

Device specifications

The table below contains the key specifications of the specific hardware model that has been certified for Nerve Blue usage. Use the article number listed here when ordering the device from the manufacturer only. Note that other device variants are not supported as Nerve Devices.

If required, contact sales@tttech-industrial.com for help with ordering Nerve Devices.

Item	Description
Article number	6AG4021-0AB12-1CA0 /SIMATIC IPC127
CPU	Intel Atom E3940
Cores	4
RAM	4 GB
Storage	4128 GB
TPM	TPM 2.0 included
Interfaces	• 3x GB LAN • 4x USB 3.0 • 1x Display Port

Setting up the device for Nerve Blue usage

Requirements for the instructions below:

• a monitor with a display port input • a keyboard • a USB drive formatted to FAT32

Refer to the user manual of the manufacturer to set up the hardware. Connect a keyboard and a monitor to the device and make sure that the device is ready to be powered on. Also, prepare a USB drive in case the BIOS version of the device needs to be updated.

BIOS update

To avoid possible issues and complications, the BIOS version on the SIMATIC IPC127E needs to be version V27.01.03. or later. Refer to the user manual of the manufacturer on information how to update the BIOS version.

Required BIOS settings for Nerve Blue

Certain BIOS settings need to be changed to ensure the desired performance of the Nerve Blue system.

1. Power on the device.
2. Press Esc while the device is booting to enter the BIOS menu.

3. Change the following settings:

Path	Setting
Power > Advanced CPU Control > VT-d	**Enabled**
Power > Advanced CPU Control > Active Processor Cores	**Disabled**
Power > Advanced CPU Control > C states	**Disabled**
Power > Advanced CPU Control > User Power Scenario	**Max Performance**
Power > Advanced CPU Control > Turbo Mode	**Disabled**
Boot > Network Stack	**Enabled**
Boot > USB Boot	**Enabled**

4. Save the changes and exit BIOS.

Installing Nerve Blue

Requirements for installing Nerve Blue on the device:

- a monitor with a display port input
- a keyboard
- a USB drive
- the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img.tar.gz` which can be downloaded from the Nerve Software Center
- a tool for creating bootable USB drives like Rufus on Windows

In addition, a workstation is required to prepare the bootable USB drive. Connect the monitor and the keyboard to the SIMATIC IPC127E.

Before beginning with the installation, make sure that the device will boot from the USB drive. Press Esc when the device is booting to enter BIOS and change the boot device settings.

On Windows

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img.tar.gz` from the Nerve Software Center to a workstation.
2. Extract the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img.tar.gz` file to retrieve the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img`. Depending on the program used, the file might need to be extracted more than once.
3. Transfer the extracted `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img` file to the USB drive using Rufus.
4. Plug the USB drive into a USB port of the Nerve Device.
5. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

On Linux

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img.tar.gz` file from the Nerve Software Center.

2. Enter the following commands to extract the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img.tar.gz` file and transfer the extracted file to the USB drive:

```
tar xf Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img.tar.gz
sudo dd if=Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc127e.img bs=4M of
sync
```

NOTE

Make sure to replace `<drivename>` with the system name of the USB drive.

3. Plug the USB drive into a USB port of the Nerve Device.
4. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

Activating the Nerve Blue license

After the installation, the product license needs to be activated so that Nerve Blue can be used on the device. Connect a workstation to port **X1 P1** and configure the network adapter of the workstation. The IP address has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask.

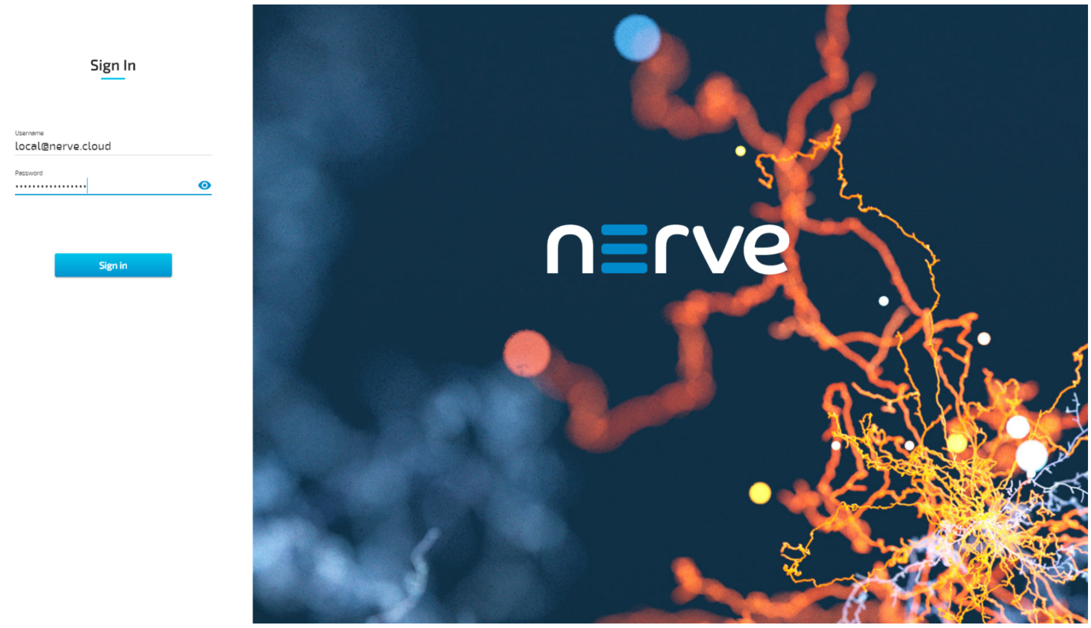
Access the license activation UI at http://172.20.2.1:3333/ and refer to License activation in the user guide for more information.

> **NOTE**
>
> It takes around 30 seconds until the Local UI can be accessed after license activation. Refresh the page until the Local UI login page is displayed.

Accessing the Local UI and registering the device

With the license activated, the node needs to be registered for use in the Management System through the Local UI. To access the Local UI, first connect a workstation to port **X1 P1** and configure the network adapter of the workstation. The IP address has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask. The credentials for the Local UI found in the customer profile are also required.

1. Follow this link to connect to the Local UI: http://172.20.2.1:3333/
2. Log in with the credentials from the customer profile.



Sign In

Username: local@nerve.cloud

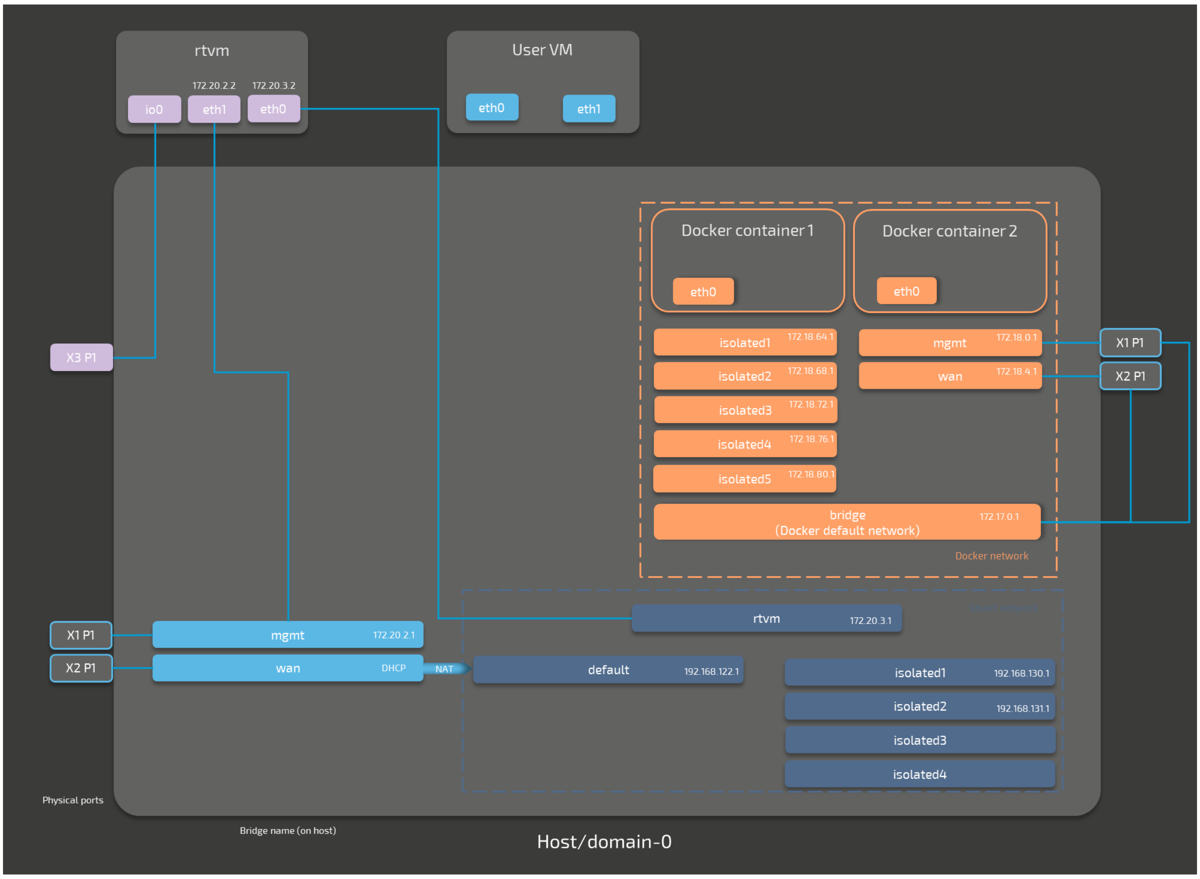
Password: ••••••••••••••••••

Continue with Node configuration for information on how to start registering the device in the Management System.

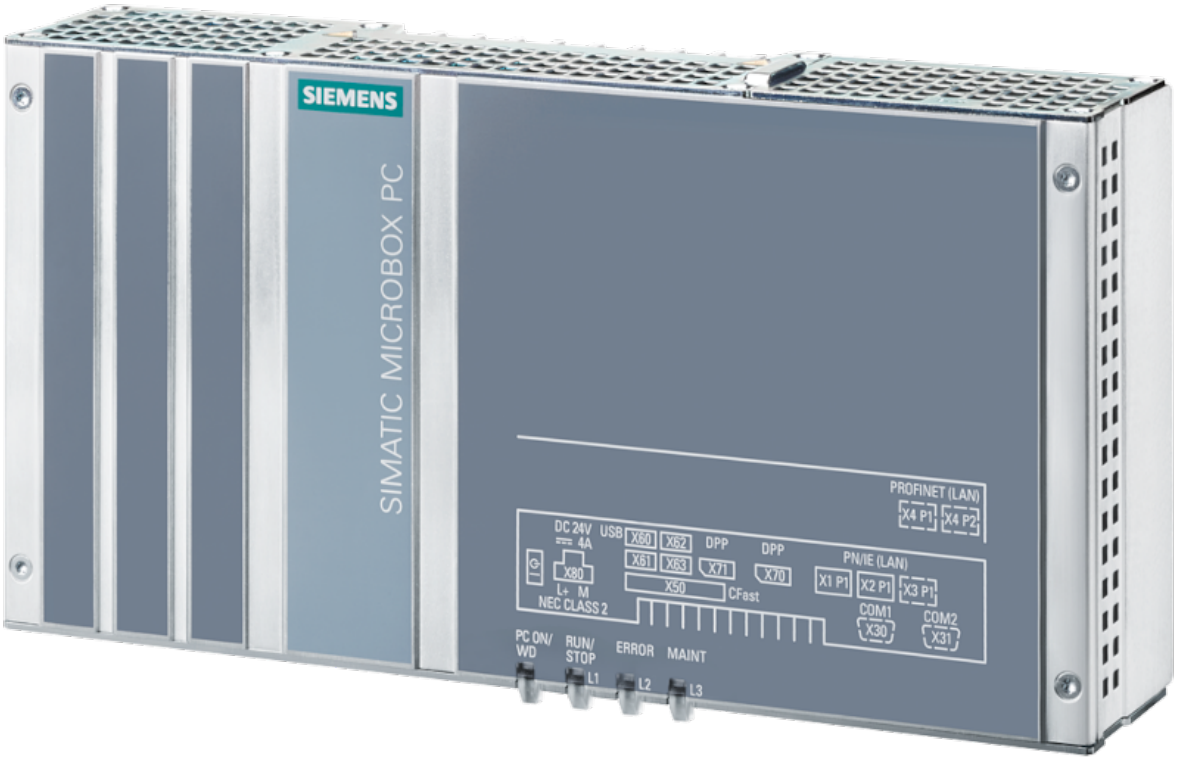
Physical ports and network interfaces

Below is a depiction of the node internal networking adapted to the SIMATIC IPC127E hardware. Refer to Node internal networking for more information. The table offers a quick overview of the network interfaces that can be reached through the physical ports of the SIMATIC IPC127E.

Physical port	Network name
X1 P1	mgmt
X2 P1	wan
X3 P1	io0



Siemens SIMATIC IPC427E



The SIMATIC IPC427E is an embedded IPC with 6th generation Intel Core-i processors that can be used for control, data collection, or communication tasks at the machine or process level. It offers low mounting depth and different mounting possibilities, with all interfaces on one side for easy integration in existing plants.

For more information refer to the information materials provided by the manufacturer:

- Product page
- User manual

Device specifications

The table below contains the key specifications of the specific hardware model that has been certified for Nerve Blue usage. Use the article number listed here when ordering the device from the manufacturer only. Note that other device variants are not supported as Nerve Devices.

If required, contact sales@tttech-industrial.com for help with ordering Nerve Devices.

Item	Description
Article number	6AG4141-5BB00-0GA8 /SIMATIC IPC427
CPU	Intel Core i5-6442EQ
Cores	4
RAM	8 GB
Storage	480 GB SATA SSD
TPM	TPM 2.0 included
Interfaces	• 3x GB LAN • 4x USB 3.0 • 2x Display Port

Setting up the device for Nerve Blue usage

Requirements for the instructions below:

- a monitor with a display port input
- a keyboard
- a USB drive formatted to FAT32

Refer to the user manual of the manufacturer to set up the hardware. Connect a keyboard and a monitor to the device and make sure that the device is ready to be powered on. Also, prepare a USB drive in case the BIOS version of the device needs to be updated.

BIOS update

To avoid possible issues and complications, the BIOS version on the SIMATIC IPC427E needs to be version 12 or later. Refer to the user manual of the manufacturer on information how to update the BIOS version.

Required BIOS settings for Nerve Blue

Certain BIOS settings need to be changed to ensure the desired performance of the Nerve Blue system.

1. Power on the device.
2. Press Esc while the device is booting to enter the BIOS menu.
3. Change the following settings:

Path	Setting
Advanced > System Agent (SA) Configuration > VT-d	**Enabled**
Advanced > Active Management Technology Support > Intel AMT Configuration Screens	**Disabled**
Advanced > Memory Configuration > Max TOLUD	**2 GB**
Power > CPU Configuration > Intel (VMX) Virtualization Technology	**Enabled**
Power > Power & Performance > CPU - Power Management Control > CPU Power Level	**Standard**
Power > Power & Performance > CPU - Power Management Control > Intel (R) SpeedStep(tm)	**Disabled**
Power > Power & Performance > CPU - Power Management Control > Intel (R) Speed Shift Technology	**Disabled**
Power > Power & Performance > CPU - Power Management Control > C states	**Disabled**
Boot > Boot Type	**Dual Boot Type**
Boot > Add Boot Options	**Auto**
Boot > USB Boot	**Enabled**
Boot > EFI Device First	**Enabled**

4. Save the changes and exit BIOS.

Installing Nerve Blue

Requirements for installing Nerve Blue on the device:

- a monitor with a display port input
- a keyboard
- a USB drive
- the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img.tar.gz` which can be downloaded from the Nerve Software Center
- a tool for creating bootable USB drives like Rufus on Windows

In addition, a workstation is required to prepare the bootable USB drive. Connect the monitor and the keyboard to the SIMATIC IPC427E.

Before beginning with the installation, make sure that the device will boot from the USB drive. Press Esc when the device is booting to enter BIOS and change the boot device settings.

On Windows

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img.tar.gz` from the Nerve Software Center to a workstation.
2. Extract the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img.tar.gz` file to retrieve the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img`. Depending on the program used, the file might need to be extracted more than once.
3. Transfer the extracted `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img` file to the USB drive using Rufus.
4. Plug the USB drive into a USB port of the Nerve Device.
5. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

On Linux

1. Download the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img.tar.gz` file from the Nerve Software Center.

2. Enter the following commands to extract the `Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img.tar.gz` file and transfer the extracted file to the USB drive:

   ```
   tar xf Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img.tar.gz
   sudo dd if=Nerve_Blue_USB-installer_2.1.2_for_siemens-simatic-ipc427e.img bs=4M of
   sync
   ```

   NOTE

   Make sure to replace `<drivename>` with the system name of the USB drive.

3. Plug the USB drive into a USB port of the Nerve Device.

4. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

Activating the Nerve Blue license

After the installation, the product license needs to be activated so that Nerve Blue can be used on the device. Connect a workstation to port **X1 P1** and configure the network adapter of the workstation. The IP address has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask.

Access the license activation UI at http://172.20.2.1:3333/ and refer to License activation in the user guide for more information.

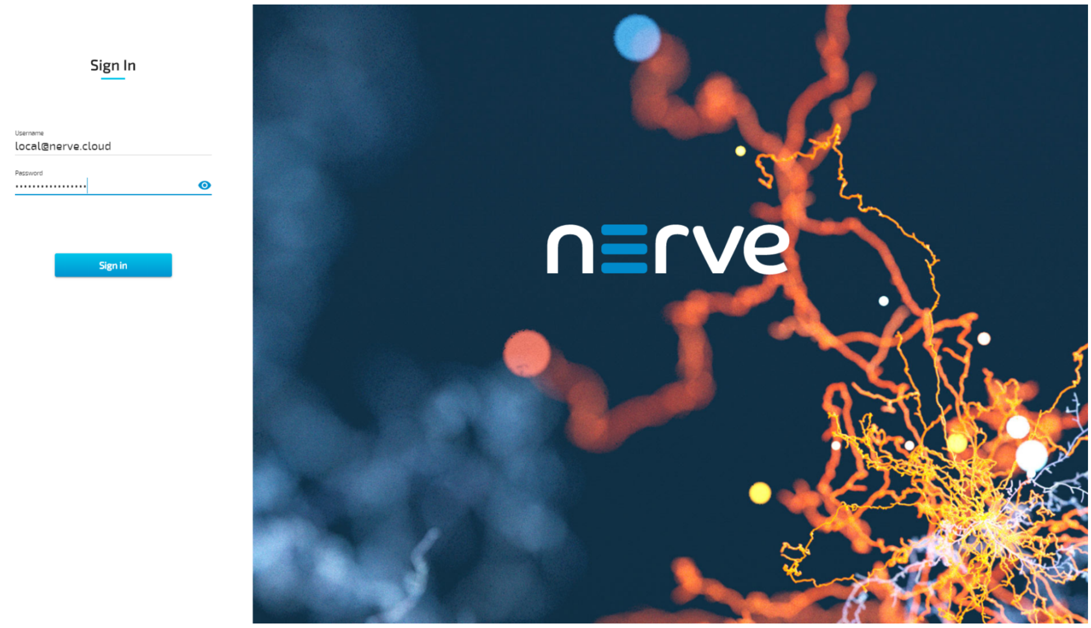
NOTE

It takes around 30 seconds until the Local UI can be accessed after license activation. Refresh the page until the Local UI login page is displayed.

Accessing the Local UI and registering the device

With the license activated, the node needs to be registered for use in the Management System through the Local UI. To access the Local UI, first connect a workstation to port **X1 P1** and configure the network adapter of the workstation. The IP address has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask. The credentials for the Local UI found in the customer profile are also required.

1. Follow this link to connect to the Local UI: http://172.20.2.1:3333/
2. Log in with the credentials from the customer profile.



Sign In

Username: local@nerve.cloud

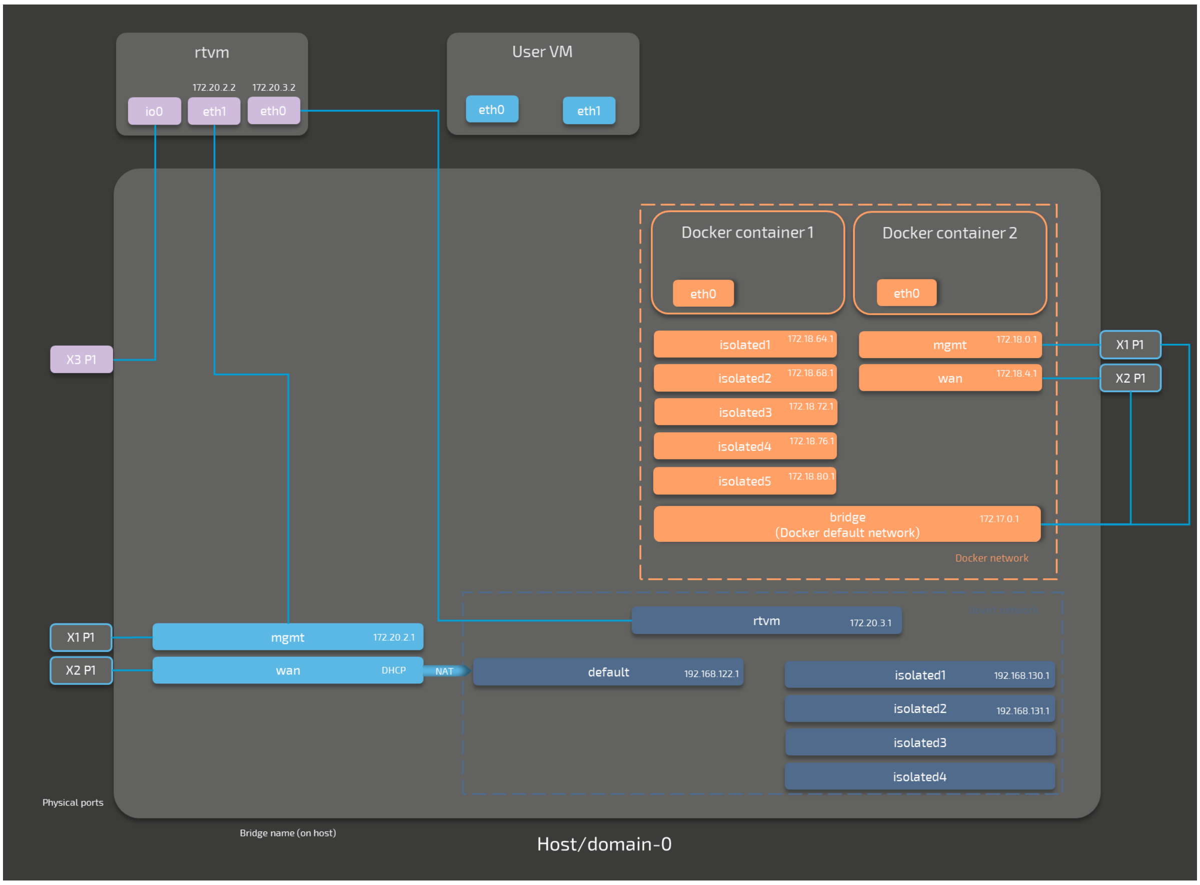
Password: ....

Continue with Node configuration for information on how to start registering the device in the Management System.

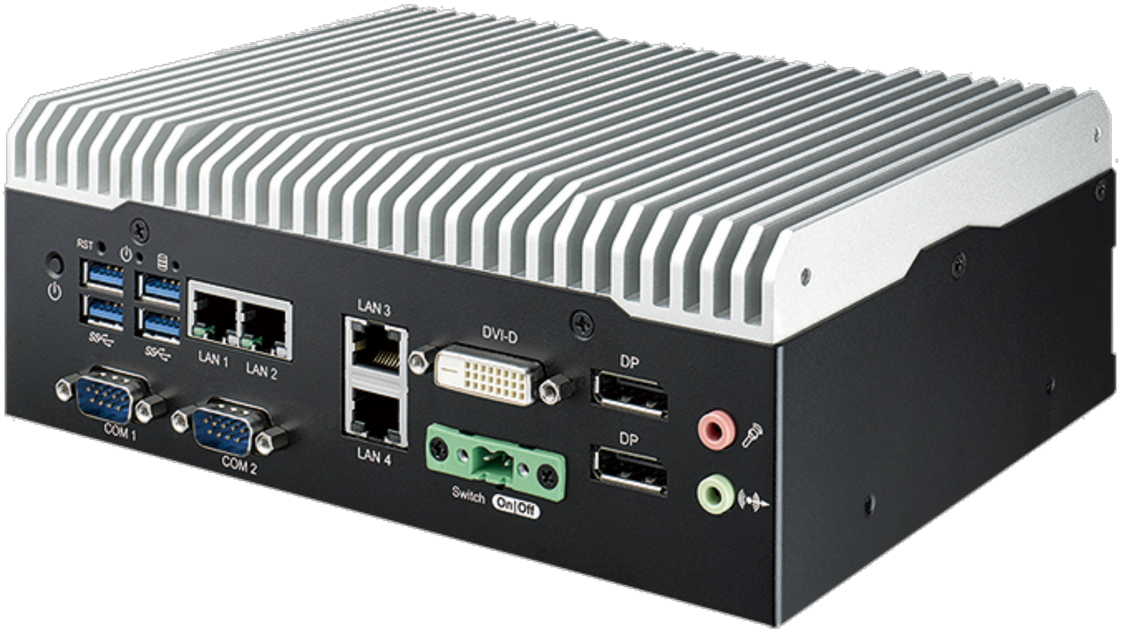
Physical ports and network interfaces

Below is a depiction of the node internal networking adapted to the SIMATIC IPC427E hardware. Refer to Node internal networking for more information. The table offers a quick overview of the network interfaces that can be reached through the physical ports of the SIMATIC IPC427E.

Physical port	Network name
X1 P1	mgmt
X2 P1	wan
X3 P1	io0



Vecow SPC-5600-i5-8500



The Vecow SPC-5600 Series is powered by 8 Cores Intel® Core™ i5 and can be used for vehicle computing, smart manufacturing, in-vehicle infotainment, intelligent control or any IIoT application.

For more information refer to the information materials provided by the manufacturer:

- Product page
- User manual

Device specifications

The table below contains the key specifications of the specific hardware model that has been certified for Nerve Blue usage. Use the article number listed here when ordering the device from the manufacturer only. Note that other device variants are not supported as Nerve Devices.

If required, contact sales@tttech-industrial.com for help with ordering Nerve Devices.

Item	Description
Article number	SPC-5600A-8500T32
CPU	Intel Core i5-8500T
Cores	6
RAM	32 GB DDR4
Storage	512 GB 2.5" SATA SSD MLC
TPM	TPM 2.0 included
Interfaces	• 4x GB LAN • 4x COM RS-232/422/485 (ESD 8KV) • 4x USB 3.1 (external) 1x USB 2.0 (internal) • 2x Display Port

Setting up the device for Nerve Blue usage

Requirements for the instructions below:

- a monitor with a display port input
- a keyboard
- a USB drive formatted to FAT32

Refer to the user manual of the manufacturer to set up the hardware. Connect a keyboard and a monitor to the device and make sure that the device is ready to be powered on. Also, prepare a USB drive in case the BIOS version of the device needs to be updated.

BIOS update

To avoid possible issues and complications, the BIOS version on the SPC-5600 needs to be version V27.01.03. or later. Refer to the user manual of the manufacturer on information how to update the BIOS version.

Required BIOS settings for Nerve Blue

Certain BIOS settings need to be changed to ensure the desired performance of the Nerve Blue system.

1. Power on the device.
2. Press Del while the device is booting to enter the BIOS menu.

3. Change the following settings:

Path	Setting
Advanced > CPU Configuration > Intel (VMX) Virtualization Technology	**Enabled**
Advanced > Power & Performance > CPU - Power Management Control > Boot performance mode	**Max Non-Turbo Performance**
Advanced > Power & Performance > CPU - Power Management Control > Intel (R) SpeedStep(tm)	**Disabled**
Advanced > Power & Performance > CPU - Power Management Control > Intel (R) Speed Shift Technology	**Disabled**
Advanced > Power & Performance > CPU - Power Management Control > C states	**Disabled**
Advanced > ACPI Settings > Enable Hibernation	**Disabled**
Advanced > ACPI Settings > ACPI Sleep State	**Suspend Disabled**
Advanced > Network Stack Configuration > Network Stack	**Enabled**
Chipset > System Agent (SA) Configuration > VT-d	**Enabled**
Security > Secure Boot > Secure Boot	**Disabled**

4. Save the changes and exit BIOS.

Installing Nerve Blue

Requirements for installing Nerve Blue on the device:

- a monitor with a display port input
- a keyboard
- a USB drive
- the Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img.tar.gz which can be downloaded from the Nerve Software Center
- a tool for creating bootable USB drives like Rufus on Windows

In addition, a workstation is required to prepare the bootable USB drive. Connect the monitor and the keyboard to the Vecow SPC-5600.

Before beginning with the installation, make sure that the device will boot from the USB drive. Press Del when the device is booting to enter BIOS and change the boot device settings.

On Windows

1. Download the Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img.tar.gz from the Nerve Software Center to a workstation.
2. Extract the Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img.tar.gz file to retrieve the Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img. Depending on the program used, the file might need to be extracted more than once.
3. Transfer the extracted Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img file to the USB drive using Rufus.
4. Plug the USB drive into a USB port of the Nerve Device.
5. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

On Linux

1. Download the Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img.tar.gz file from the Nerve Software Center.

2. Enter the following commands to extract the Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img.tar.gz file and transfer the extracted file to the USB drive:

```
tar xf Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img.tar.gz
sudo dd if=Nerve_Blue_USB-installer_2.1.2_for_vecow-spc-5600.img bs=4M of=/dev/sd<
sync
```

> **NOTE**
>
> Make sure to replace <drivename> with the system name of the USB drive.

3. Plug the USB drive into a USB port of the Nerve Device.

4. Make sure that the device will boot from the USB drive and power on the device.

The setup will start automatically and take a few minutes to complete. Select **OK** when the installation is complete and remove the USB drive. The device will reboot and reach a log in screen, asking for host access log in credentials. Make sure that the device will boot from the hard disk before rebooting the device.

Activating the Nerve Blue license

After the installation, the product license needs to be activated so that Nerve Blue can be used on the device. Connect a workstation to port **LAN 1** and configure the network adapter of the workstation. The IP address has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask.

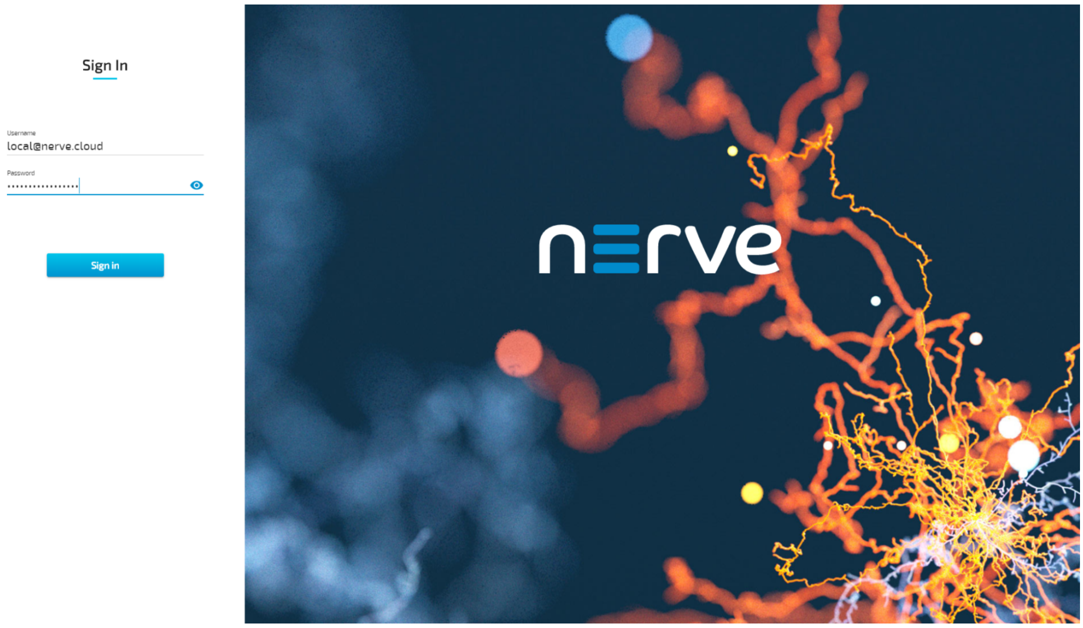
Access the license activation UI at http://172.20.2.1:3333/ and refer to License activation in the user guide for more information.

> **NOTE**
>
> It takes around 30 seconds until the Local UI can be accessed after license activation. Refresh the page until the Local UI login page is displayed.

Accessing the Local UI and registering the device

With the license activated, the node needs to be registered for use in the Management System through the Local UI. To access the Local UI, first connect a workstation to port **LAN 1** and configure the network adapter of the workstation. The IP address has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask. The credentials for the Local UI found in the customer profile are also required.

1. Follow this link to connect to the Local UI: http://172.20.2.1:3333/
2. Log in with the credentials from the customer profile.



Sign In

Username: local@nerve.cloud

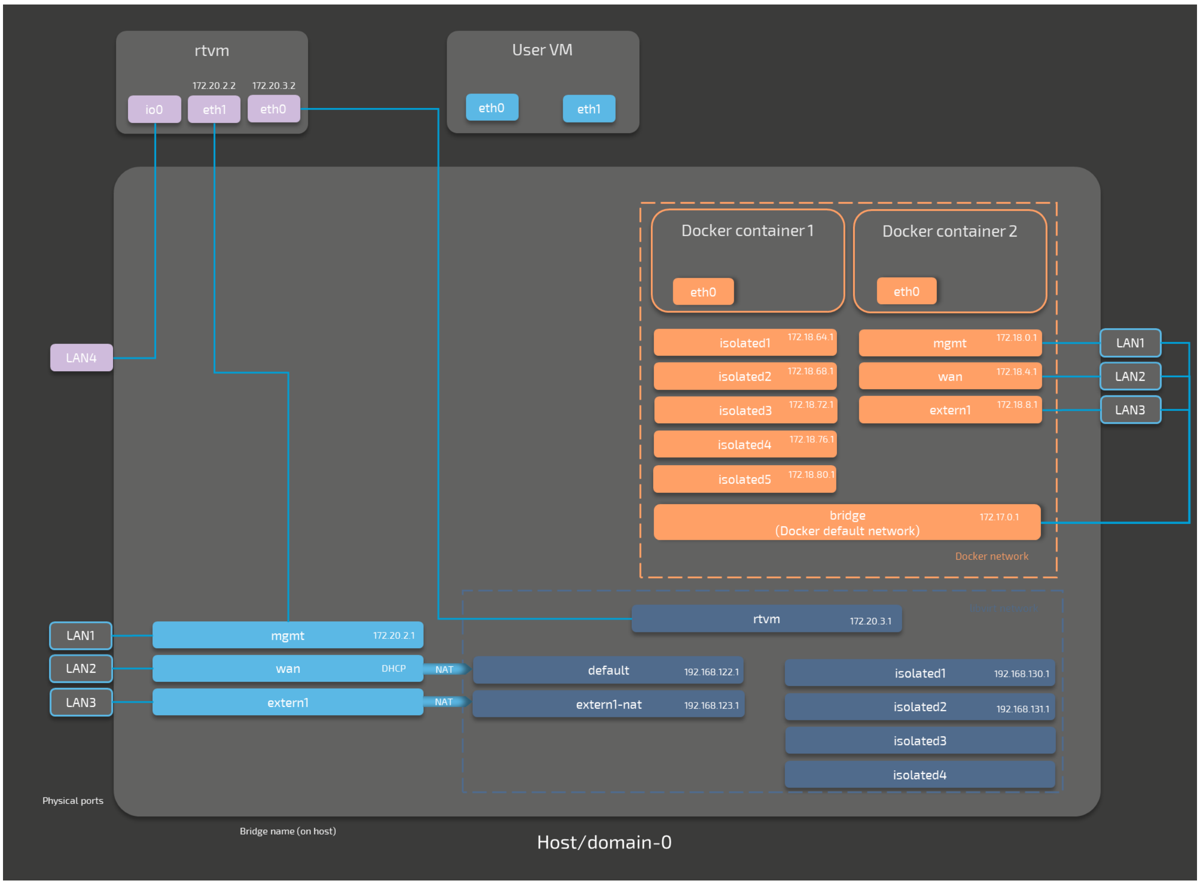
Password: ••••••••••••••••••

Continue with Node configuration for information on how to start registering the device in the Management System.

Physical ports and network interfaces

Below is a depiction of the node internal networking adapted to the SPC-5600 hardware. Refer to Node internal networking for more information. The table offers a quick overview of the network interfaces that can be reached through the physical ports of the SPC-5600.

Physical port	Network name
LAN 1	mgmt
LAN 2	wan
LAN 3	extern1
LAN 4	io0



Developer Guide

Developer Guide

The developer guide is a continuation of the user guide. It aims to give support to developers with tasks going beyond the regular usage of the Nerve Blue system. In this version of the Nerve Blue documentation it offers support with the following topics:

- an overview of node internal networking in the Nerve Blue system
- first steps to programming CODESYS applications on Nerve Devices
- an introduction to the usage of the Management System API for working with workloads

This guide will be expanded with future releases.

Nerve Data Services

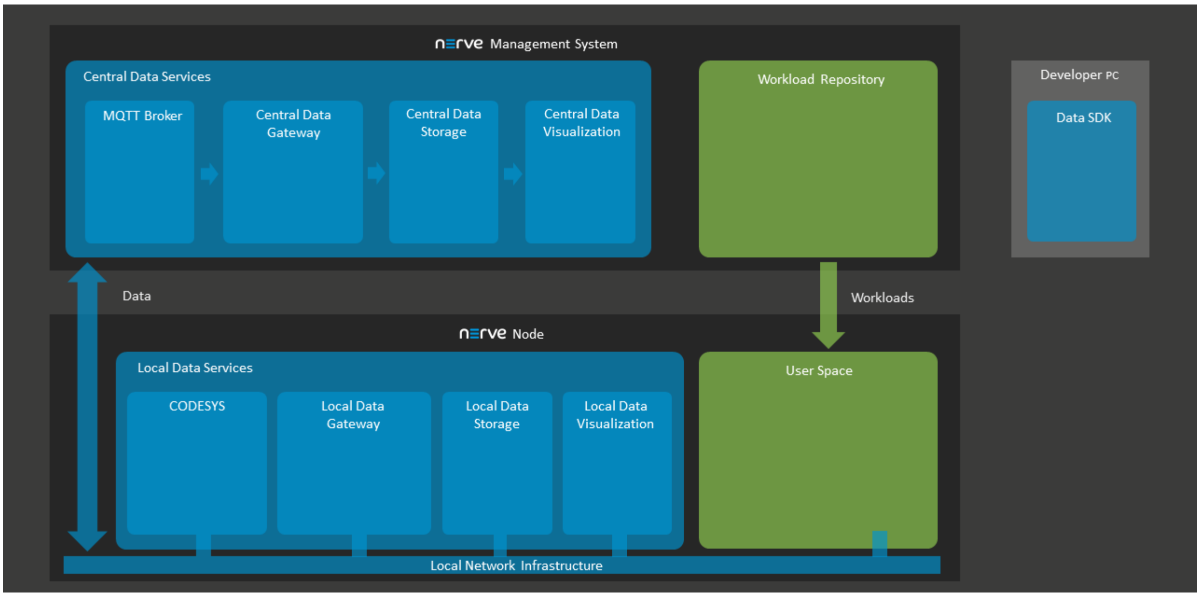
Nerve Data Services

The Nerve Data Services are a collection of services and interfaces that allow to collect, store, analyze, visualize and distribute data. These services and interfaces are available on the Nerve Blue node and in the Management System. Using Nerve Data Services, data can be collected at high speeds on the node and stored or visualized locally at the Node or centrally in the Management System.

Elements of Nerve Data Services

Nerve Data Services	Description and elements
Local Data Services	The Local Data Services are accessible on the node. They include the following elements: • CODESYS • Local Data Gateway • Local Data Storage • Local Data Visualization
Central Data Services	The Central Data Services are part of the Management System. They include the following elements: • Central Data Gateway • Central Data Storage • Central Data Visualization
User PC	Data SDK With the Data SDK users can create analytics applications, which can be deployed to a node.

All of the above elements are individually configurable. This allows for an application in a large number of scenarios and use cases. However, this also implies deeper knowledge of the elements and their possible interactions. Further explanations of the elements can be found in separate chapters. Find a high-level graphical overview below.



Supported protocols

Inputs	Outputs
MQTT subscriber	MQTT publisher
OPC UA PubSub over MQTT	OPC UA PubSub over MQTT
OPC UA PubSub over UDP	OPC UA PubSub over UDP
OPC UA Client	OPC UA server
Modbus server	ZeroMQ publisher
S7 server	Timescale DB
	Influx DB

Model workflow

The following section gives an overview of a typical work flow, presenting what can be done with the Nerve Data Services on a high level:

- Setting up data sources
- Configuring the local Gateway on the node to collect data and defining where to distribute it
- Configuring Grafana on the node for local data visualization
- Configuring the central Gateway in the Management System to collect data from different nodes
- Configuring Grafana in the Management System for central data visualization
- Configuring Grafana to visualize data
- Implementing an analytics app and configuring the local Gateway again to send data to the analytics app
- Reconfiguring Grafana to visualize the result of the processing performed by the analytics app

The paragraphs below give short explanations to each step.

Ingesting data on the node

CODESYS can be used to translate fieldbus protocols to OPC UA Server. The local Gateway can then be configured to collect data from CODESYS and distribute it to various destinations. Data sources that support protocols which are also native to the local gateway can be linked there directly and do not need CODESYS.

Distributing data on the node

Data is distributed by the local Gateway to user defined consumers. Unless the respective protocol of a Gateway output defines a specific data format, all data received by the Gateway is normalized to the Nerve Data Services data format. Typical consumers of data are the local data storage on the node or the central data storage in the Management System, third party MQTT brokers, applications or providing the data as OPC UA Server. To enable reuse of local Gateway configurations across multiple nodes, keywords can be used to abstract node specific information from the configuration.

Storage on the node

Data is stored on the node in a TimescaleDB database. Any application running on the node can access data stored in there. For application development, a Python API is provided in the Data SDK that simplifies reading and writing data. A time window can be configured for how long data shall be kept before deleting it.

Visualization on the node

Visualization is available via Grafana which is pre-configured to allow access to data stored in the local data storage. It is also possible to configure access from Grafana to other user defined data sources without any restrictions, meaning that data in an influxDB that is installed on the node can also be visualized.

Using Data in analytics

A Python SDK is available for creating custom analytics applications. An API is available that simplifies reading data from the local storage or receiving it directly from the Gateway when it arrives there. The API also allows to write back analytics results to the local storage for visualization or send it to an MQTT broker to distribute it to other consumers. It is also possible to use third party applications such as NodeRed or Crosser for analytics together with the Nerve Data Services.

Storing and visualizing in the Management System

Storage and visualization in the Management System has a similar structure technologically speaking but offer limited configuration possibilities as the network is static and there are no option to include workloads in the Management System.

Data received on the Node can be forwarded to the Management for visualization. To allow this, the local gateway and the central gateway as well as the central visualization must be configured. Keywords are in place to simplify the configuration process. Each node has a separate database in the Central Data Storage.

Nerve Data Services data format

Inputs and Outputs at the local and central gateway using a protocol that does not define a data format (i.e. MQTT) send or receive data in JSON format. The same applies to the Python API in the Analytics Data SDK. As a result, data within the Nerve Data Services is normalized to this format.

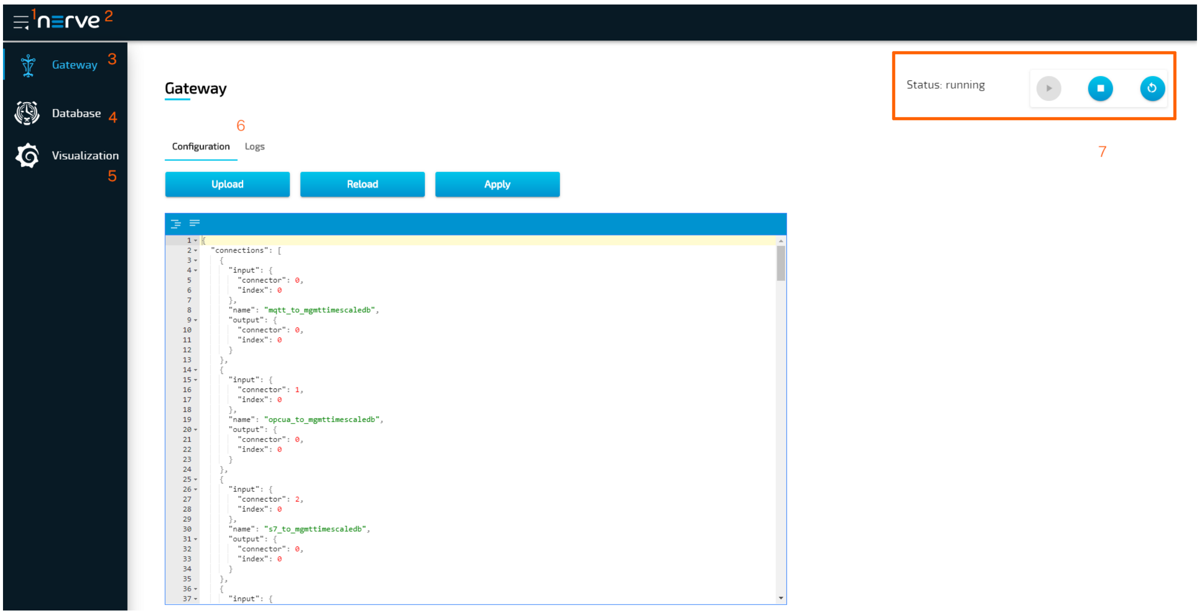
The JSON schema below describes the data format in more detail. The JSON schema is also used to validate data upon reception. If a data frame is received that does not comply with the schema, it is silently dropped by the Nerve Data Services.

```
{
    "$schema": "http://json-schema.org/draft/2019-09/schema#",
    "$id": "https://nerve.cloud/dp/dp_data_model.schema.json",
    "title": "Nerve Data Services JSON data model schema",
    "description": "Schema that represents default data model that Nerve Data Services
    "type": "object",
    "properties": {
        "variables": {
            "type": "object",
            "additionalProperties": {
                "anyOf": [
                    {
                        "type": [ "boolean", "number", "string" ]
                    },
                    {
                        "type": "array",
                        "items": {
                            "type": "boolean"
                        }
                    },
                    {
                        "type": "array",
                        "items": {
                            "type": "number"
                        }
                    },
                    {
                        "type": "array",
                        "items": {
                            "type": "string"
                        }
                    }
                ]
            }
        },
        "timestamp": {
            "type": [ "number", "string", "array" ],
            "items": {
                "type": [ "number", "string" ]
            }
        }
    },
    "required": [ "variables" ],
    "additionalProperties": false
}
```

The schema defines and allows only two properties in the root object: timestamp and variables. Timestamp is optional, and can be a single value or an array of values. Variables is an object containing arbitrary number of properties, which can be single values or arrays of values.

Data Services UI

An instance of the data Gateway is present on both the node and on the cloud. However, the instances are configured separately. The supervisor UI is identical in both instances, in the Local UI on the node and in the Management System.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

Item	Description
Burger menu (1)	Click here to expand and collapse the navigation on the left.
Nerve logo (2)	Return to the landing page of the Local UI or the Management System by clicking here.

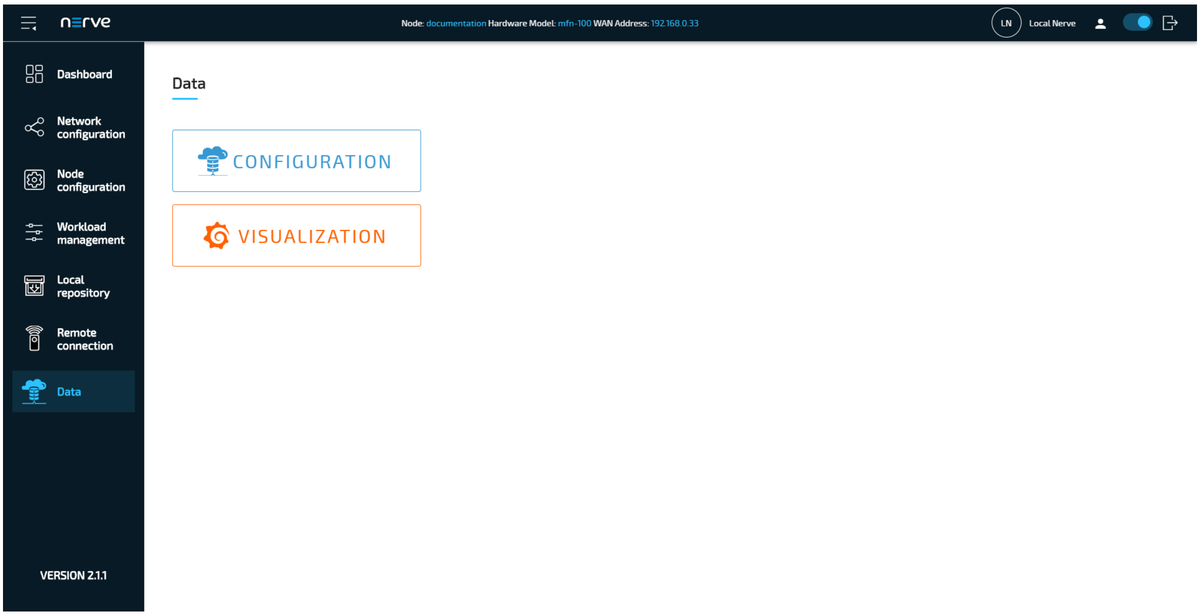
Item	Description
Gateway (3)	This is the central application of the Data Services. The current configuration of the Gateway is loaded into the open JSON editor by default. A new configuration can be typed in or loaded from a JSON file.
Database (4)	Data from databases created when a node is setup can be viewed here. The databases are named after the serial numbers of nodes and can be used to confirm the flow of data.
Visualization (5)	Data can be visualized in graphs. Refer to Data visualization for more information.
Configuration and Logs tabs (6)	Logs can be viewed within every Data Services element that can be selected in the navigation on the left. The latest logs of the selected Data Services element can be viewed within the **Logs** tab. The entirety of the logs can be exported into a file using the **Download** button in that tab. The **Configuration** tab is displayed by default.
Gateway controls (7)	This is the status indicator and the controls of the Gateway. **Running** indicates that the configuration has been accepted and the Gateway is running. When the status is **stopped**, there is either a problem in the configuration or the Gateway has been stopped manually.

Accessing the Data Services UI on the node

To access the Data Services UI on the node, access the Local UI. Connecting to the Local UI depends on the Nerve Device. Follow the link to the Local UI according to the used Nerve Device in the table below:

Nerve Device	Physical port	License activation UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

Select **Data** in the navigation on the left and select **CONFIGURATION** on the right to access the Data Services UI.

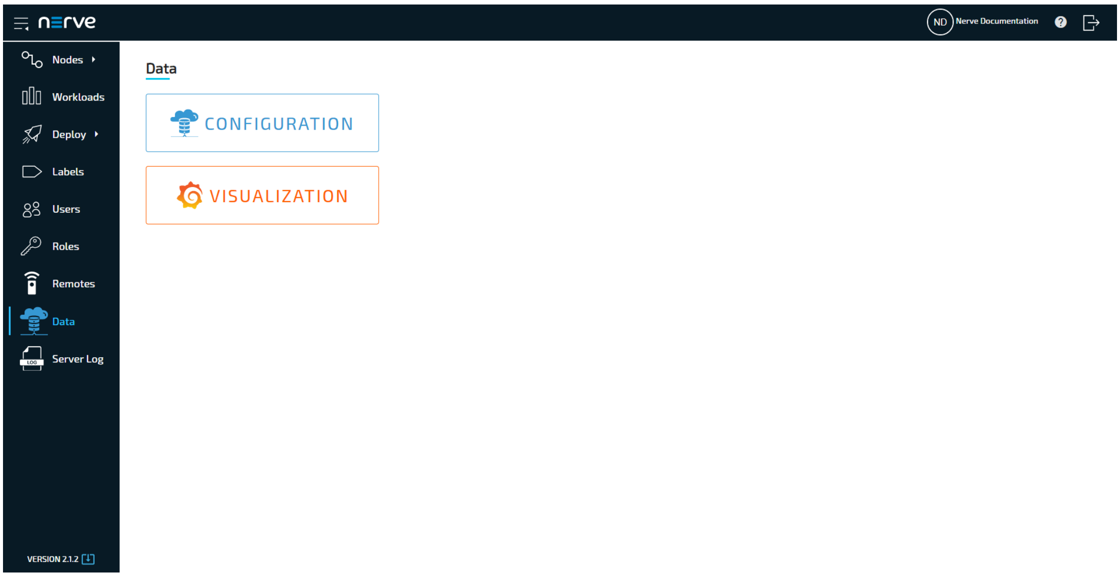


Accessing the Data Services UI in the Management System

By default, the permission to access to Data Services is not granted. Users that have the permission to create user roles can enable the Data Services entry in the Management System navigation by adding the **UI_NAV_DATAPATH:VIEW** permission to a user role. A **Data Services** user role is also defined in the Management System by default that can be added to a user. Refer to Users for information on how to create new users and to Roles and permissions for information on how to add a new role.

1. Log in to the Management System. Make sure the user has a role with the correct permission assigned.
2. Select **Data** in the navigation on the left.

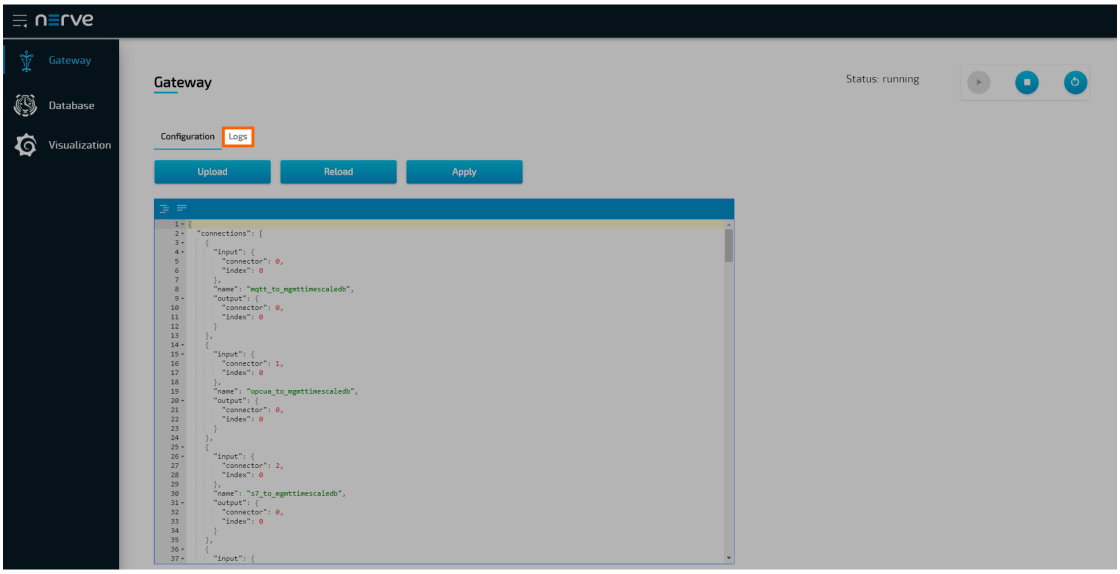
3. Select **CONFIGURATION** to access the Data Services UI.



How to export a log file

Every element of the Data Services is producing log files locally and centrally that can be viewed in the **Logs** tab. These logs can also be downloaded through the Data Services UI.

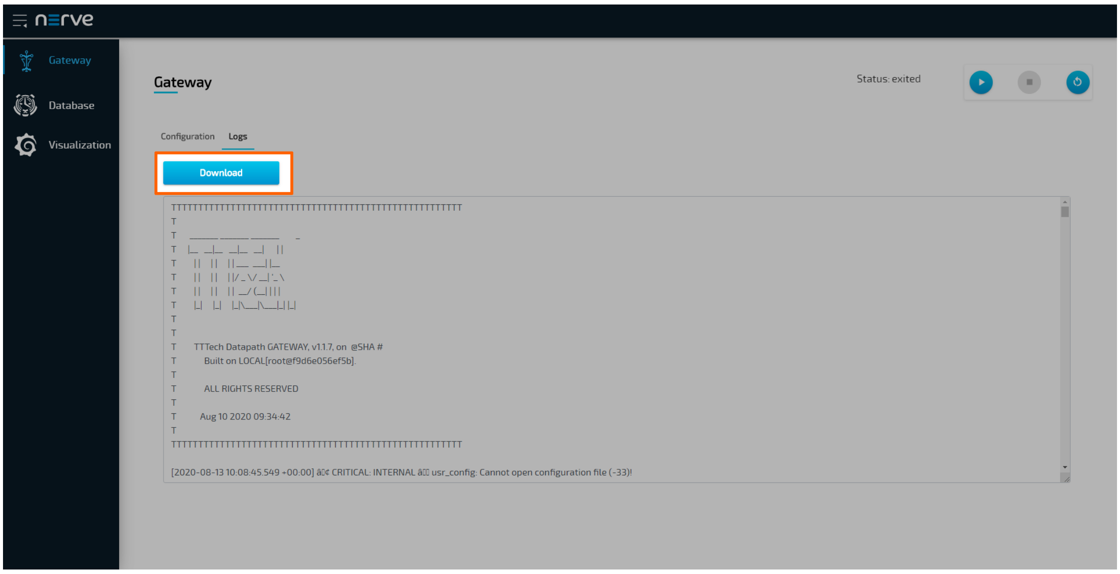
1. Select a Data Services element in the navigation on the left.
2. Select the **Logs** tab on the right.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

> **NOTE**
>
> Note that the visualization element does not have a **Logs** tab. Instead the logs are displayed on the first level and can be downloaded by selecting **Download**.

3. Select **Download** to export a LOG file.



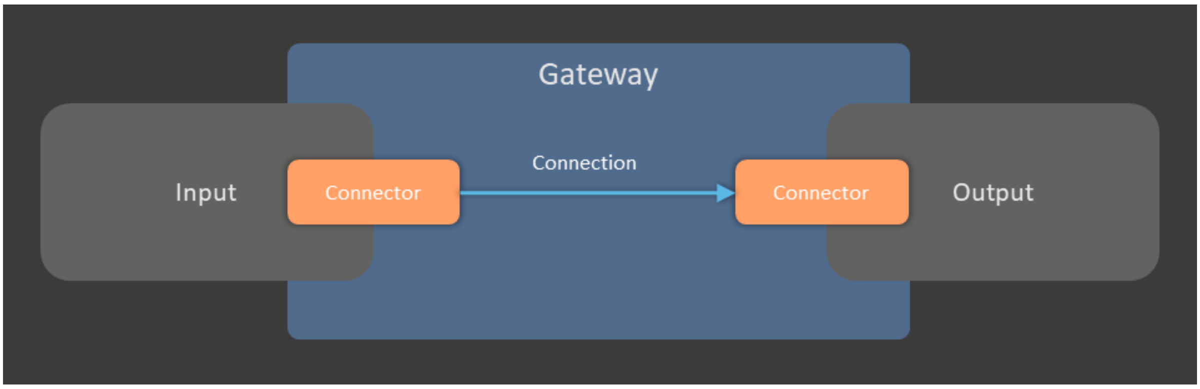
```
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
T
T    _______ _______ _______       _
T   |__   __|__   __|__   __|     | |
T      | |     | |     | | ___  ___| |__
T      | |     | |     | |/ _ \/ __| '_ \
T      | |     | |     | |  __/ (__| | | |
T      |_|     |_|     |_|\___|\___|_| |_|
T
T
T      TTTech Datapath GATEWAY, v1.1.7, on  @SHA #
T         Built on LOCAL[root@f9d6e056ef5b].
T
T         ALL RIGHTS RESERVED
T
T        Aug 10 2020 09:34:42
T
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

[2020-08-13 10:08:45.549 +00:00] â¢ CRITICAL: INTERNAL â usr_config: Cannot open configuration file (-33)!
```

Nerve Data Services Gateway

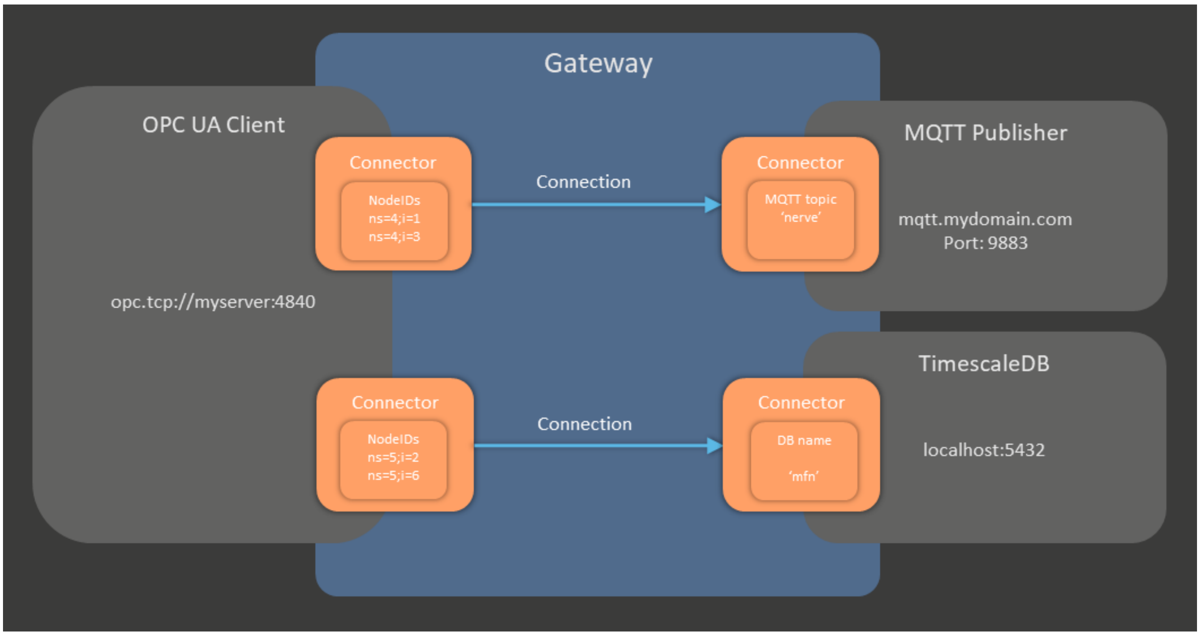
The Gateway is the central application of the Data Services. Its purpose is receiving data from a source via a certain protocol on an input interface and forwarding it to a destination using a different protocol on an output interface. This behavior is reflected in the user configuration as well. In general, the configuration file consists of inputs, outputs and connections between them:

- Inputs are collection interfaces where data is received.
- Outputs are providing interfaces where data can be received from.
- Connections are logical links between inputs and outputs.

Each input and output provides connectors where a connection can be attached to. A connector is a subset of the data available at an input. This subset is forwarded to the connector of an output where it is distributed further to the data receiver. On an abstract level, this concept is illustrated in the figure below.

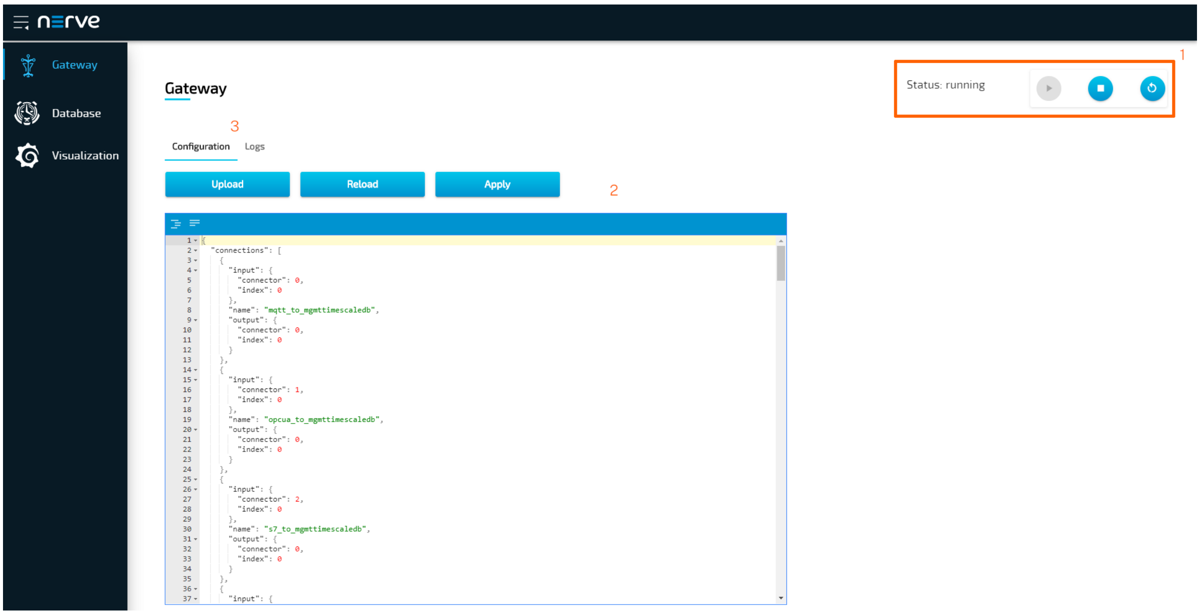


Inputs and outputs hold general configuration data for the protocol they implement. This can be a URL to a server, a port number or something related to timing. A connector of an input or output holds more specific information on how to get or where to send data, for example the node IDs of an OPC UA Server an OPC UA Client input connects to, or the name of a topic for an MQTT Publisher. Each input and output can provide multiple connectors at a time. Data from an input connector can also be connected to multiple output connectors at the same time to distribute data to multiple data consumers. The figure below shows a more specific example of that concept.



Data Gateway UI

The Data Gateway UI is the landing page of the Data Services and will be displayed by default after selecting **Data > CONFIGURATION** in the navigation on the left in either Local UI or Management System. The editor in the middle is used to apply Gateway configurations in the JSON format.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

Item	Description
Gateway status and controls (1)	The status of the Gateway is displayed in the upper-right corner with controls next to it that allow starting, stopping or restarting of the Gateway. **Running** indicates that the configuration has been accepted and the Gateway is running. When the status is **stopped**, there is either a problem in the configuration or the Gateway has been stopped manually.
Editor (2)	Enter a configuration in the JSON format into the editor to configure the Gateway. The configuration can be written directly in the editor or uploaded from the workstation. • **Upload** Select this to upload the contents of an existing JSON file from the workstation into the editor. • **Reload** Select this to load the currently applied configuration of the Gateway into the editor. • **Apply** Click here to set the current content of the editor as the Gateway configuration. The Gateway will be restarted.
Configuration and Logs tabs (3)	The **Configuration** tab is displayed by default. The latest logs of the Gateway can be viewed within the **Logs** tab. The entirety of the logs can be exported into a file using the **Download** button in this tab.

Applying a configuration to the Gateway

The configuration of the Gateway is applied in the editor in the Gateway section of the Data Services UI. The current configuration of the Gateway is loaded into the JSON editor if there is a configuration present in the Gateway. However, the editor will be empty on initial startup. A new configuration can be entered into the editor directly or imported from an existing JSON file.

- Upload existing JSON files using the **Upload** button.
- Apply the configuration in the editor by selecting **Apply**. When the configuration is successful, the status in the upper-right corner will display **running**.
- Check the **Logs** tab for any warnings or errors.

The basic structure of the configuration consist of a list of inputs such as an OPC UA server or an MQTT subscriber, outputs such as a Timescale database or an MQTT publisher, and connections between these inputs and outputs. Refer to Gateway configuration file syntax below for the exact syntax.

The examples in the Examples section show step by step applications in common use cases.

Gateway configuration file syntax

The following list gives additional clarification about how to set parameters with the JSON editor. The Gateway configuration JSON file includes sections, which are defined as main sections (inputs, outputs, connections). The main sections may contain one set of subsections or an array of items, which in this case are defined as an array of subsection sets.

Subsections can also include additional subsections. A section is defined as an array if the section contains two or more subsections. Subsection keys are defined as parameters. Every parameter has its field for a value. The main sections can contain the same and different parameters as other sections.

The syntax of the various inputs and outputs of the Gateway available in its configuration file is illustrated below for the available protocols.

Inputs	Outputs
MQTT subscriber	**MQTT publisher**
OPC UA PubSub over MQTT	**OPC UA PubSub over MQTT**
OPC UA PubSub over UDP	**OPC UA PubSub over UDP**
OPC UA Client	**OPC UA server**
Modbus server	**ZeroMQ publisher**
S7 server	**Timescale DB**
	Influx DB

Inputs

In the tables below, fields in **bold** are mandatory and fields in *italics* are optional when writing a configuration for the corresponding protocol.

OPC UA Client

Item	Description
type	Type of input (OPC_UA_CLIENT).
name	Name of OPC_UA_CLIENT instance used in log messages.
serverUrl	URL of the server to connect to, i.e. `opc.tcp://myserver.com:4840`.

Item	Description
login	Authentication parameters. **username** Username for client authentication. **password** Password for client authentication.
certificate	Secure connection parameters. **certFilePath** Certificate file path for client authentication. **keyFilePath** Key file path for client authentication.
trustList	Array of certificates of trusted servers.
security	Connection security mode and policy. **securityMode** Security mode (None, Sign, SignAndEncrypt). **securityPolicy** Security policy (None, Basic128Rsa15, Basic256, Basic256Sha256).
description	Description of the client. *applicationUri* Application URI of the client, used in secure connection certificate.
pollingInterval_ms	Interval at which to poll values from the server in milliseconds.
connectors	Array of connectors for OPC_UA_CLIENT. **name** Connector name. **accessType** Access mode, either `polling` or `subscription`. *publishingIntervalAtServer_ms* Interval at which the server is publishing values in milliseconds. *samplingIntervalAtServer_ms* Sampling intervall at which the server updates its values in milliseconds. **nodes** Array of nodeID strings, e.g. `ns=4;i=1`, `ns=mynamespace;s=myvar.int0`. If the nodeID of an object is provided, all variables of that object are used.

MQTT Subscriber

Objects	Values and descriptions
type	Type of input (MQTT_SUBSCRIBER).
name	Name of the MQTT_SUBSCRIBER input used in log messages.
clientId	Client ID of the MQTT_SUBSCRIBER.

Objects	Values and descriptions
serverUrl	Protocol and URL of the server to connect to, i.e. `tcp://myserver.com:4840`. Possible `ssl`, `ws` or `wss`.
username	MQTT broker authentication username.
password	MQTT broker authentication password.
keepAliveInterval_s	Maximum time allowed of no communication between client and server.
cleanSession	Defines whether the server should remember state for the client across reconnect.
qos	Quality of Service value (0-2).
sslOptions	Secure connection options: *ca_certsFilePath* Path to CA certificate file. *certFilePath* Path to certificate file. *keyFilePath* Path to key file. *keyPassword* Password for the given key. *serverAuthRequired* Defines whether server authentication is required.
connectors	Array of connectors for the MQTT_SUBSCRIBER: **name** Connector name. **topic** Name of the MQTT topic to use. **variables** Array of variables expected to be received: • **name** Variable name as found in the received message. • **type** Data type Gateway uses for further processing, must match receiving variable data t `sbyte`, `byte`, `int16`, `uint16`, `int32`, `uint32`, `int64`, `uint64`, `double`, `float`, or `guid`.

OPC UA PubSub Subscriber

Item	Description
type	Type of input (OPC_UA_PS_SUBSCRIBER).
name	Name of the OPC_UA_PS_SUBSCRIBER input used in log messages.
mqttClientId	Client ID of the OPC_UA_PS_SUBSCRIBER when MQTT transport is used.
networkAddressUrl	URL of the OPC UA PubSub connection to subscribe to.
networkInterface	Name of the network interface to use.

Item	Description
secureConnectionOptions	Secure connection options for MQTT transport: *ca_certsFilePath* Path to CA certificate file. *certFilePath* Path to certificate file. *keyFilePath* Path to key file. *keyPassword* Password for the given key. *serverAuthRequired* Defines whether server authentication is required. *username* Username for authentication. *password* Password for authentication.
writerGroups	Array of writer groups at this connection: *publisherId* Publisher ID for the writer group. *writerGroupId* Writer group ID. *fieldsReversePublished* Some publishers send fields within a data set message in a reversed order compared publisher. If this field is set to true, the user can use field indexes in the config file in th these publishers. Must be set to `true` when connecting to a publisher using the TTTec *mqttTopic* Name of the MQTT topic if transport protocol is MQTT, ignored otherwise.

Item	Description
connectors	Array of connectors for the OPC_UA_PS_SUBSCRIBER: **name** Connector name. *writerGroupIdx* Index of the writer group from the array of writer groups above. *dataSetWriterId* Data set writer ID. **variables** Array of variables expected to be received: • *fieldIndex* Field index of the variable in the data set. • **name** Name of the variable as found in the received message. • **type** Data type of the variable. Possible values are `bool`, `int8`, `uint8`, `int16`, `uin` `double`, `float`, `string`, `bytestring`, `datetime` or `guid`.

S7 Client

Item	Description
type	Type of output (S7_CLIENT).
name	Name of the S7_CLIENT instance used in log messages.
serverUrl	URL of the S7 server to connect to.
port	Port at which to connect to.
connectionType	Connection type (PG, OP, S7_BASIC).
localTsap	Local tsap, mandatory if s7 basic connection type.
remoteTsap	Remote tsap, mandatory if s7 basic connection type.
pollingInterval_ms	Interval for polling values from server in miliseconds.
rack	S7 device rack.
slot	S7 device slot.

connectors

Array of connectors for S7_CLIENT:

name
Connector name.

merkers
Array of S7 merkers:

- **name**
  S7 merker name.
- **offset**
  Address of first data.
- *quantity*
  Number of merkers to be read.
- **type**
  S7 merker data type.

inputs
Array of S7 inputs:

- **name**
  S7 input name.
- **offset**
  Address of first data.
- *quantity*
  Number of inputs to be read.
- **type**
  S7 input data type.

outputs
Array of S7 outputs:

- **name**
  S7 output name.
- **offset**
  Address of first data.
- *quantity*
  Number of outputs to be read.
- **type**
  S7 output data type.

timers
Array of S7 timers:

- **name**
  S7 timer name.
- **offset**
  Address of first data.
- *quantity*
  Number of timers to be read.

counters
Array of S7 counters:

- **name**
  S7 counter name.
- **offset**
  Address of first data.
- *quantity*
  Number of counters to be read.

datablocks
Array of S7 datablocks:

- **name**
  S7 datablock name.
- **datablock**
  S7 datablock number

Modbus Client

Item	Description
type	Type of output (MODBUS_CLIENT).
name	Name of the MODBUS_CLIENT instance used in log messages.
serverUrl	URL of the Modbus server to connect to.
port	Port at which to connect to.
pollingInterval_ms	Interval at which to poll values from the server in milliseconds.
connectors	Array of connectors for the MODBUS_CLIENT: **name** Connector name. *coils* Array of Modbus coil type: • **name** Coil name, single string or an array of string. • **address** Starting address of first instance. • *quantity* Number of coils to be read. *discreteInputs* Array of Modbus discrete input type: • **name** Discrete input name, single string or an array of string. • **address** Starting address of first instance. • *quantity* Number of discrete inputs to be read. *inputRegisters* Array of Modbus input register type: • **name** Input register name, single string or an array of string. • **address** Starting address of first instance. • *quantity* Number of input registers to be read. • **type** Input register data type. *holdingRegisters* Array of Modbus holding register type: • **name** Holding register name, single string or an array of string. • **address** Starting address of first instance. • *quantity* Number of holding registers to be read. • **type** Holding register data type.

Outputs

In the tables below, fields in **bold** are mandatory and fields in *italics* are optional when writing a configuration for the corresponding protocol.

TimescaleDB

Item	Description
type	Type of output (DB_TIMESCALE).
name	Name of the DB_TIMESCALE instance used in log messages.
url	URL of the TimescaleDB server to connect to.
port	Port at which to connect to.
dataRate_MBps	Hypertable data rate in MegaBytes per second.
chunkTimeInterval_s	Hypertable chunk time interval in seconds.
connectors	Array of connectors for DB_TIMESCALE: **name** Connector name. *dbName* Database name to connect to. *tableName* Table name to write into. *user* Username for authentication. *password* Password for authentication. *booleanAsSmallint* Whether boolean should be represented as PostgreSQL small int data type.

InfluxDB

Item	Description
type	Type of output (DB_INFLUX).
name	Name of the DB_INFLUX instance used in log messages.
url	URL of the InfluxDB server to connect to.
port	Port at which to connect to.

Item	Description
connectors	Array of connectors for DB_INFLUX: **name** Connector name. **dbName** Database name to connect to. *user* Username for authentication. *password* Password for authentication.

MQTT Publisher

Item	Description
type	Type of output (MQTT_PUBLISHER).
name	Name of the MQTT_PUBLISHER instance used in log messages.
clientId	Client ID of the MQTT_PUBLISHER.
serverUrl	Protocol and URL of the server to connect to, i.e. `tcp://myserver.com:4840`. Possible protocols are: `tcp`, `mqtt`, `mqtts`, `ssl`, `ws` or `wss`.
username	MQTT broker authentication username.
password	MQTT broker authentication password.
keepAliveInterval_s	Maximum time allowed of no communication between client and server.
cleanSessions	Defines whether the server should remember state for the client across reconnect.
qos	Quality of Service value (0-2).
sslOptions	Secure connection options: *ca_certsFilePath* Path to CA certificate file. *certFilePath* Path to certificate file. *keyFilePath* Path to key file. *keyPassword* Password for the given key. *serverAuthRequired* Defines whether server authentication is required.

Item	Description
connectors	Array of connectors for MQTT_PUBLISHER: **name** Connector name. **topic** Name of the MQTT topic to use. *timestampRequired* Defines whether a timestamp is added to each message or not. *timestampFormat* Format of the timestamp in a message, either `iso` (ISO 8601) or `unix_ns` (UNIX time in nanoseconds since Jan 01 1970 (UTC)). *maxListSize* Maximum array size of variables, if sent as arrays. If size is larger than the max data will be sent in chunks of maxListSize.

OPC UA PubSub Publisher

Items	Description
type	Type of output (OPC_UA_PS_PUBLISHER).
name	Name of OPC_UA_PS_PUBLISHER instance used in log messages.
mqttClientId	Client ID of OPC_UA_PS_PUBLISHER when MQTT transport is used.
networkAddressUrl	URL of the OPC UA PubSub connection to publish to.
networkInterface	Name of the network interface to use.
keepAlive_s	Maximum time allowed of no MQTT communication between client and server.
secureConnectionOptions	Secure connection options for MQTT transport: *ca_certsFilePath* Path to CA certificate file. *certFilePath* Path to certificate file. *keyFilePath* Path to key file. *keyPassword* Password for the given key. *serverAuthRequired* Defines whether server authentication is required. *username* Username for authentication. *password* Password for authentication.

Items	Description
pubsubConnections	Array of OPC UA PubSub connections *publisherId* Publisher ID of the connection. **writerGroups** List of writer groups for this Publisher ID: • *writerGroupId* Writer group ID. • **publishInterval_ms** Interval at which client publishes data in milliseconds. • *mqttTopic* MQTT topic name. • *qos* Quality of Service value (0-2). • **dataSetWriters** Array of data set writers: ▪ *dataSetWriterId* Data set writer ID. ▪ *keyFrameCount* Number of delta frames allowed to be sent before sending a key fr ▪ **publishedDataSetIdxi** Index in the list of published data set to be used by this dat set wri **publishedDataSets** Array of published data sets: • **fields** Array of fields (variables) of the published data sets: ▪ **name** Name of the field (variable) used within the Gateway. ▪ **type** Data type of the field (variable). Possible values are `bool`, `int8`, `uint64`, `double`, `float`, `string`, `bytestring`, `datetime` or
connectors	Array of connectors for OPC_UA_PS_PUBLISHER: **name** Connector name. **publisherIdx** Index of the publisher in the array of OPC UA PubSub connections. **writerGroupIdx** Index in the list of writer groups at the OPC UA PubSub connection. **dataSetWriterId** Index in the list of data set writers in the writer group at the OPC UA PubSub connecti

OPC UA Server

Item	Description
type	Type of output (OPC_UA_SERVER).
name	Name of OPC_UA_SERVER instance used in log messages.
customHostname	Custom hostname of the server, used for discovery URL.

Item	Description
port	Port to listen at (defaults to 4840).
addressSpaceFolders	OPC_UA_SERVER address space tree: **browseName** OPC UA Folder browse name. *displayName* OPC UA Folder display name. *description* OPC UA Folder description. *namespaceURI* OPC UA Folder namespace URI. **identifier** OPC UA Folder nodeID identifier. *children* Array of OPC UA childFolders, same format as addressSpaceFolders field. *connectorIndicies* Array of indicies of connectors, placed in the OPC UA Folder.
connectors	Array of connectors for OPC_UA_SERVER: **name** Connector name. **browseName** OPC UA node browse name. *displayName* OPC UA node display name. *description* OPC UA node description. *namespaceURI* OPC UA node namespace URI. **identifier** OPC UA node nodeID identifier.

ZeroMQ Publisher

Item	Description
type	Type of output (ZEROMQ_PUBLISHER).
name	Name of the ZEROMQ_PUBLISHER instance used in log messages.
serverUrl	Server endpoint to publish to. Subscribers connect to it.

Item	Description
connectors	Array of connectors for ZEROMQ_PUBLISHER: **name** Connector name. **topic** Name of the ZeroMQ topic to publish to. *timestampRequired* Defines whether a timestamp is added to each message or not. *timestampFormat* Format of the timestamp in a message, either `iso` (ISO 8601) or `unix_ns` (UNIX time in nanoseconds since Jan 01 1970 (UTC)). *maxListSize* Maximum array size of variables, if sent as arrays. If size is larger than the max data will be sent in chunks of maxListSize.

Connections

Item	Description
name	Name of the connection, will define the name of the TABLE within TimescaledDB if `tableName` field is omitted.
input	Input of the connection. **index** Index in the array of INPUTs. **connector** Index in the array of connectors at the input.
output	Output of the connection. **index** Index in the array of OUTPUTs. **connector** Index in the array of connectors at the output.

Gateway configuration file keywords

Parts of the Gateway configuration are eased by the use of keywords. Keywords can be used on both Gateway instances on the node and in the Management System. However, the keyowrds that can be used differ. All keywords must be put in angled brackets, for example `<SN>`. To be able to use keywords. the node must be registered in the Management System.

Refer to the list below for keywords that can be used with usage examples right after.

Supported on the node

Keyword	Description
SN	• Replaces the keyword with the serial number of the node. • Used as a username for MQTT Subscriber/Publisher when connecting to the Management System broker. • Can also be used as a root MQTT topic.
SID	• Replaces the keyword with the secure ID of the node. • Used as a password for MQTT Subscriber/Publisher when connecting to the Management System broker.
MGMT-SYS	• Replaces the keyword with the URL of the Management System the node is registered in. • Used in `serverUrl` field of the MQTT Subscriber/Publisher when connecting to the Management System broker. • Adds unique `clientId` field to the input or output the keyword is found in. If `clientId` already exists, it is replaced.
LOCAL	• Replaces the keyword with `localhost`. • Used in `url` field of TimescaleDB output. • Adds `port` field to the TimescaleDB output. • Adds `connectors` array with a single connector to the TimescaleDB output. If `connectors` already exists, replaces it. Adds `name`, `dbName`, `user` and `password` fields to that connector.

Supported in the Management System

Keyword	Description
SN:<nodename>	• Replaces the keyword with the serial number of the node that was specified in the keyword. • Used in `dbName` field of TimescaleDB outputs connectors. • Can also be used as a root MQTT topic. • When used in `dbName` field of TimescaleDB outputs connector, adds `user` and `password` fields to it.
MS-MQTT-BROKER	• Replaces the keyword with the URL of the MQTT broker in the Management System. • Used in `serverUrl` field of the MQTT Subscriber/Publisher when connecting to the Management System broker. • Adds unique `clientId` field to the input or output the keyword is found in. If `clientId` already exists, it is replaced. Also adds `username` and `password` fields to the same input or output.

Examples of keyword usage

SN:<nodename>

```
{
    "name": "cloud_timescale",
    "type": "DB_TIMESCALE",
    "url": "localhost",
    "port": 5432,
    "connectors": [
        {
            "name": "EXAMPLE-NODE",
            "dbName": "<SN:EXAMPLE-NODE>"
        }
    ]
}
```

SN, SID, MGMT-SYS

```
{
    "name": "ms_mqtt_broker",
    "type": "MQTT_SUBSCRIBER",
    "serverUrl": "wss://<MGMT-SYS>:443/dp/mqtt/wss",
    "username": "<SN>",
    "password": "<SID>",
    "keepAliveInterval_s": 20,
    "qos": 1,
    "cleanSession": true,
    "sslOptions": {
        "serverAuthRequired": false
    },
    "connectors": [
        {
            "name": "demo_sensor_connector_0",
            "topic": "<SN>/demo_sensor_topic_0",
            "variables": [
                {
                    "name": "temperature",
                    "type": "int64"
                }
            ]
        }
    ]
}
```

LOCAL

```
{
    "type": "DB_TIMESCALE",
    "name": "timescaledb_0",
    "url": "<LOCAL>"
}
```

SN:<nodename>, MS-MQTT-BROKER

```
{
    "name": "ms_mqtt_broker",
    "type": "MQTT_SUBSCRIBER",
    "serverUrl": "<MS-MQTT-BROKER>",
    "keepAliveInterval_s": 20,
    "qos": 1,
    "cleanSession": true,
    "connectors": [
        {
            "name": "demo_sensor_connector_1",
            "topic": "<SN:EXAMPLE-NODE>/demo_sensor_topic_1",
            "variables": [
                {
                    "name": "humidity",
                    "type": "uint64"
                }
            ]
        }
    ]
}
```

Data visualization

Data stored in the Data Services can be visualized via Grafana, an open source web application which provides charts, graphs and alerts for data visualization. An instance of Grafana is available on each node and in the Management System to visualize data stored in the TimescaleDB on the node or in the Management System respectively. This chapter describes only how Grafana is configured and how it can be accessed. Refer to the Grafana documentation for general information on how to use Grafana.

> **NOTE**
>
> To use Grafana, the user must be logged in to the Local UI or the Management System respectively. When logged out, Grafana is no longer able to perform its internal operations and reports random errors. To continue working with Grafana, close the Grafana tab, log in again at either the Local UI or the Management System, depending on which Grafana instance is being accessed, and re-open Grafana.

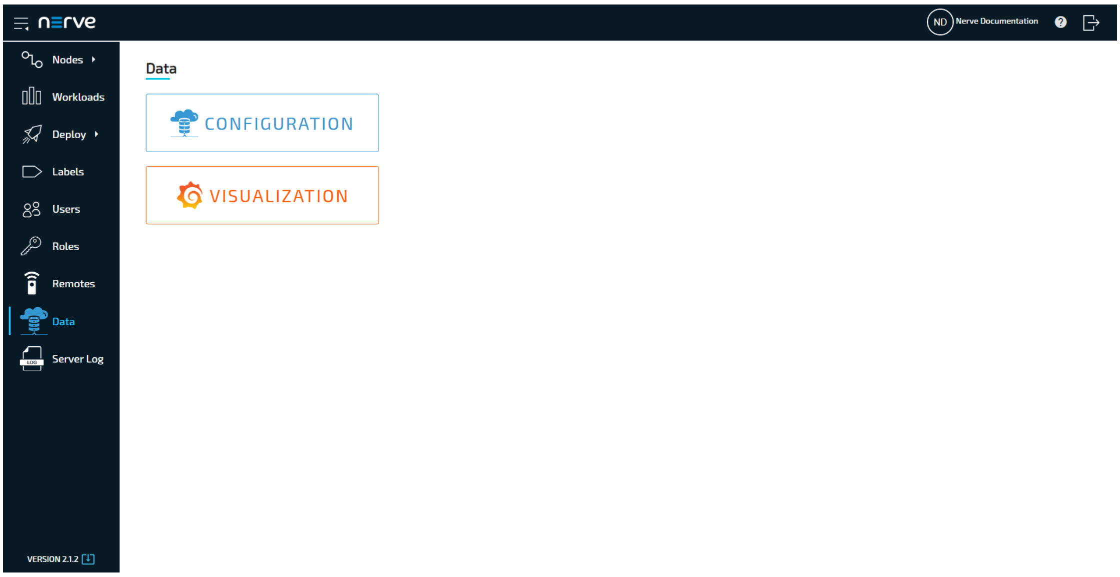
Central Data Visualization in the Management System

In the Management System, each registered node is represented as a separate data source in Grafana. The data source is created during the registration of a node. A data source is named after the node it represents, formatted as `<nodename> (<serialnumber>)`. When creating a new panel, select the node which sent the data to be displayed as the data source. The name of the table must match the name provided in the Gateway configuration file.

1. Log in to the Management System.
2. Select **Data** in the navigation on the left.

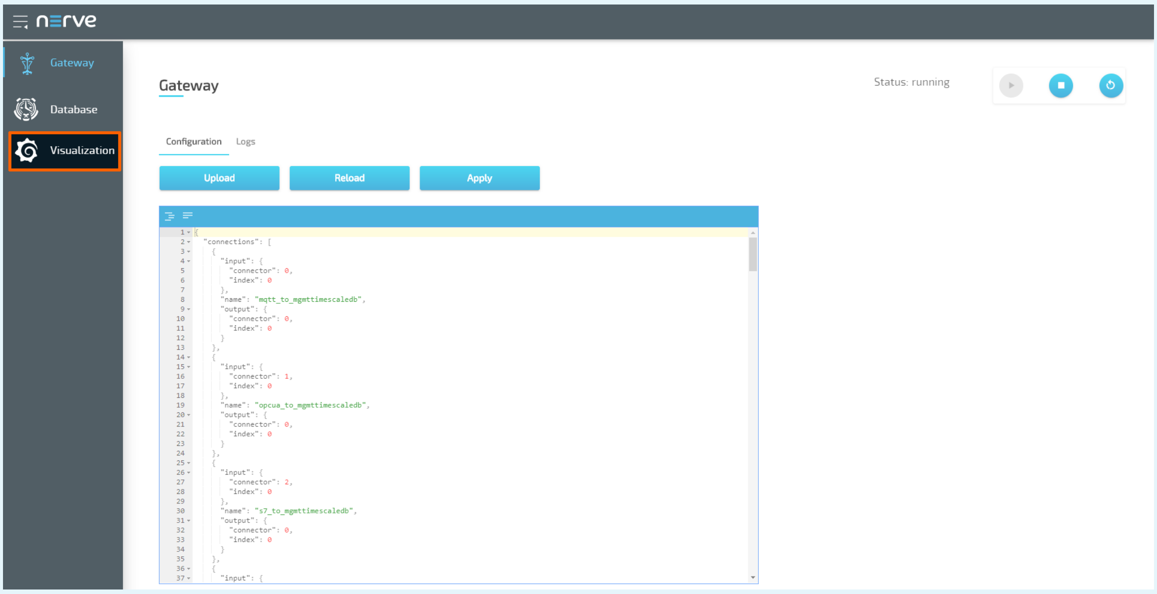
   > **NOTE**
   >
   > If the menu item **Data** is not available, make sure the logged in user has the permission to access the Data Services. Refer to Assigning a role to a user for more information.

3. Select **VISUALIZATION**.



NOTE

The visualization element can also be reached from the Data Services UI. When in the Data Services UI, select **Visualization** in the navigation on the left and select **Open** to reach the Grafana UI.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

A new tab opens that shows the home screen of Grafana. Here dashboards can be created as described below.

Local Data Visualization on the node

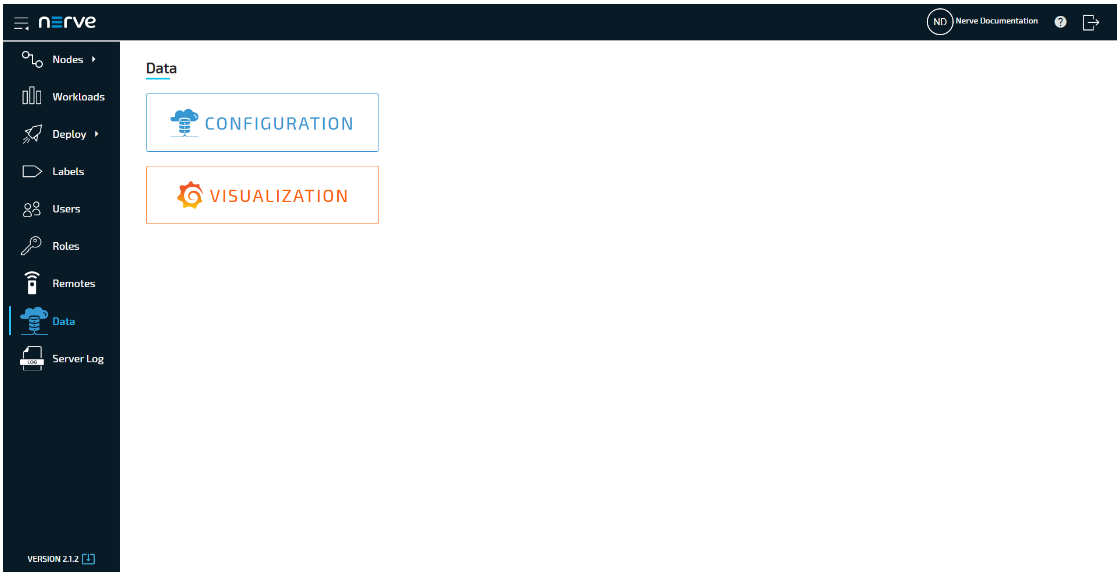
On the node, only a single data source exists in Grafana, formatted as `<serialnumber>`. Use this data source in all panels to visualize data received by the Data Services.

1. Access the Local UI on the node. This is Nerve Device specific. Refer to the table below for device specific links to the Local UI. The login credentials to the Local UI can be found in the customer profile.

Nerve Device	Physical port	Local UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

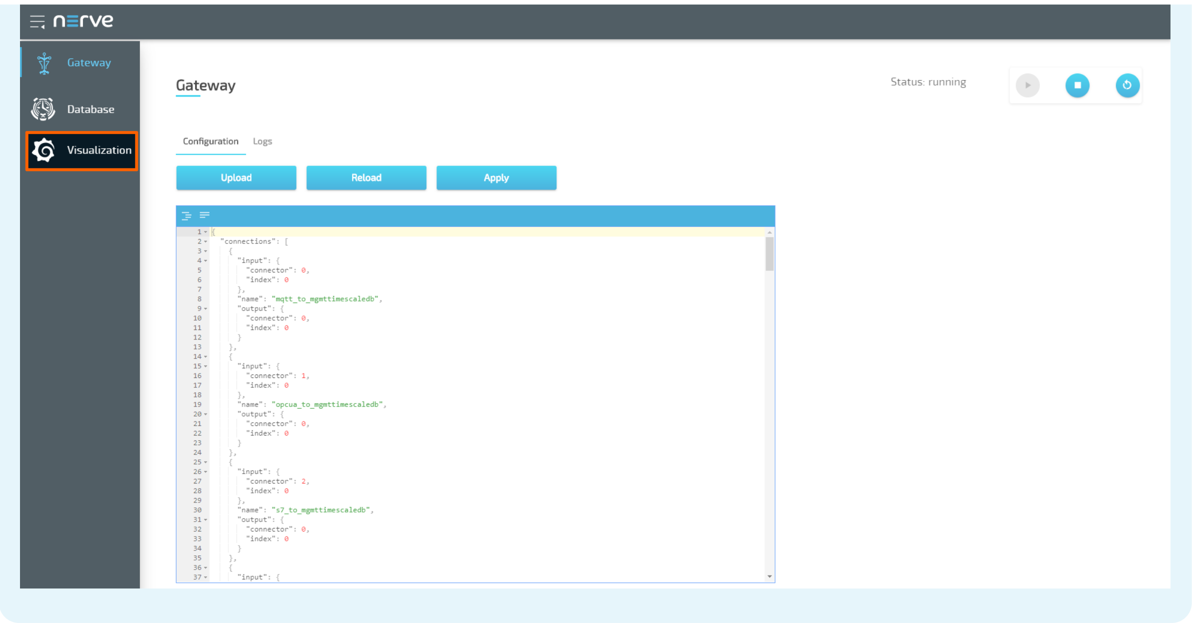
2. Select **Data** in the navigation on the left.

3. Select **VISUALIZATION**.



NOTE

The visualization element can also be reached from the Data Services UI. When in the Data Services UI, select **Visualization** in the navigation on the left and select **Open** to reach the Grafana UI.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

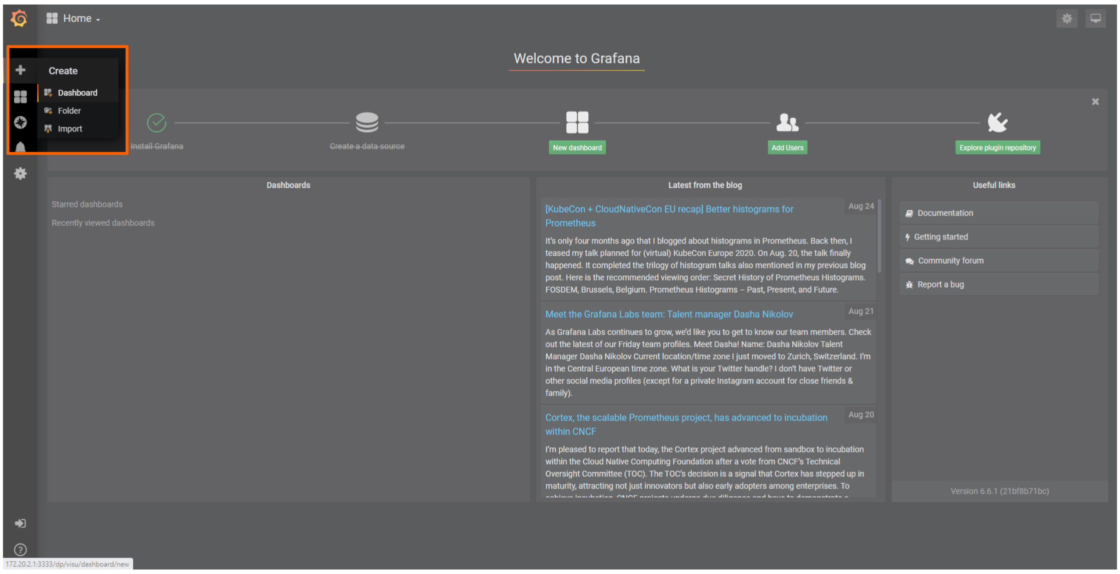
A new tab opens that shows the home screen of Grafana. Here dashboards can be created as described below.

Creating a dashboard

Dashboards need to be configured per data source. The instructions below cover the general workflow. Refer to Examples for specific use cases.

1. Access the visualization element on either the node or in the Management System.

2. Select **+ > Dashboard** in the navigation on the left. A box will appear.



[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus — Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

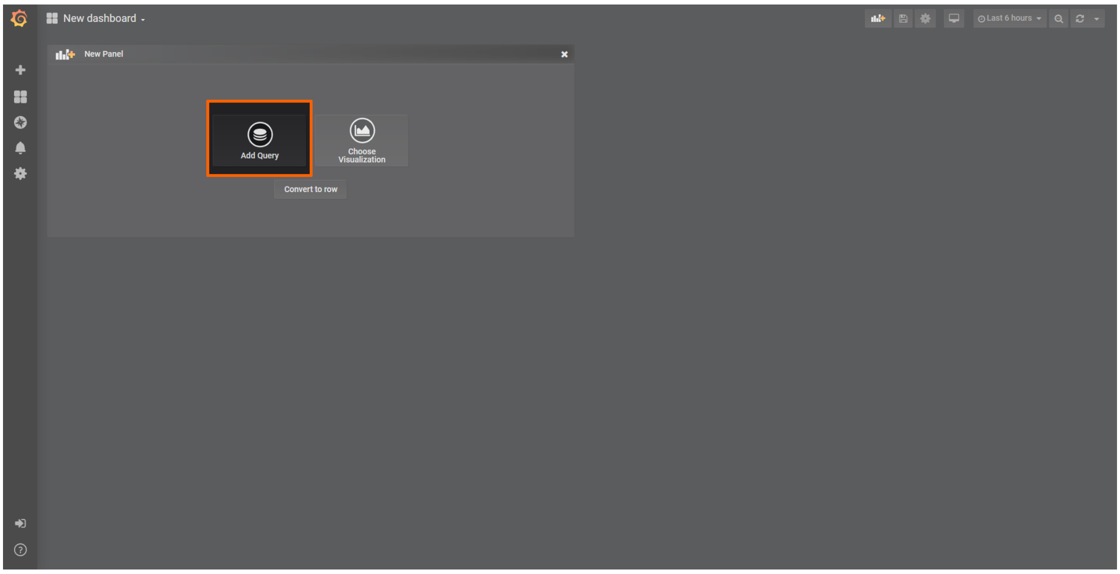
Meet the Grafana Labs team: Talent manager Dasha Nikolov — Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends & family).

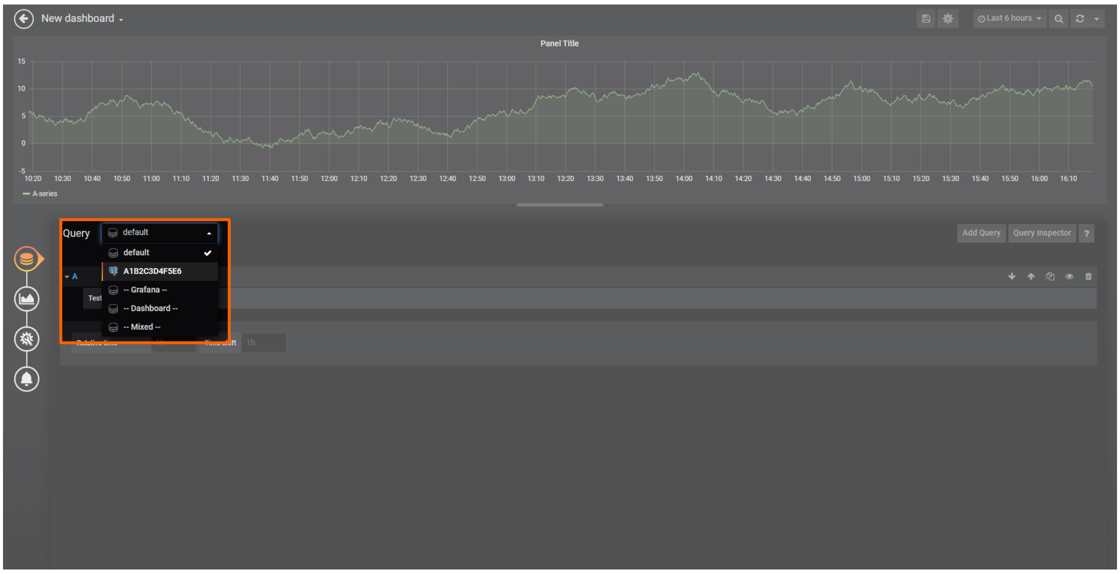
Cortex, the scalable Prometheus project, has advanced to incubation within CNCF — Aug 20

I'm pleased to report that today, the Cortex project advanced from sandbox to incubation within the Cloud Native Computing Foundation after a vote from CNCF's Technical Oversight Committee (TOC). The TOC's decision is a signal that Cortex has stepped up in maturity, attracting not just innovators but also early adopters among enterprises. To

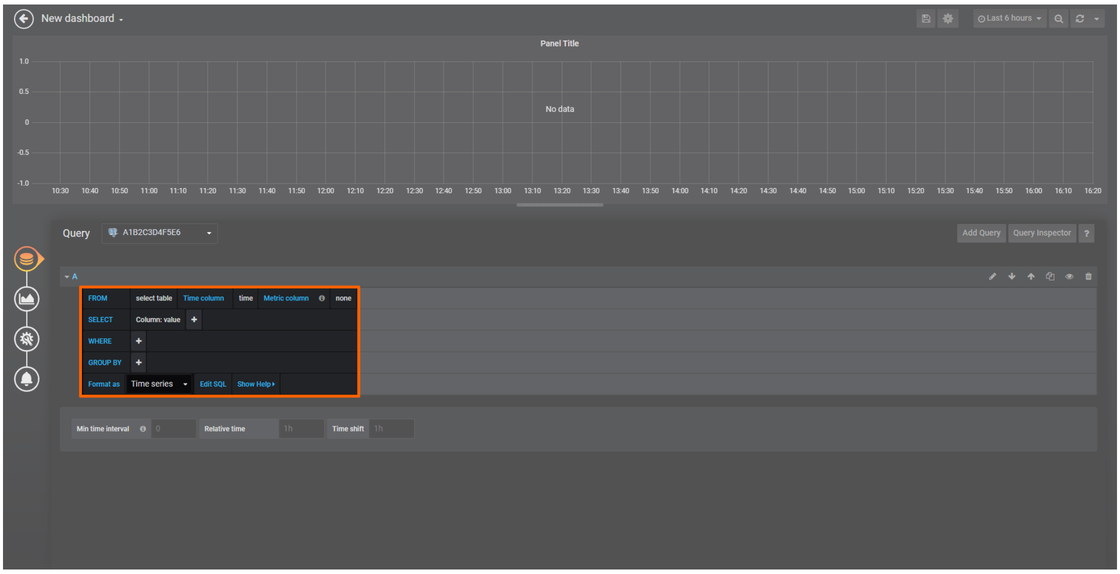
3. Select **Add Query** in the **New Panel** box.



4. Select the data source from the drop-down menu. The name of the data source is the serial number of the node.

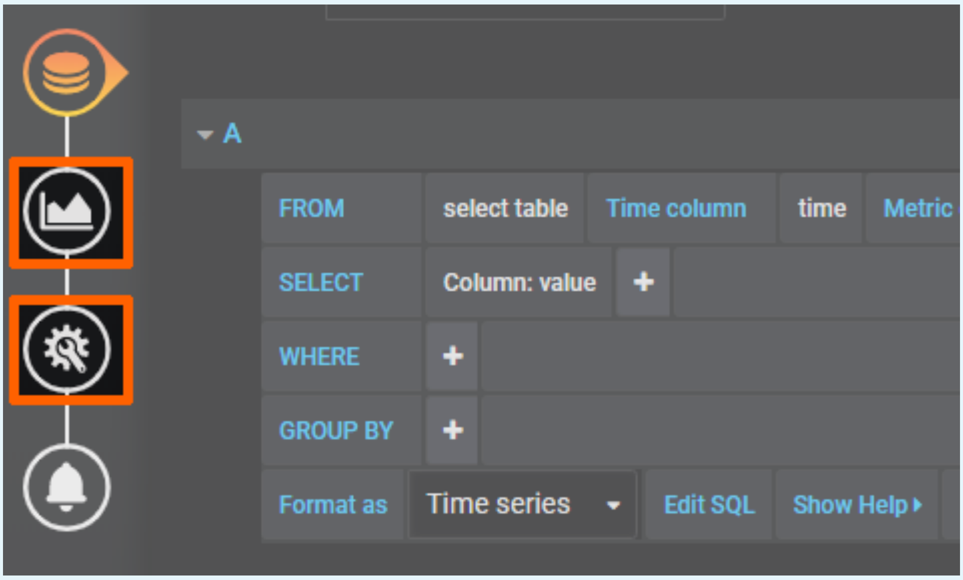


5. Fill in the query information below that appears below. This information depends on the use case and the Gateway configuration.

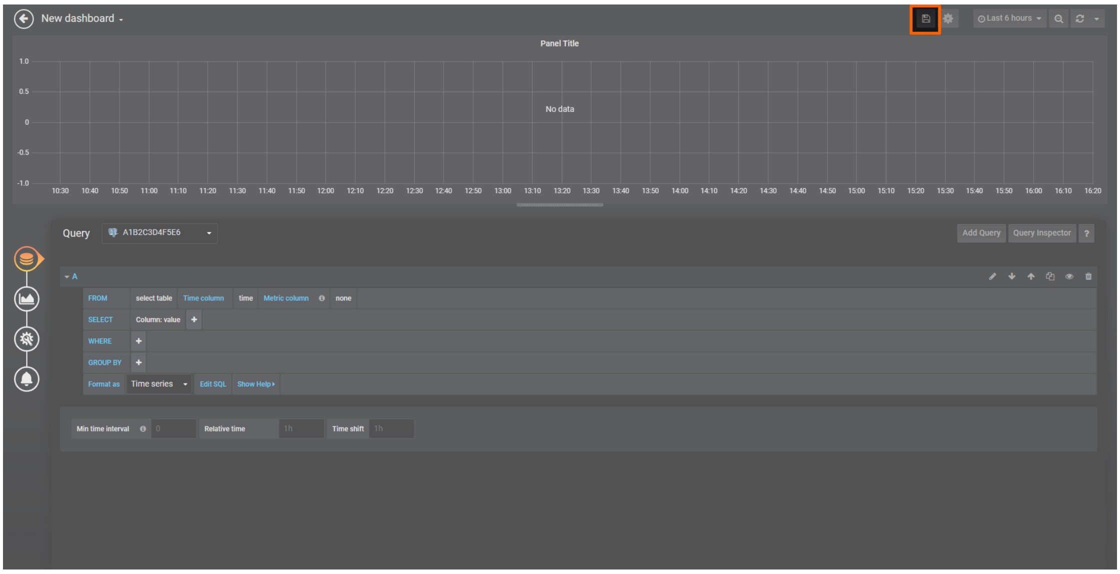


NOTE

Customize general settings and visualization settings by selecting **General** and **Visualization** in the navigation on the left. Refer to the Grafana documentation for more information.



6. Select the save icon in the upper-right corner to save the dashboard.



Query A1B2C3D4F5E6

FROM select table Time column time Metric column none

SELECT Column: value

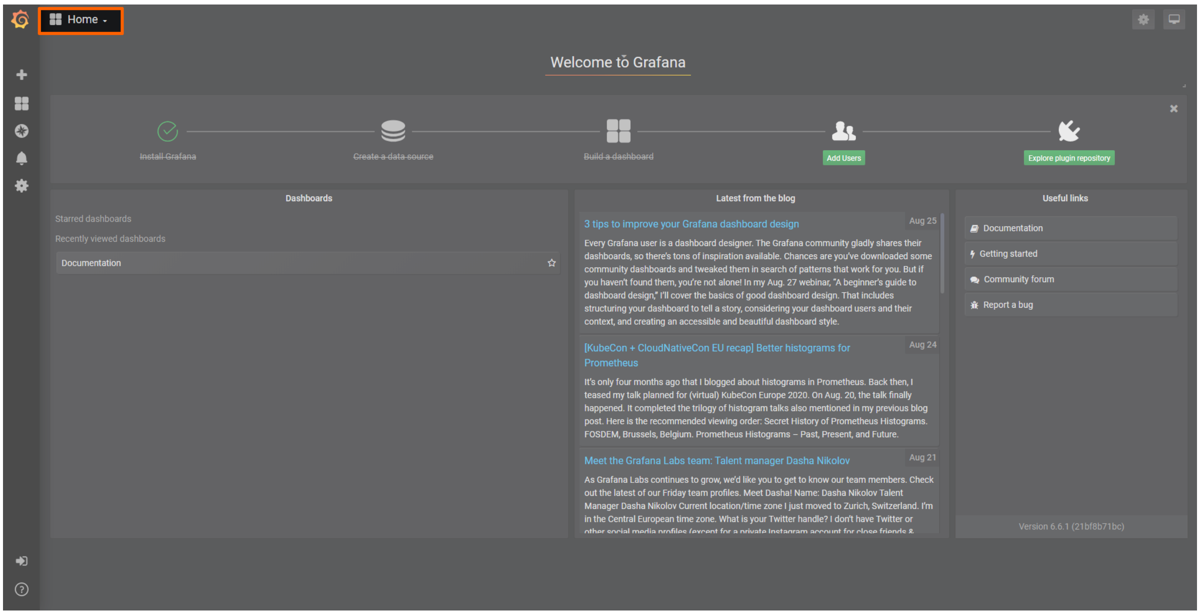
WHERE

GROUP BY

Format as Time series

Min time interval 0 Relative time 1h Time shift 1h

Access the dashboard from the home menu.



3 tips to improve your Grafana dashboard design Aug 25

Every Grafana user is a dashboard designer. The Grafana community gladly shares their dashboards, so there's tons of inspiration available. Chances are you've downloaded some community dashboards and tweaked them in search of patterns that work for you. But if you haven't found them, you're not alone! In my Aug. 27 webinar, "A beginner's guide to dashboard design," I'll cover the basics of good dashboard design. That includes structuring your dashboard to tell a story, considering your dashboard users and their context, and creating an accessible and beautiful dashboard style.

[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

Meet the Grafana Labs team: Talent manager Dasha Nikolov Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends &

For specific examples on how to use Grafana with Nerve Data Services, refer to Examples.

Data analytics

The analytics element of the Nerve Data Services is primarily used for processing and analysis of data collected by the Gateway, and provides an easy way to collect and store that data. It consist of a Software Development Kit (SDK) and an Application Programming Interface (API) both written in Python. Analytics are supported on all operating systems based on Debian Buster (tested on Ubuntu 18.04 LTS and Ubuntu 20.04 LTS). The recommended Python version to use when writing analytics apps is 3.7 or later.

The release package contains:

- API documentation
- an installation script
- the installation script README file

- the SDK python wheel (containing the API python wheel)

The Nerve Data SDK can be downloaded from the Nerve Software Center.

Software Development Kit

NOTE

The Software Development Kit can only be used on Linux and requires `sudo`, `python` and `pip` for the installation.

The Analytics Data SDK is used to create, build, push and provision analytics apps as Docker images. These functions will be explained in detail in the following sections. The SDK is installed using the installation script. Navigate to the folder containing the installation script and execute the following command:

```
source ./nerve_dp_analytics_install.sh
```

Follow the instructions in the installation script. It will ask for confirmation before installing the following requirements:

- Docker
  - docker-ce
  - curl
  - apt-transport-https
  - ca-certificates
  - gnupg-agent
  - software-properties-common
- Miniconda

It will also add an official Docker GPG key to the system and the official Docker stable repository to apt repositories. The Conda environment `nerve-dp-analytics` will be created and activated upon installation of the Analytics Data SDK and API wheels, as well as all their dependencies.

NOTE

Before working with the SDK, make sure that the Conda environment is active. If the Conda environment is active it will be displayed in parentheses in front. Activate the Conda environment by entering the following command:

```
source miniconda/bin/activate <environmentname>
```

The Conda environment automatically deactivates after a restart.

Create command

The create command is used to initialize the environment for analytics app development.

Item	Description
Usage	`nerve-analytics create [-h] [-c [CLONE]] [-t {minimal,slim,standard,intel,custom}] app_name path`

Item	Description

Positional arguments	`app_name`
	The application name.
	
	`path`
	Path where to create the application.

Optional arguments	`-h` or `--help`
	Use this argument to show the help message in the command line.
	
	`-c [CLONE]` or `--clone [CLONE]`
	Clones base Nerve analytics environment into a new environment. If an environment name is not passed with the argument, the new environment name will be auto-generated.
	
	`-t {minimal,slim,standard,intel,custom}` or `--type {minimal,slim,standard,intel,custom}`
	Defines the project type to be created.

Using the create command will create a directory named after the app name. Inside, it will create a Dockerfile, a Python script, and add the analytics API wheel. If `type` is not provided, `standard` will be used.

Build command

The build command is used to build the analytics app Docker image.

Item	Description
Usage	`nerve-analytics build [-h] [-t TAG] [-p PATH]`

Optional arguments	`-h` or `--help`
	Use this argument to show the help message in the command line.
	
	`-t TAG` or `--tag TAG`
	Name of the Docker tag that will be used when tagging the Docker image.
	
	`-p PATH` or `--path PATH`
	Path to the application folder.

If `tag` is not provided, `latest` will be used. If `path` is not provided, the build command will look for a Dockerfile in the current directory.

Push command

The push command is used to push the previously built Docker image to a Docker registry.

Item	Description
Usage	`nerve-analytics push [-h] [-t TAG] [-r RE_TAG] [-n NEW_NAME] [-u USERNAME] [-e EMAIL] [-p PASSWORD] reg_path`
Positional arguments	`reg_path` Path of the Docker registry.

Item	Description

Optional arguments	`-h` or `--help`
	Use this argument to show the help message in the command line.
	
	`-t TAG` or `--tag TAG`
	Tag name of the Docker image to be pushed.
	
	`-r RE_TAG` or `--re-tag RE_TAG`
	Define tag name of the Docker image in the registry.
	
	`-n NEW_NAME` or `--name NEW_NAME`
	New name of the repository.
	
	`-u USERNAME` or `--username USERNAME`
	Username which is used to log in into registry.
	
	`-e EMAIL` or `--email EMAIL`Email which is used to log in into the registry.
	
	`-p PASSWORD` or `--password PASSWORD`
	Password which is used to log in into the registry.

If `tag` is not provided, `latest` will be used. If `password` is not provided, a prompt to enter it will be displayed to the user.

Provision command

The provision command is used to upload the previously built Docker image as a Docker workload to a Nerve Management System. The Docker image is also archived as a TAR.GZ file.

Item	Description

Usage	`nerve-analytics provision [-h] [-f CFG_FILE] -u URL -vn VERSION_NAME -rn RELEASE_NAME [-usr USERNAME] [-pas PASSWORD] [-n NAME] [-i IMAGE] [-d DEPLOYED_CONTAINER_NAME] [-net NETWORKS] [-desc DESCRIPTION] [-p PORT] [-e ENV] [-m MEM_LIMIT] [-cl CPU_LIMIT] [-r {no,on-failure,always,unless-stopped}] [-rel] [--verbose]`
	
	Arguments that start with `--` (eg. `--url`) can also be set in a config file (specified via `-f <filepath>`). The config file syntax accepts the following separated by line breaks:
	
	• key=value
	• flag=true
	• stuff={a,b,c}
	
	Example:
	`name=test-workload`
	`verbose=true`
	`restart={on-failure}`
	
	For details on the syntax refer to https://pypi.org/project/ConfigArgParse/. If an argument is specified in more than one place, then the command line values override the config file values which override the defaults.

Optional arguments

`-h` or `--help`
Use this argument to show the help message in the command line.

`-f CFG_FILE` or `--cfg-file CFG_FILE`
Configuration file for the provision command.

`-u URL` or `--url URL`
URL of the Nerve Management System to which the workload will be uploaded. This flag is mandatory.

`-vn VERSION_NAME` or `--version-name VERSION_NAME`
Version name of the Docker workload. This flag is mandatory.

`-rn RELEASE_NAME` or `--release-name RELEASE_NAME`
Release name of the Docker workload. This flag is mandatory.

`-usr USERNAME` or `--username USERNAME`
Username which is used to log in to the Nerve Management System.

`-pas PASSWORD` or `--password PASSWORD`
Password which is used to log in to the Nerve Management System.

`-n NAME` or `--name NAME`
Name of the Docker workload which will be uploaded to the Nerve Management System.

`-i IMAGE` or `--image IMAGE`
This is the of the Docker image that has been created with the create and build commands and will be provisioned to the Nerve Management System.

`-d DEPLOYED_CONTAINER_NAME` or `--deployed-container-name DEPLOYED_CONTAINER_NAME`
Name of the Docker container upon deployment to a node.

`-net NETWORKS` or `--networks NETWORKS`
Docker network names. This is set to `nerve-dp` by default if not specified.

`-desc DESCRIPTION` or `--description DESCRIPTION`
Description for the Docker workload to be uploaded.

`-p PORT` or `--port PORT`
Port mapping for the container to be deployed, for example `5432:5432` or `123:234/udp`.

`-e ENV` or `--env ENV`
Environment variables for the container to be deployed, for example `VAR1=myval`.

`-m MEM_LIMIT` or `--mem-limit MEM_LIMIT`
Memory limit in MB for the workload on the node.

`-cl CPU_LIMIT` or `--cpu-limit CPU_LIMIT`
Max number of CPUs to be used by the workload on the node. 1 by default.

`-r {no,on-failure,always,unless-stopped}` or `--restart {no,on-failure,always,unless-stopped}`
Restart policy for the container to be deployed on the node.

`-rel` or `--released`
This sets the workload version to released. Released versions cannot be updated.

`--verbose`
Print workload information.

The user that is used to provision the analytics app must already exist in the Management System. All parameters used when uploading a workload from the Management System can also be specified in the provision command.

If `password` is not provided, a prompt to enter the password will be displayed. If a workload with the same `name` already exists in the Management System, it must be a Docker workload in order to add a new version to it. Otherwise, the provisioning cannot be done. When adding a new version to an existing Docker workload in the Management System, `version` must be different than that of the existing version.

Application Programming Interface

The Analytics API is used for writing Python applications that collect and store data from and to other elements of the Data Services. The recommended Python version to use when writing analytics apps is 3.7 or later. The Analytics API provides the following modules:

Module	Description
Batch Input TimescaleDB	Used to fetch data from a TimescaleDB database.
Batch Output TimescaleDB	Used to store data in a TimescaleDB database.
Stream Input ZeroMQ	ZeroMQ Subscriber used to collect data from a ZeroMQ Publisher. Can work in both asynchronous and synchronous modes. **NOTE** In order to connect the ZeroMQ Publisher output of the Gateway and the Stream Input ZeroMQ of the analytics, the IP address `172.20.10.1` must be used. This is the case only if the analytics container is running inside the Docker network `nerve-dp`. If the Docker network is `host`, `localhost` can be used.
Stream Output MQTT	MQTT Publisher used to send data to a MQTT broker.

More in depth documentation regarding the API modules can be found in the release package of the analytics element that has been downloaded from the Nerve Software Center.

Examples

Gateway configuration examples

This chapter covers use cases for the supported protocols of the Nerve Data Services, showcasing the usage of each Data Services element in the process. This section is subject to change and further examples will be added in the near future. Currently available examples are:

- OPC UA Server to cloud
- S7 Client to cloud
- MQTT Publisher to OPC UA Server at the node
- Receiving data via MQTT for Analytics and Visualization

OPC UA Server to cloud for visualization

In this example, a sensor is providing temperature and humidity data as an OPC UA Server. Temperature data is displayed at the node while humidity data is presented at the Management System. Data is visualized with the visualization element of the Data Services.

The instructions below cover the following steps:

- Provisioning an OPC UA Server demoSensor as a Docker workload
- Deploying the provisioned Docker workload to the target node
- Configuring the Data Services Gateway on the node
- Local data visualization at the node
- Configuring the Data Services Gateway in the Management System
- Central data visualization in the Management System

Provisioning and deploying an OPC UA Server at the node

First, the temperature sensor simulation OPC UA Server must be deployed to the node as a Docker workload. Download the **Data Services OPC UA demo sensor** found under **Example Applications** from the Nerve Software Center. This is the Docker image that is required for provisioning the demo sensor as a Docker workload.

1. Log in to the Management System. Make sure that the user has the permissions to access the Data Services.

2. Provision a Docker workload by following Provisioning a Docker workload. This example uses **TTTech OPC UA Server - demoSensor** as the workload name. Use the following workload version settings:

Setting	Value
Name	Enter any name for the workload version.
Release name	Enter any release name.
DOCKER IMAGE	Select **Upload** to add the Docker image of the temperature sensor simulation that has been downloaded before.
DOCKER SPECIFIC INFO	Select **New port** and enter the following settings: ◦ **Protocol**: TCP ◦ **Host Port**: 4848 ◦ **Container Port**: 4848
Container name	`tttech-opcua-server-demosensor`
Network name	`bridge`

3. Deploy the provisioned Docker workload by following Deploying a workload.

> **NOTE**
>
> Remember the node name and serial number of the target node. They are needed for the JSON configuration.

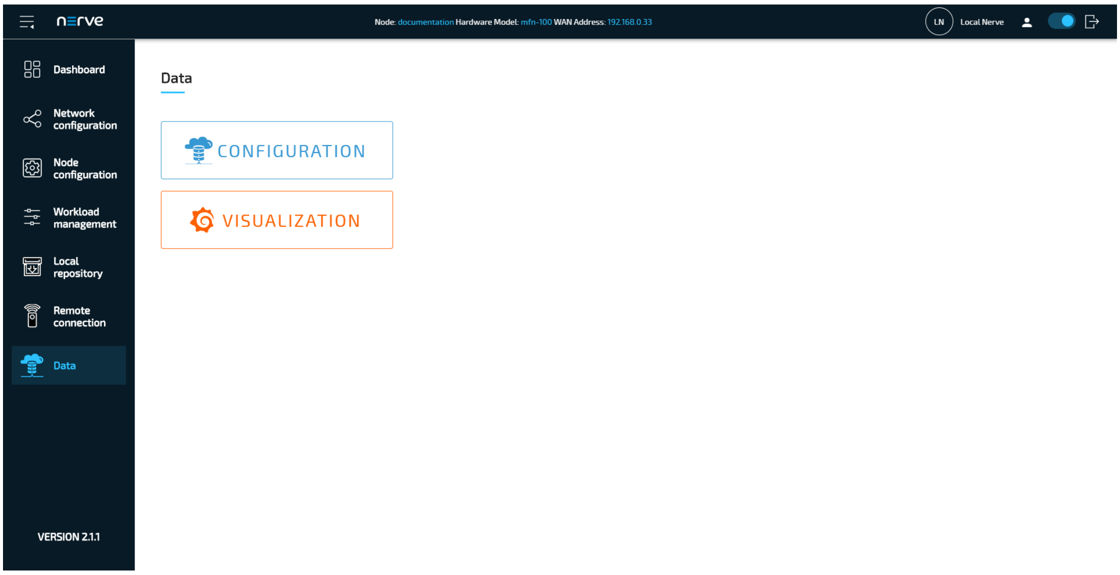
Configuring the Data Services Gateway on the node

With the OPC UA Server deployed, the Gateway instance on the node must be configured next.

1. Access the Local UI on the node. This is Nerve Device specific. Refer to the table below for device specific links to the Local UI. The login credentials to the Local UI can be found in the customer profile.

Nerve Device	Physical port	Local UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

2. Select **Data** in the navigation on the left.

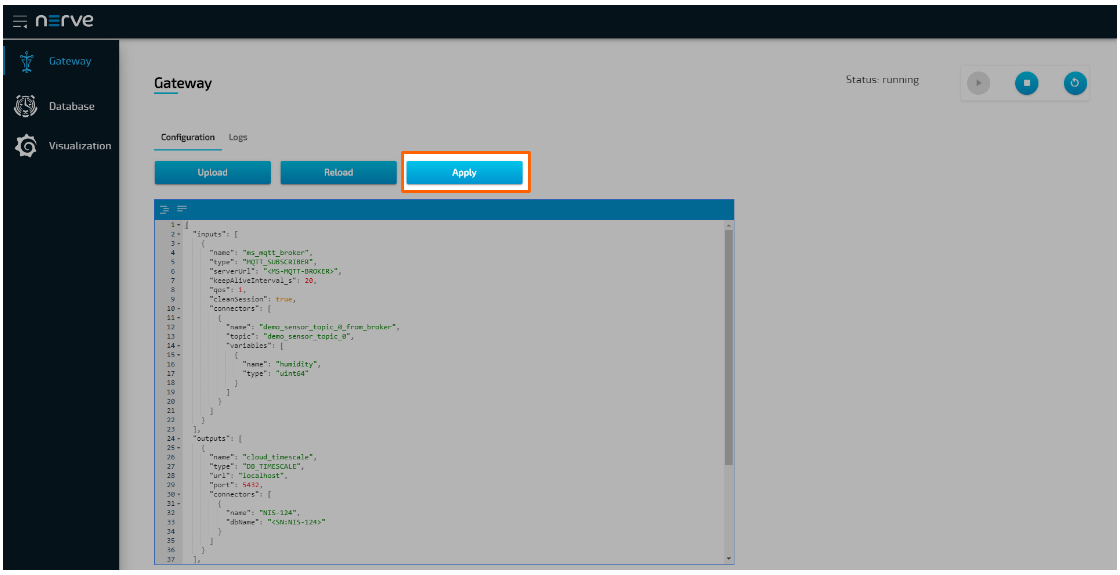


3. Select **CONFIGURATION**.

4. Enter the following configuration into the configuration editor of the Gateway:

```
{
  "inputs": [
    {
      "name": "client_demo_sensor",
      "type": "OPC_UA_CLIENT",
      "serverUrl": "opc.tcp://localhost:4848",
      "pollingInterval_ms": 1000,
      "connectors": [
        {
          "name": "poll",
          "accessType": "polling",
          "nodes": [
            "ns=2;i=2"
          ]
        },
        {
          "name": "subs",
          "accessType": "subscription",
          "publishingIntervalAtServer_ms": 1000,
          "samplingIntervalAtServer_ms": 1000,
          "nodes": [
            "ns=2;i=4"
          ]
        }
      ]
    }
  ],
  "outputs": [
    {
      "name": "local_timescale",
      "type": "DB_TIMESCALE",
      "url": "<LOCAL>"
    },
    {
      "name": "mqtt_pub_to_mgmtsys",
      "type": "MQTT_PUBLISHER",
      "serverUrl": "wss://<MGMT-SYS>:443/dp/mqtt/wss",
      "username": "<SN>",
      "password": "<SID>",
      "keepAliveInterval_s": 20,
      "qos": 0,
      "sslOptions": {
        "serverAuthRequired": false
      },
      "cleanSession": true,
      "connectors": [
        {
          "name": "demo_sensor_topic_0_to_mgmtsys",
          "topic": "demo_sensor_topic_0",
          "timestampFormat": "iso",
          "timestampRequired": true
        }
      ]
    }
  ],
  "connections": [
    {
      "name": "demo_sensor_to_local_db",
      "input": {
        "index": 0,
        "connector": 0
      },
      "output": {
        "index": 0,
        "connector": 0
      }
    },
    {
      "input": {
        "index": 0,
        "connector": 1
      },
      "name": "demo_sensor_to_mgmtsys",
      "output": {
        "index": 1,
        "connector": 0
      }
    }
  ]
}
```

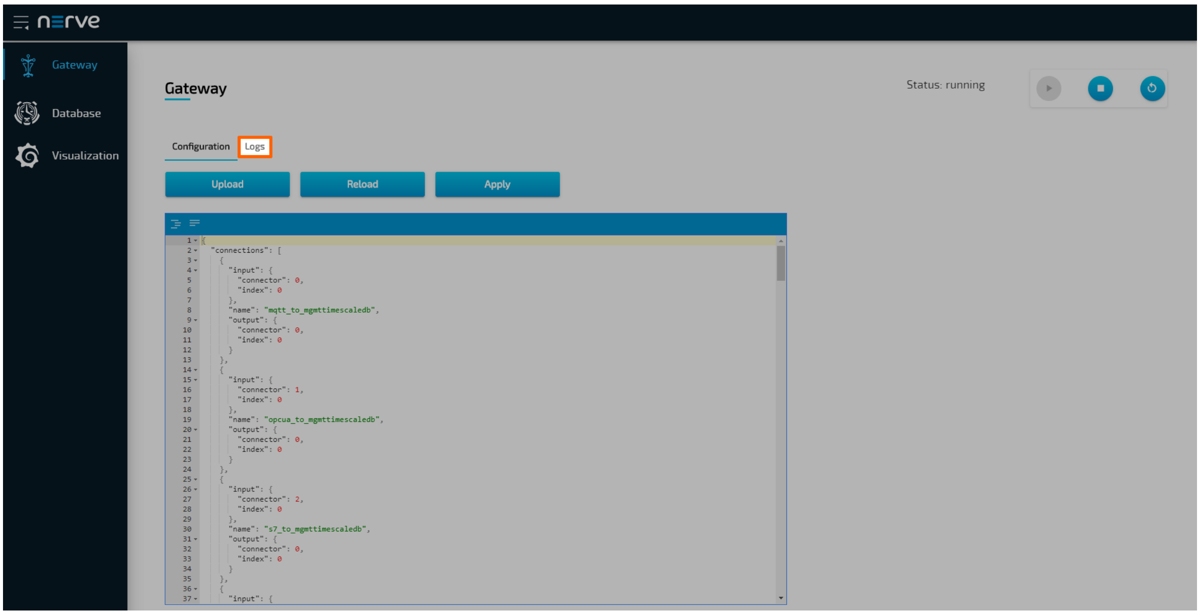
5. Select **Apply** to save the Gateway configuration. The Gateway will restart automatically.



```
{
  "inputs": [
    {
      "name": "ms_mqtt_broker",
      "type": "MQTT_SUBSCRIBER",
      "serverUrl": "<MS-MQTT-BROKER>",
      "keepAliveInterval_s": 20,
      "qos": 1,
      "cleanSession": true,
      "connectors": [
        {
          "name": "demo_sensor_topic_0_from_broker",
          "topic": "demo_sensor_topic_0",
          "variables": [
            {
              "name": "humidity",
              "type": "uint64"
            }
          ]
        }
      ]
    }
  ],
  "outputs": [
    {
      "name": "cloud_timescale",
      "type": "DB_TIMESCALE",
      "url": "localhost",
      "port": 5432,
      "connectors": [
        {
          "name": "NIS-124",
          "dbName": "<SN:NIS-124>"
        }
      ]
    }
  ],
```

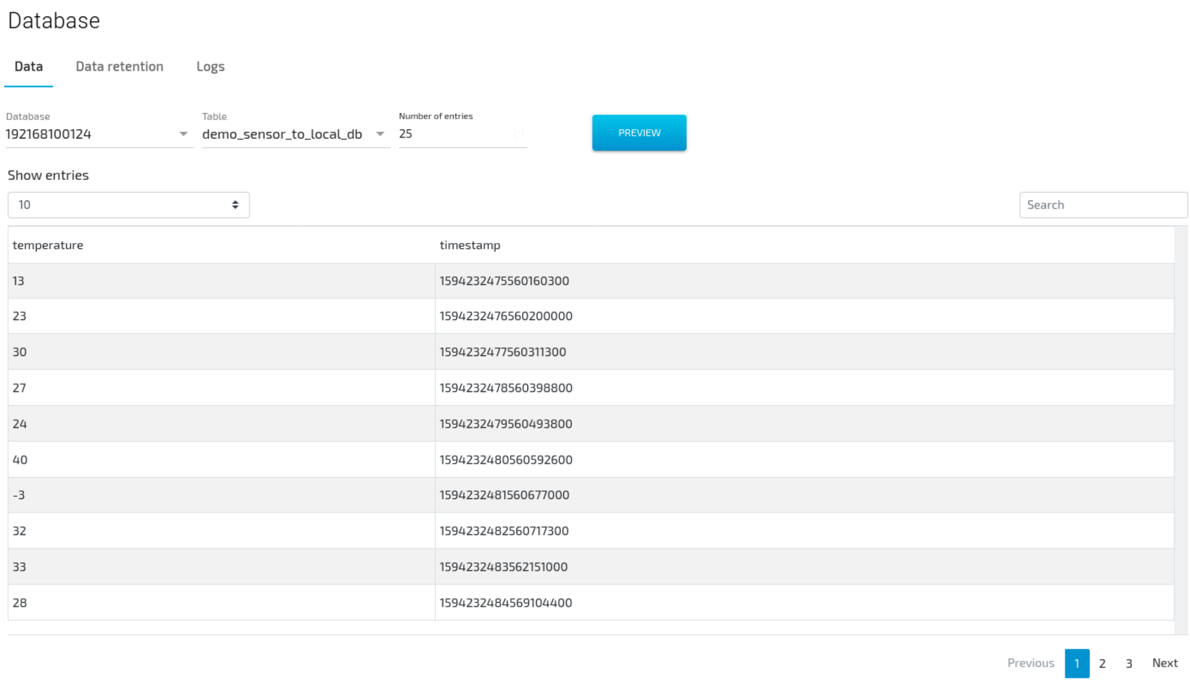
The configuration defines the OPC UA Server demoSensor that was deployed earlier as a data input. The local Timescale DB storage and the MQTT broker in the Management System are defined as data outputs. An OPC UA Client connects to the OPC UA Server demoSensor and periodically reads the value of temperature and humidity. Temperature values are forwarded to the storage on the node and humidity values are sent to the Management System via MQTT.

To verify that the Gateway accepted the configuration and works as expected, select the **Logs** tab in the Gateway UI and check for errors.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

To verify if data is stored in the database, select **Database** in the navigation on the left. Select the serial number of the node under **Database** and **demo_sensor_to_local_db** under **Table**. Select **Preview** to display the data stored in the database as a table.



Database: 192168100124

Table: demo_sensor_to_local_db

Number of entries: 25

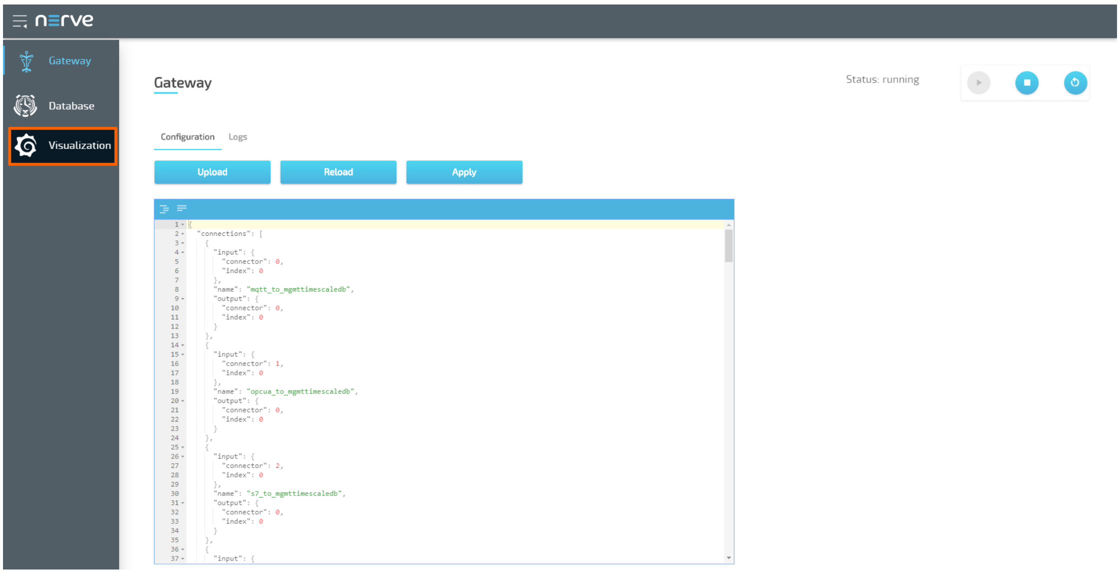
Show entries: 10

temperature	timestamp
13	1594232475560160300
23	1594232476560200000
30	1594232477560311300
27	1594232478560398800
24	1594232479560493800
40	1594232480560592600
-3	1594232481560677000
32	1594232482560717300
33	1594232483562151000
28	1594232484569104400

Local data visualization at the node

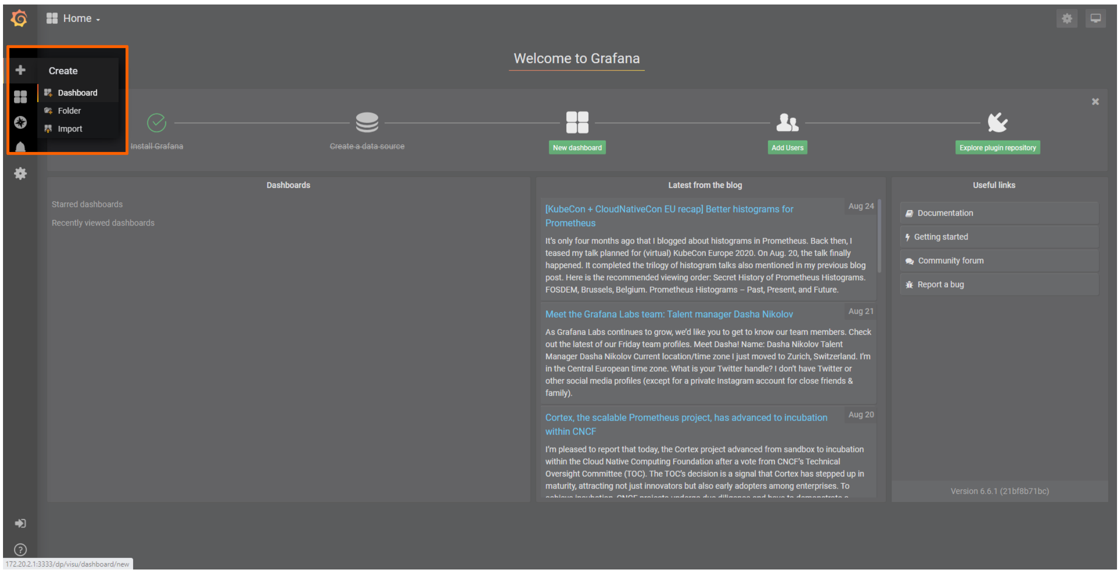
To visualize temperature data at the node, open the local data visualization element through the Data Services UI on the node.

1. Select **Visualization** in the navigation on the left. The Grafana UI will open.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

2. Select **+ > Dashboard** in the navigation on the left. A box will appear.



[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus — Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

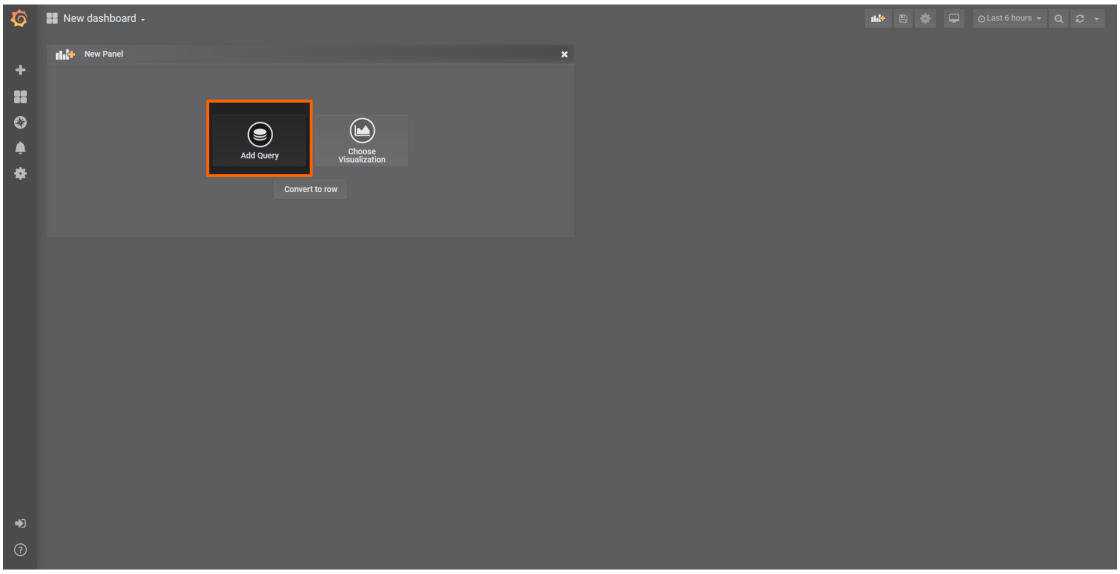
Meet the Grafana Labs team: Talent manager Dasha Nikolov — Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends & family).

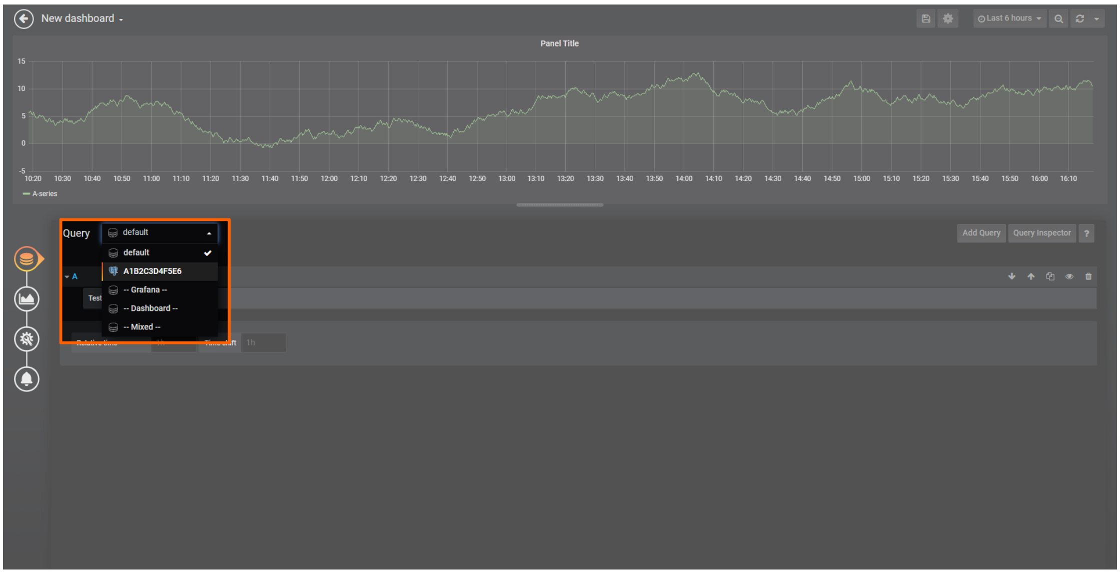
Cortex, the scalable Prometheus project, has advanced to incubation within CNCF — Aug 20

I'm pleased to report that today, the Cortex project advanced from sandbox to incubation within the Cloud Native Computing Foundation after a vote from CNCF's Technical Oversight Committee (TOC). The TOC's decision is a signal that Cortex has stepped up in maturity, attracting not just innovators but also early adopters among enterprises. To

3. Select **Add Query** in the **New Panel** box.



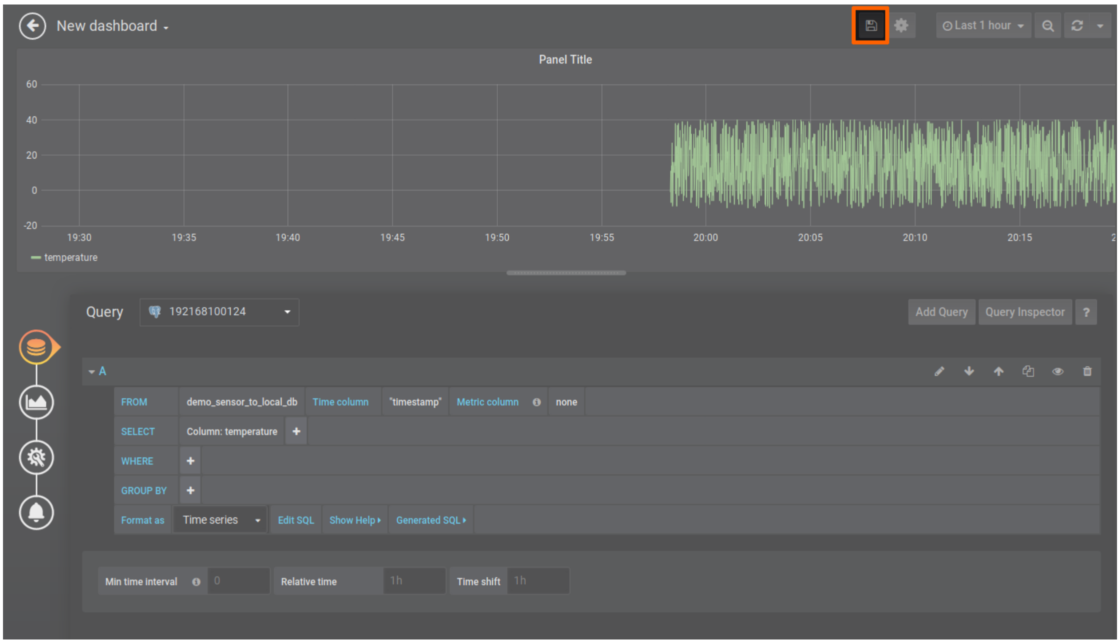
4. Select the data source from the drop-down menu. The name of the data source is the serial number of the node.



5. Fill in the following query information:

Setting	Value
FROM	`demo_sensor_to_local_db` **Time column**: "timestamp"
SELECT	Column: temperature
Format as	Time series

6. Select the save icon in the upper-right corner to save the dashboard.



Query 192168100124

A

FROM demo_sensor_to_local_db Time column "timestamp" Metric column none

SELECT Column: temperature

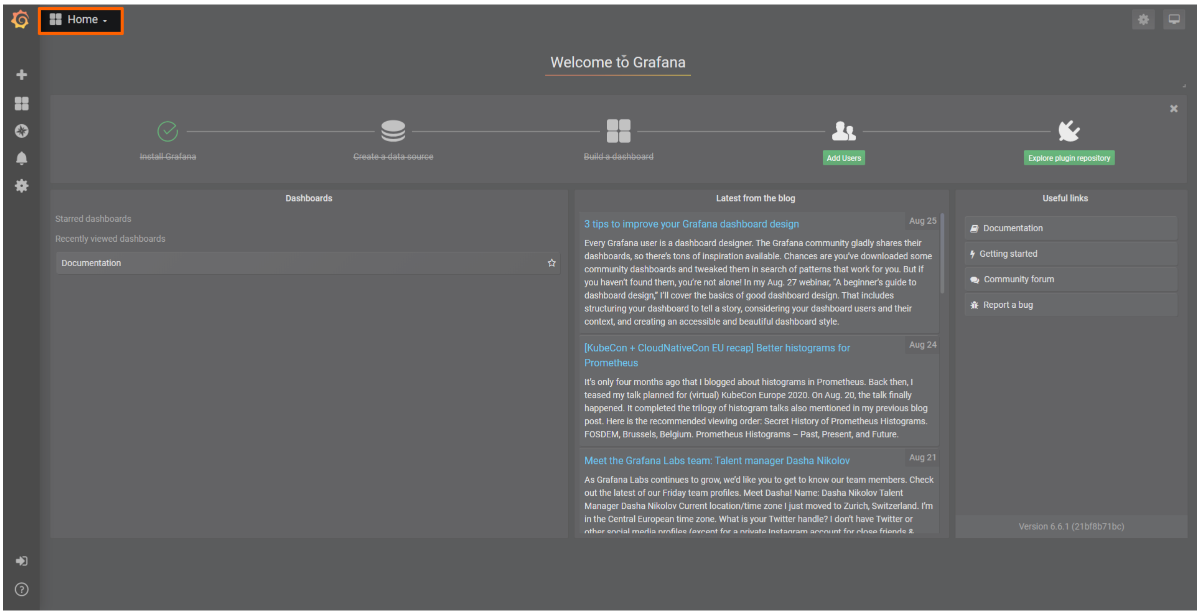
WHERE

GROUP BY

Format as Time series Edit SQL Show Help Generated SQL

Min time interval 0 Relative time 1h Time shift 1h

The dashboard can be accessed from the Grafana home menu.



3 tips to improve your Grafana dashboard design — Aug 25

Every Grafana user is a dashboard designer. The Grafana community gladly shares their dashboards, so there's tons of inspiration available. Chances are you've downloaded some community dashboards and tweaked them in search of patterns that work for you. But if you haven't found them, you're not alone! In my Aug. 27 webinar, "A beginner's guide to dashboard design," I'll cover the basics of good dashboard design. That includes structuring your dashboard to tell a story, considering your dashboard users and their context, and creating an accessible and beautiful dashboard style.

[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus — Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

Meet the Grafana Labs team: Talent manager Dasha Nikolov — Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends &

Configuring the Data Services Gateway in the Management System

At this point the Gateway of the node is handling two data flows:

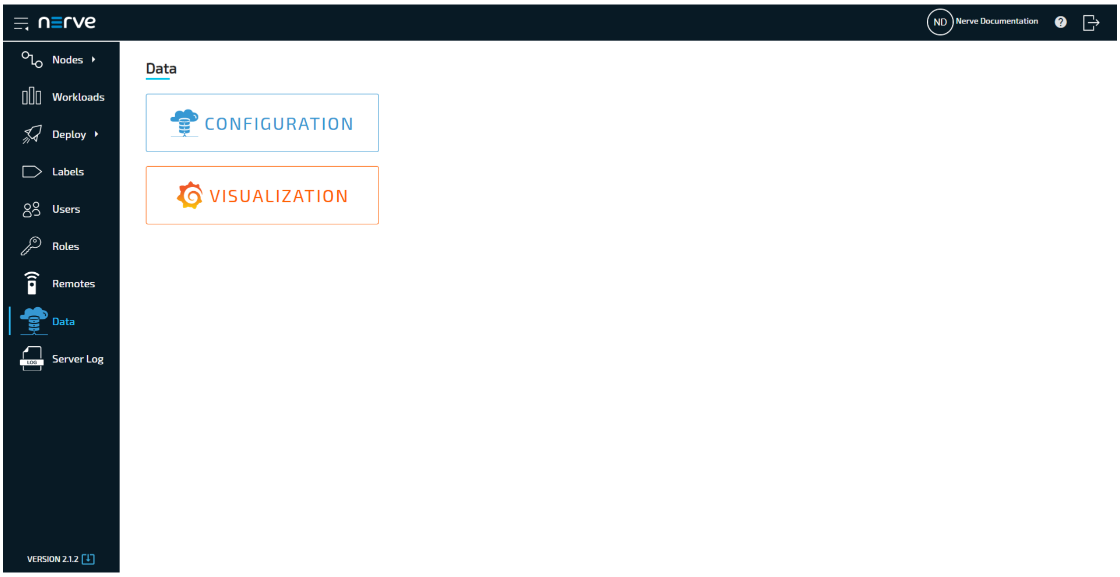
- Temperature data from the demoSensor to the TimescaleDB at the node.
- Humidity data to the MQTT broker of the Management System.

The next step is collecting and visualizing humidity data at the Management System. The example configuration in the instructions below connects the input of the Management System Gateway to the MQTT broker of the Management System, and transfers data through its output into the TimescaleDB in the Management System.

1. Log in to the Management System

2. Select **Data** in the navigation on the left. Make sure that the user has the necessary permissions to access the Data Services.

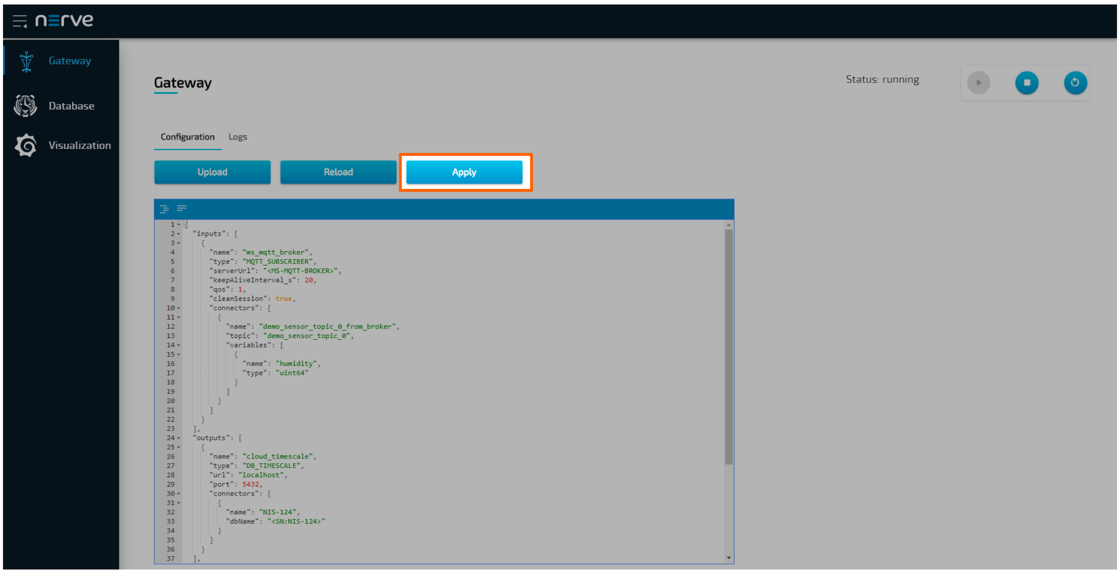
3. Select **CONFIGURATION** to access the Data Services UI.



4. Enter the following configuration into the configuration editor of the Gateway. Replace `<nodename>` in the configuration below with the name of the node that the Docker workload has been deployed to earlier.

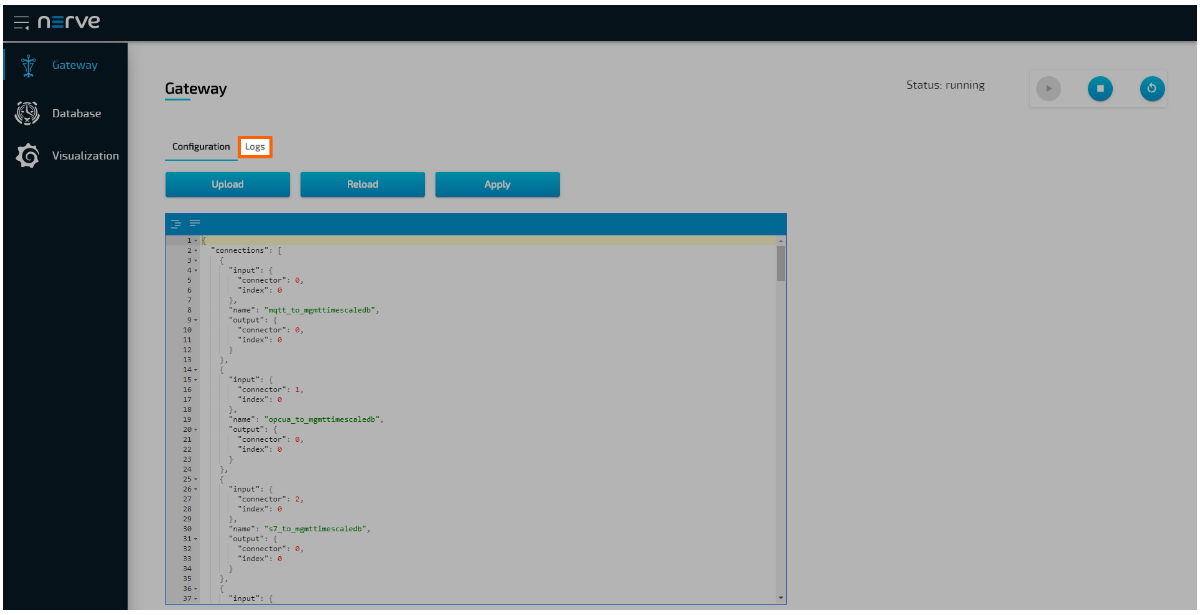
```
{
  "inputs": [
    {
      "name": "ms_mqtt_broker",
      "type": "MQTT_SUBSCRIBER",
      "serverUrl": "<MS-MQTT-BROKER>",
      "keepAliveInterval_s": 20,
      "qos": 1,
      "cleanSession": true,
      "connectors": [
        {
          "name": "demo_sensor_topic_0_from_broker",
          "topic": "demo_sensor_topic_0",
          "variables": [
            {
              "name": "humidity",
              "type": "uint64"
            }
          ]
        }
      ]
    }
  ],
  "outputs": [
    {
      "name": "cloud_timescale",
      "type": "DB_TIMESCALE",
      "url": "localhost",
      "port": 5432,
      "connectors": [
        {
          "name": "<nodename>",
          "dbName": "<SN:<nodename>>"
        }
      ]
    }
  ],
  "connections": [
    {
      "name": "ms_mqtt_broker_to_cloud_timescale_db",
      "input": {
        "index": 0,
        "connector": 0
      },
      "output": {
        "index": 0,
        "connector": 0
      }
    }
  ]
}
```

5. Select **Apply** to save the Gateway configuration. The Gateway will restart automatically.



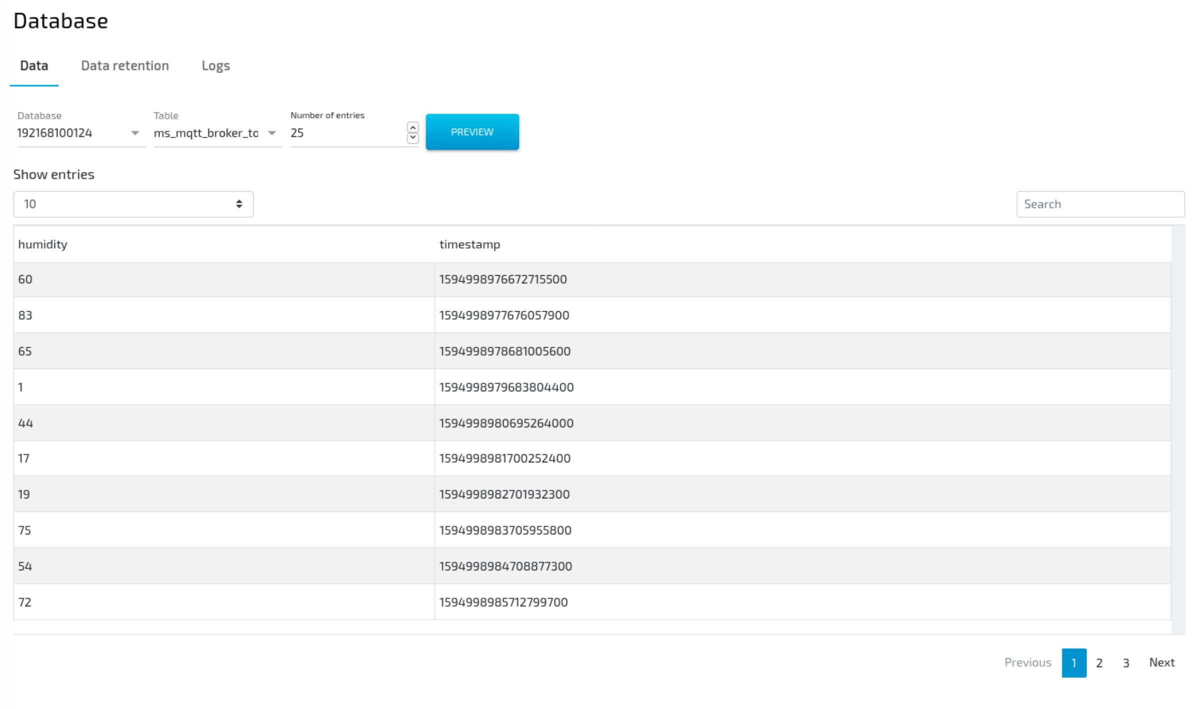
```
{
  "inputs": [
    {
      "name": "ms_mqtt_broker",
      "type": "MQTT_SUBSCRIBER",
      "serverUrl": "<MS-MQTT-BROKER>",
      "keepAliveInterval_s": 20,
      "qos": 1,
      "cleanSession": true,
      "connectors": [
        {
          "name": "demo_sensor_topic_0_from_broker",
          "topic": "demo_sensor_topic_0",
          "variables": [
            {
              "name": "humidity",
              "type": "uint64"
            }
          ]
        }
      ]
    }
  ],
  "outputs": [
    {
      "name": "cloud_timescale",
      "type": "DB_TIMESCALE",
      "url": "localhost",
      "port": 5432,
      "connectors": [
        {
          "name": "NIS-124",
          "dbName": "<SN:NIS-124>"
        }
      ]
    }
  ],
```

To verify that the Gateway accepted the configuration and works as expected, select the **Logs** tab and check for errors.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

To verify if data is stored in the database, select **Database** in the navigation on the left. Select the serial number of the node under **Database** and **ms_mqtt_broker_to_cloud_timescale_db** under **Table**. Select **Preview** to display the data stored in the database as a table.



Database: 192168100124
Table: ms_mqtt_broker_to
Number of entries: 25

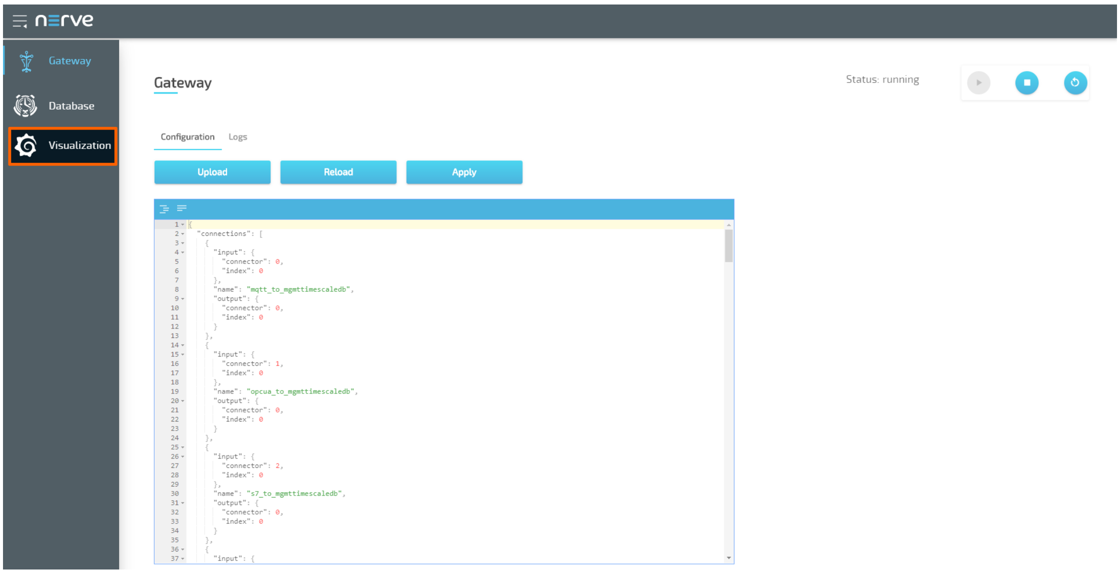
Show entries: 10

humidity	timestamp
60	1594998976672715500
83	1594998977676057900
65	1594998978681005600
1	1594998979683804400
44	1594998980695264000
17	1594998981700252400
19	1594998982701932300
75	1594998983705955800
54	1594998984708877300
72	1594998985712799700

Central data visualization in the Management System

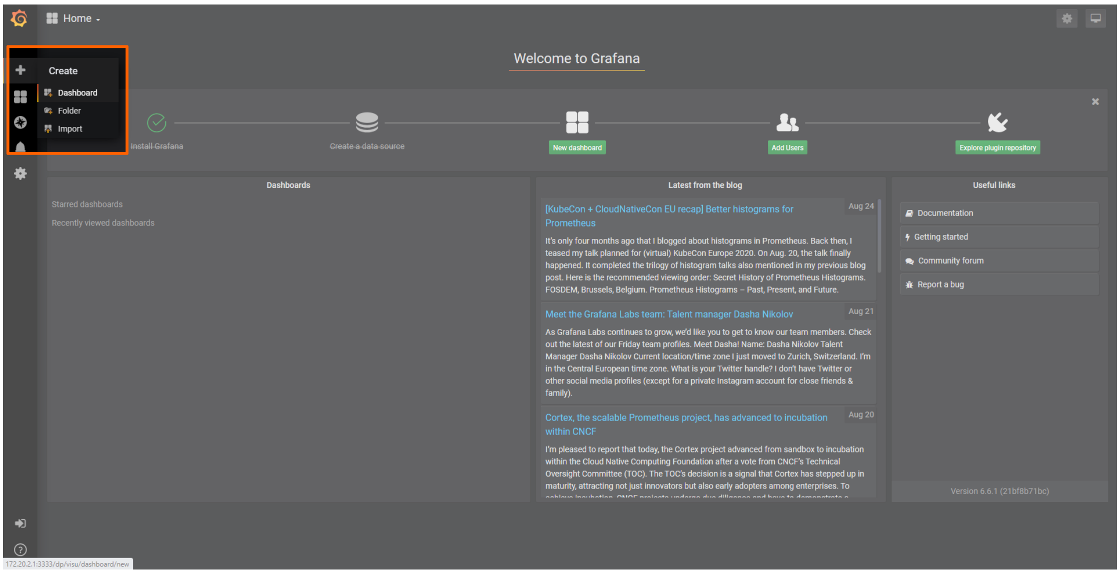
To visualize humidity data in the Management System, open the central visualization element through the Data Services UI in the Management System.

1. Select **Visualization** in the navigation on the left. The Grafana UI will open.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

2. Select **+ > Dashboard** in the navigation on the left. A box will appear.



[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

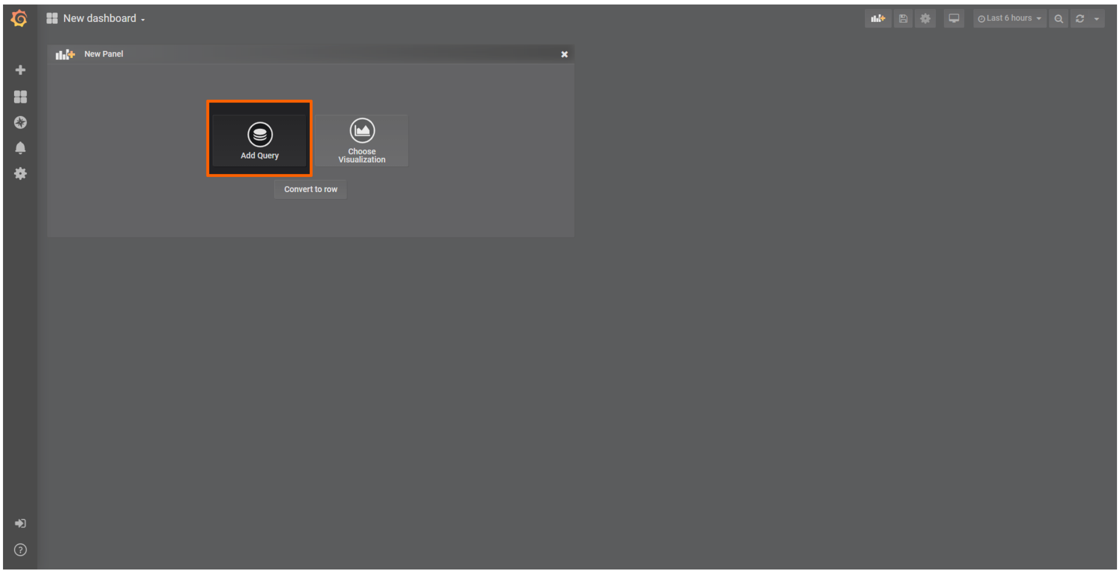
Meet the Grafana Labs team: Talent manager Dasha Nikolov Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends & family).

Cortex, the scalable Prometheus project, has advanced to incubation within CNCF Aug 20

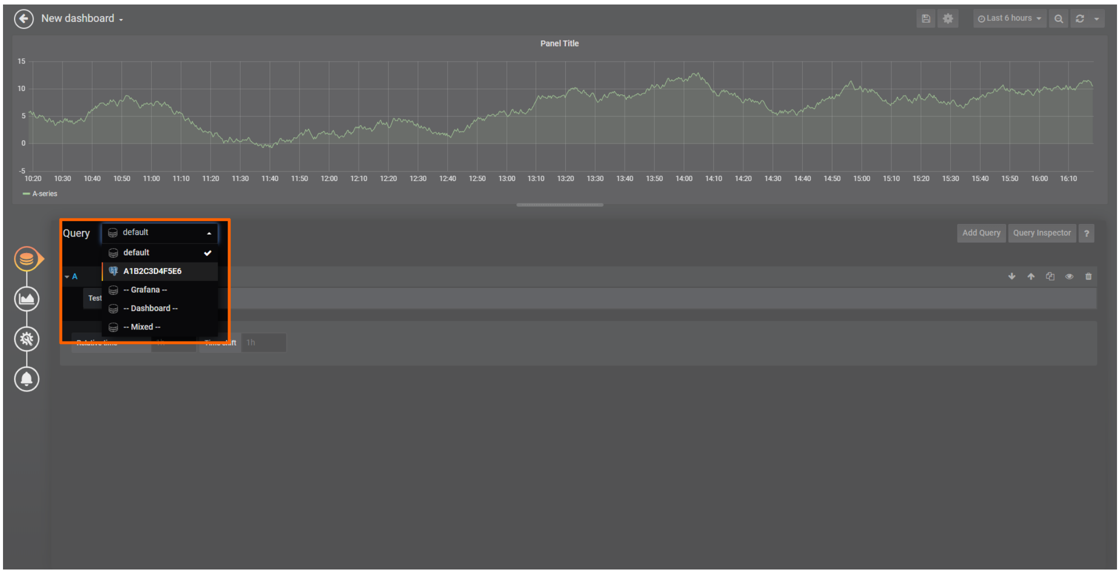
I'm pleased to report that today, the Cortex project advanced from sandbox to incubation within the Cloud Native Computing Foundation after a vote from CNCF's Technical Oversight Committee (TOC). The TOC's decision is a signal that Cortex has stepped up in maturity, attracting not just innovators but also early adopters among enterprises. To

3. Select **Add Query** in the **New Panel** box.



4. Select the data source from the drop-down menu. The name of the data source is the name and serial number of the node, formatted as `<nodename> (<serialnumber>)`.

> **NOTE**
>
> Note that multiple data sources can be selected in the Grafana instance of the Management System, depending on the number of nodes that are registered. Make sure to remember the serial number of the node that was used for workload deployment before.

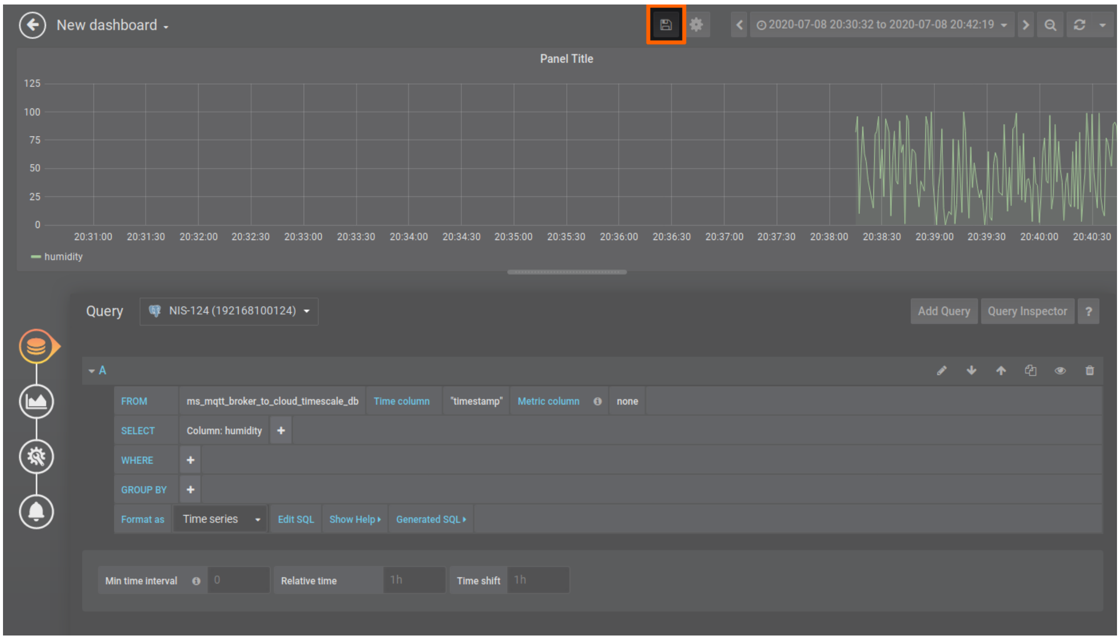


5. Fill in the following query information:

Setting	Value
FROM	`ms_mqtt_broker_to_cloud_timescale_db` **Time column**: "timestamp"
SELECT	Column: humidity

Setting	Value
Format as	Time series

6. Select the save icon in the upper-right corner to save the dashboard.



Query: NIS-124 (192168100124)

FROM ms_mqtt_broker_to_cloud_timescale_db Time column "timestamp" Metric column none

SELECT Column: humidity

WHERE

GROUP BY

Format as Time series

Min time interval 0

Relative time 1h

Time shift 1h

S7 Client to cloud for visualization

This example demonstrates how to read data from an S7 demo server that provides temperature, revolutions and active digital inputs of a motor connected to an S7 PLC. The data is then visualized with the Central Data Services in the Management System.

The instructions below cover the following steps:

- Provisioning an S7 demo server as a Docker workload
- Deploying the provisioned Docker workload to the target node
- Configuring the Data Services Gateway on the node
- Configuring the Data Services Gateway in the Management System
- Central data visualization in the Management System

Provisioning and deploying an OPC UA Server at the node

First, the S7 demo server must be deployed to the node as a Docker workload. Download the **Data Services S7 demo server** found under **Example Applications** from the Nerve Software Center. This is the Docker image that is required for provisioning the demo server as a Docker workload.

1. Log in to the Management System. Make sure that the user has the permissions to access the Data Services.

2. Provision a Docker workload by following Provisioning a Docker workload. This example uses **s7-demo** as the workload name. Use the following workload version settings:

Setting	Value
Name	Enter any name for the workload version.

Setting	Value
Release name	Enter any release name.
DOCKER IMAGE	Select **Upload** to add the Docker image of the S7 demo sensor that has been downloaded before.
DOCKER SPECIFIC INFO	Select **New port** and enter the following settings: ◦ **Protocol**: TCP ◦ **Host Port**: 102 ◦ **Container Port**: 102
Container name	`tttech-s7-server-demo`
Network name	`bridge`

3. Deploy the provisioned Docker workload by following Deploying a workload.

> **NOTE**
>
> Remember the node name and serial number of the target node. They are needed for the JSON configuration.

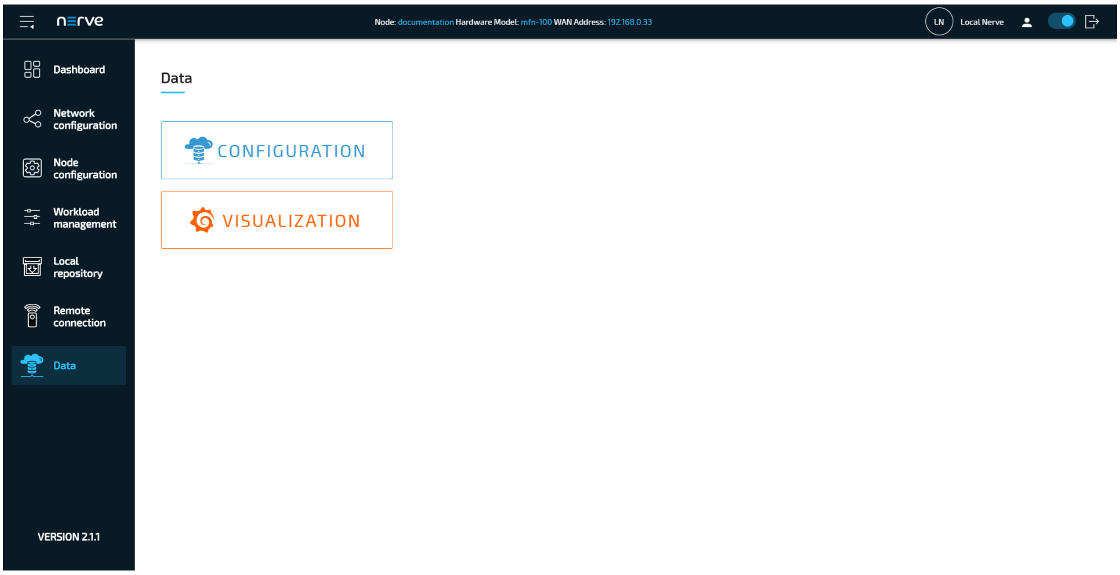
Configuring the Data Services Gateway on the node

With the S7 demo server deployed, the Gateway instance on the node must be configured next.

1. Access the Local UI on the node. This is Nerve Device specific. Refer to the table below for device specific links to the Local UI. The login credentials to the Local UI can be found in the customer profile.

Nerve Device	Physical port	Local UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

2. Select **Data** in the navigation on the left.

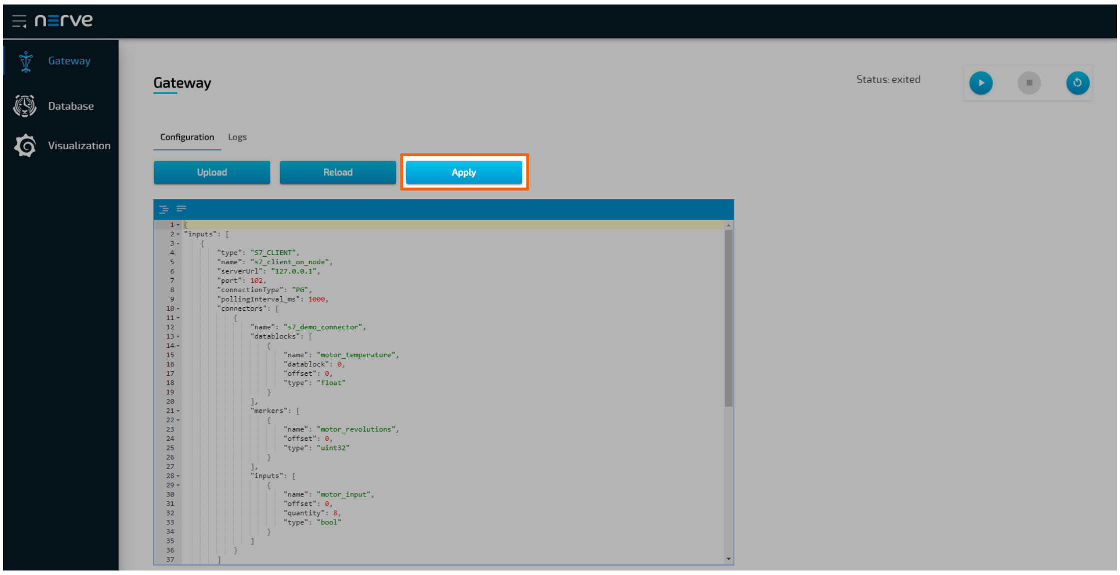


3. Select **CONFIGURATION**.

4. Enter the following configuration into the configuration editor of the Gateway:

```
{
"inputs": [
    {
        "type": "S7_CLIENT",
        "name": "s7_client_on_node",
        "serverUrl": "127.0.0.1",
        "port": 102,
        "connectionType": "PG",
        "pollingInterval_ms": 1000,
        "connectors": [
            {
                "name": "s7_demo_connector",
                "datablocks": [
                    {
                        "name": "motor_temperature",
                        "datablock": 0,
                        "offset": 0,
                        "type": "float"
                    }
                ],
                "merkers": [
                    {
                        "name": "motor_revolutions",
                        "offset": 0,
                        "type": "uint32"
                    }
                ],
                "inputs": [
                    {
                        "name": "motor_input",
                        "offset": 0,
                        "quantity": 8,
                        "type": "bool"
                    }
                ]
            }
        ]
    }
],
"outputs": [
    {
        "name": "ms_mqtt_broker",
        "type": "MQTT_PUBLISHER",
        "serverUrl": "wss://<MGMT-SYS>:443/dp/mqtt/wss",
        "username": "<SN>",
        "password": "<SID>",
        "keepAliveInterval_s": 20,
        "qos": 1,
        "cleanSession": true,
        "sslOptions": {
            "serverAuthRequired": false
        },
        "connectors": [
            {
                "name": "mqtt_demo_connector",
                "topic": "<SN>/demo_s7_motor_topic_0"
            }
        ]
    }
],
"connections" : [
    {
        "name" : "s7_client_to_mqtt_publisher",
        "input": { "index" : 0, "connector" : 0 },
        "output": { "index" : 0, "connector" : 0 }
    }
]
}
```

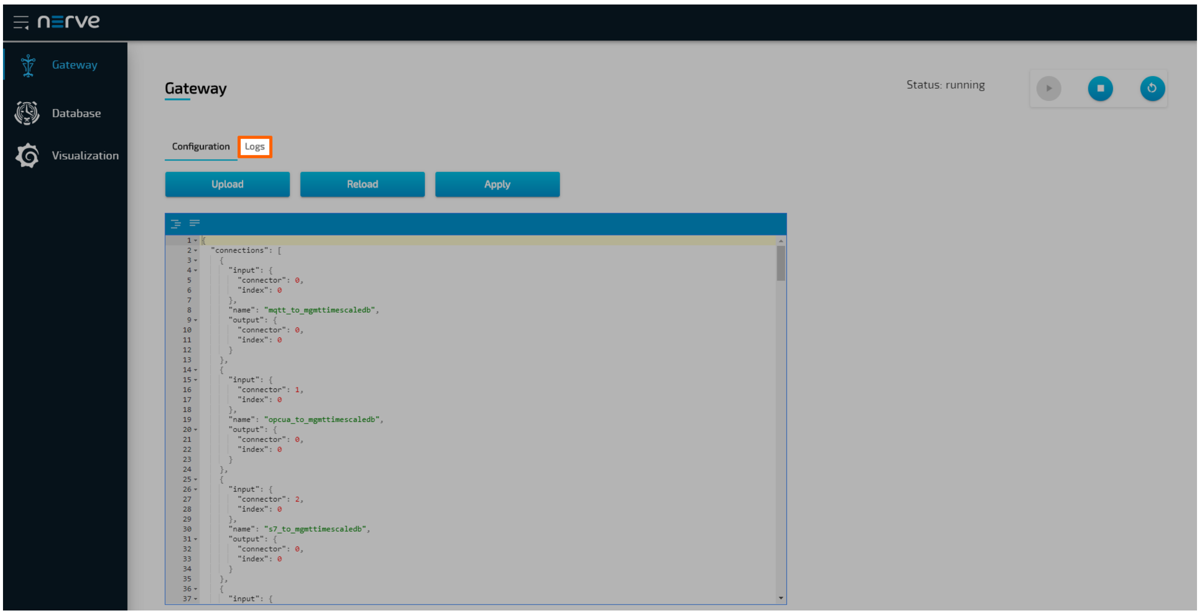
5. Select **Apply** to save the Gateway configuration. The Gateway will restart automatically.



```
{
"inputs": [
    {
        "type": "S7_CLIENT",
        "name": "s7_client_on_node",
        "serverUrl": "127.0.0.1",
        "port": 102,
        "connectionType": "PG",
        "pollingInterval_ms": 1000,
        "connectors": [
            {
                "name": "s7_demo_connector",
                "datablocks": [
                    {
                        "name": "motor_temperature",
                        "datablock": 0,
                        "offset": 0,
                        "type": "float"
                    }
                ],
                "merkers": [
                    {
                        "name": "motor_revolutions",
                        "offset": 0,
                        "type": "uint32"
                    }
                ],
                "inputs": [
                    {
                        "name": "motor_input",
                        "offset": 0,
                        "quantity": 8,
                        "type": "bool"
                    }
                ]
            }
        ]
```

The configuration defines the S7 demo server that was deployed earlier as a data input and the MQTT broker in the Management System as a data output. The S7 client will periodically read data from the S7 demo server and publish it to the MQTT broker of the Management System. The temperature of the motor is stored in the datablock `0` with offset `0` and data type `float`. The number of revolutions of the motor is stored in the merker with offset `0` and data type `uint32`. Eight digital inputs from the demo server are also read as eight different boolean values.

To verify that the Gateway accepted the configuration and works as expected, select the **Logs** tab in the Gateway UI and check for errors.

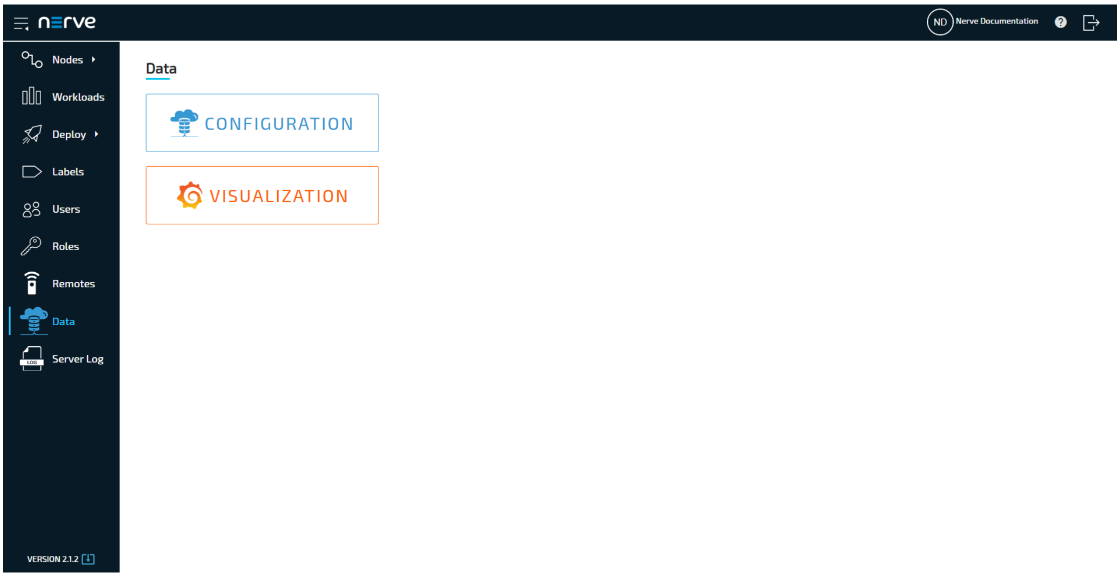


```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

Configuring the Data Services Gateway in the Management System

In order to visualize the data collected by the S7 client, the data published to the MQTT broker in the Management System needs to be collected and stored in the database. The example configuration below connects the input of the Central Data Gateway to the MQTT broker of the Management System, and transfers data through its output into Management System TimescaleDB storage.

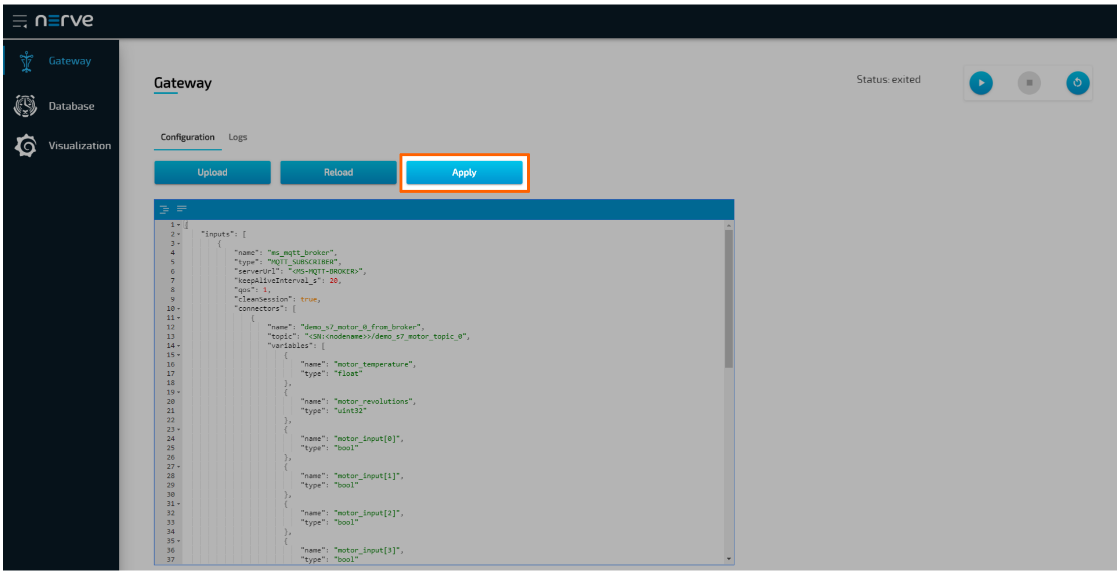
1. Log in to the Management System.
2. Select **Data** in the navigation on the left. Make sure that the user has the necessary permissions to access the Data Services.
3. Select **Configuration** to access the Data Services UI.



4. Enter the following configuration into the configuration editor of the Gateway. Replace `<nodename>` in the configuration below with the name of the node that the Docker workload has been deployed to earlier.

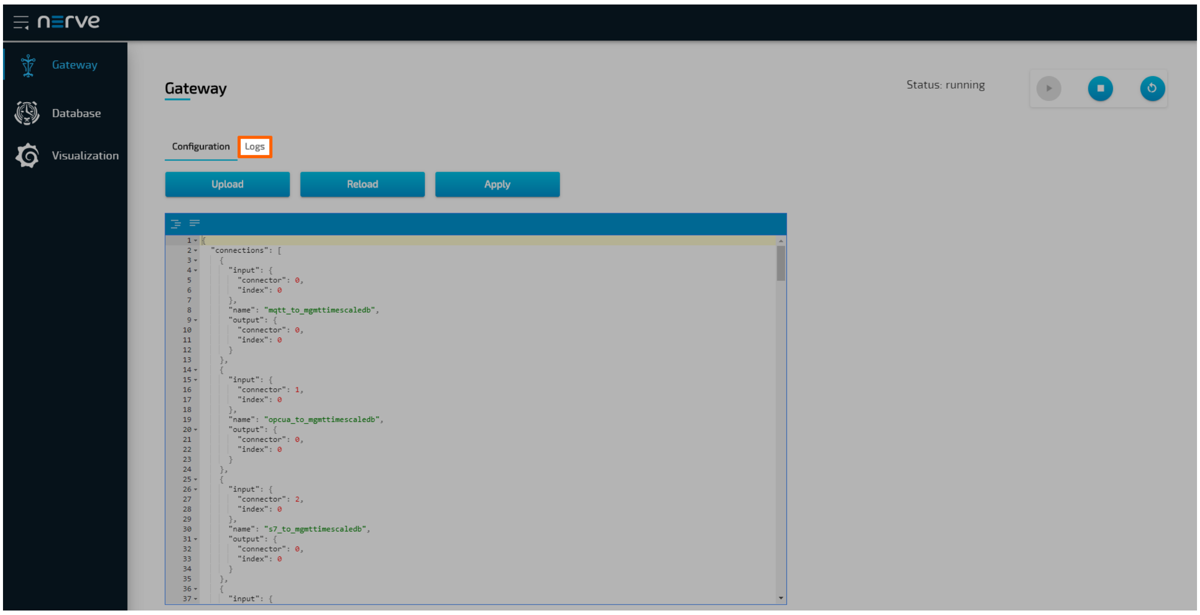
```
{
    "inputs": [
        {
            "name": "ms_mqtt_broker",
            "type": "MQTT_SUBSCRIBER",
            "serverUrl": "<MS-MQTT-BROKER>",
            "keepAliveInterval_s": 20,
            "qos": 1,
            "cleanSession": true,
            "connectors": [
                {
                    "name": "demo_s7_motor_0_from_broker",
                    "topic": "<SN:<nodename>>/demo_s7_motor_topic_0",
                    "variables": [
                        {
                            "name": "motor_temperature",
                            "type": "float"
                        },
                        {
                            "name": "motor_revolutions",
                            "type": "uint32"
                        },
                        {
                            "name": "motor_input[0]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[1]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[2]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[3]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[4]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[5]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[6]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[7]",
                            "type": "bool"
                        }
                    ]
                }
            ]
        }
    ],
    "outputs": [
        {
            "name": "cloud_timescale",
            "type": "DB_TIMESCALE",
            "url": "localhost",
            "port": 5432,
            "connectors": [
                {
                    "name": "<nodename>",
                    "dbName": "<SN:<nodename>>"
                }
            ]
        }
    ],
    "connections": [
        {
            "name": "s7_data",
            "input": {
                "index": 0,
                "connector": 0
            },
            "output": {
                "index": 0,
                "connector": 0
            }
        }
    ]
}
```

5. Select **Apply** to save the Gateway configuration. The Gateway will restart automatically.



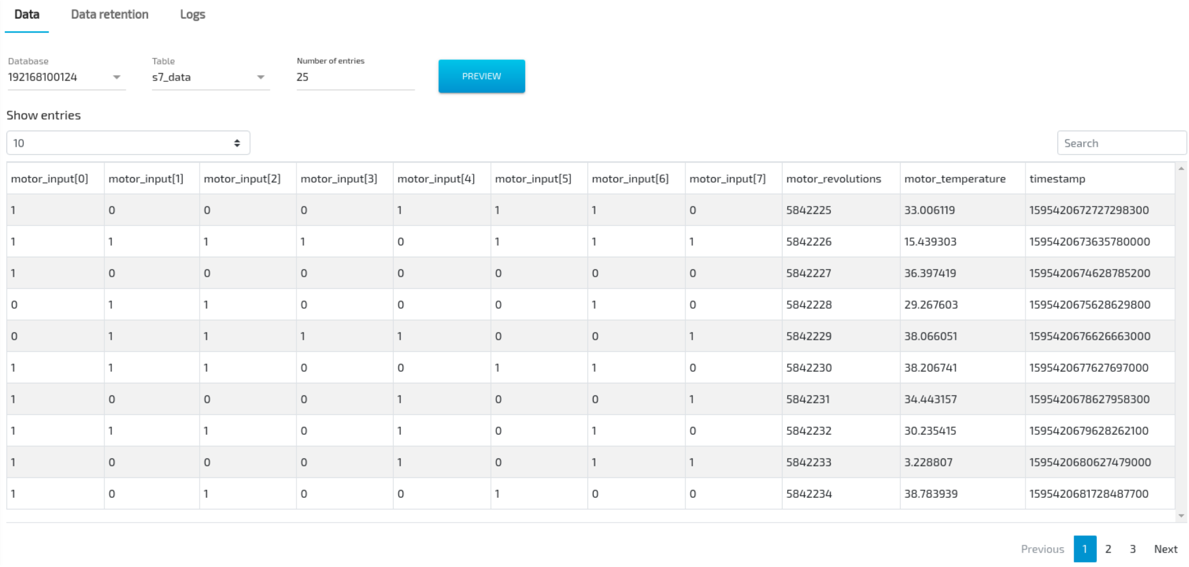
```
{
    "inputs": [
        {
            "name": "ms_mqtt_broker",
            "type": "MQTT_SUBSCRIBER",
            "serverUrl": "<MS-MQTT-BROKER>",
            "keepAliveInterval_s": 20,
            "qos": 1,
            "cleanSession": true,
            "connectors": [
                {
                    "name": "demo_s7_motor_0_from_broker",
                    "topic": "<SN:<nodename>>/demo_s7_motor_topic_0",
                    "variables": [
                        {
                            "name": "motor_temperature",
                            "type": "float"
                        },
                        {
                            "name": "motor_revolutions",
                            "type": "uint32"
                        },
                        {
                            "name": "motor_input[0]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[1]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[2]",
                            "type": "bool"
                        },
                        {
                            "name": "motor_input[3]",
                            "type": "bool"
```

To verify that the Gateway accepted the configuration and works as expected, select the **Logs** tab and check for errors.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

To verify if data is stored in the database, select **Database** in the navigation on the left. Select the serial number of the node under **Database** and **s7_data** under **Table**. Select **Preview** to display the data stored in the database as a table.



Database: 192168100124

Table: s7_data

Number of entries: 25

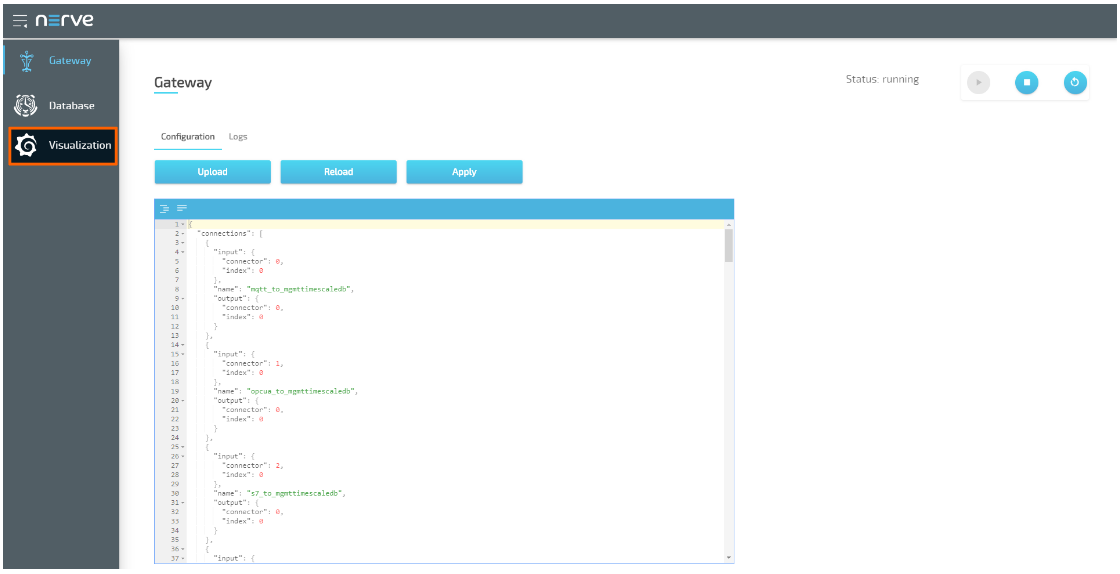
Show entries: 10

motor_input[0]	motor_input[1]	motor_input[2]	motor_input[3]	motor_input[4]	motor_input[5]	motor_input[6]	motor_input[7]	motor_revolutions	motor_temperature	timestamp
1	0	0	0	1	1	1	0	5842225	33.006119	1595420672727298300
1	1	1	1	0	1	1	1	5842226	15.439303	1595420673635780000
1	0	0	0	0	0	0	0	5842227	36.397419	1595420674628785200
0	1	1	0	0	0	1	0	5842228	29.267603	1595420675628629800
0	1	1	1	1	0	0	1	5842229	38.066051	1595420676626663000
1	1	1	0	0	1	1	0	5842230	38.206741	1595420677627697000
1	0	0	0	1	0	0	1	5842231	34.443157	1595420678627958300
1	1	1	0	1	0	1	0	5842232	30.235415	1595420679628262100
1	0	0	0	1	0	1	1	5842233	3.228807	1595420680627479000
1	0	1	0	0	1	0	0	5842234	38.783939	1595420681728487700

Central data visualization in the Management System

To visualize the motor temperature data in the Management System, open the central visualization element through the Data Services UI in the Management System.

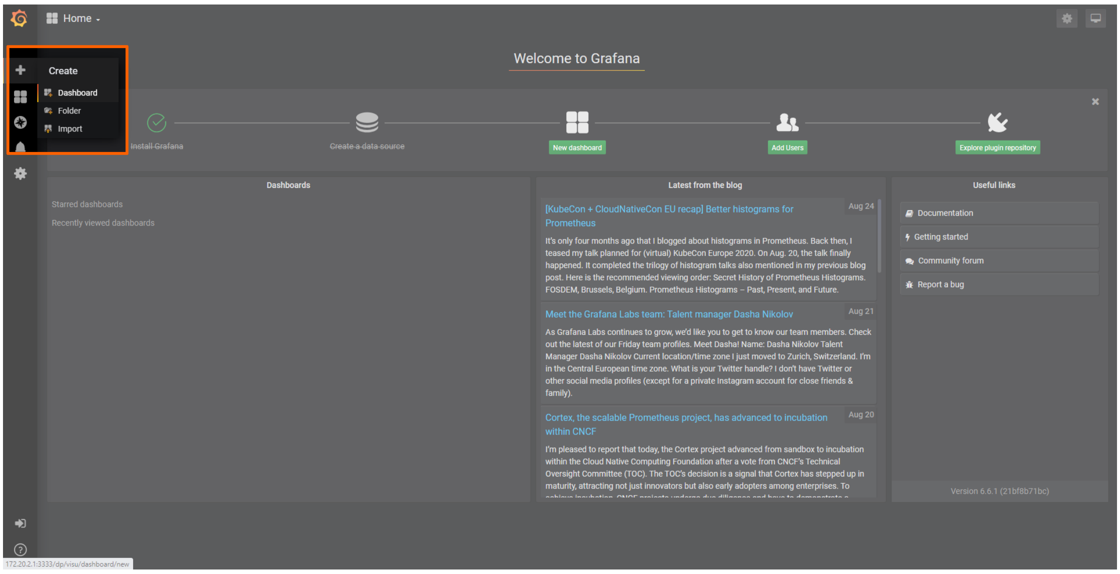
1. Select **Visualization** in the navigation on the left. The Grafana UI will open.



Status: running

```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

2. Select **+ > Dashboard** in the navigation on the left. A box will appear.



[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

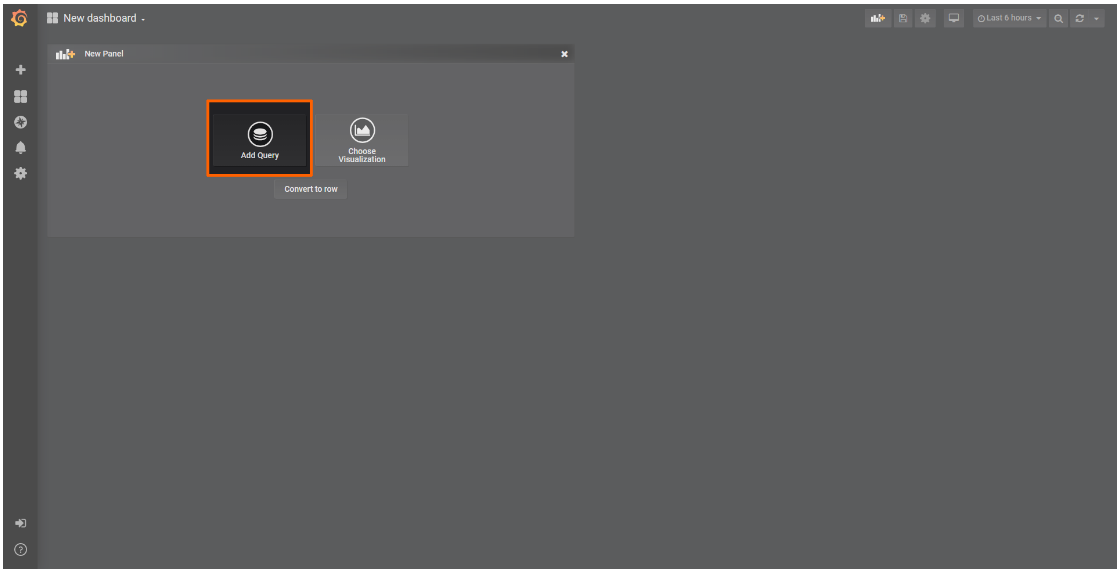
Meet the Grafana Labs team: Talent manager Dasha Nikolov Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends & family).

Cortex, the scalable Prometheus project, has advanced to incubation within CNCF Aug 20

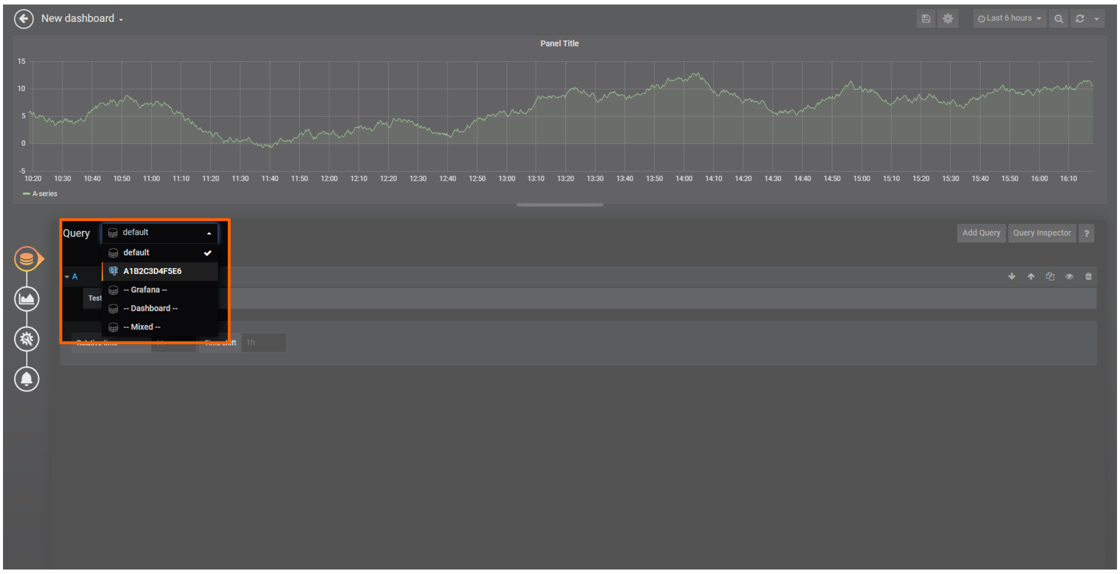
I'm pleased to report that today, the Cortex project advanced from sandbox to incubation within the Cloud Native Computing Foundation after a vote from CNCF's Technical Oversight Committee (TOC). The TOC's decision is a signal that Cortex has stepped up in maturity, attracting not just innovators but also early adopters among enterprises. To

3. Select **Add Query** in the **New Panel** box.



4. Select the data source from the drop-down menu. The name of the data source is the name and serial number of the node, formatted as `<nodename> (<serialnumber>)`.

> **NOTE**
>
> Note that multiple data sources can be selected in the Grafana instance in the Management System, depending on the number of nodes that are registered. Make sure to remember the serial number of the node that was used for workload deployment before.

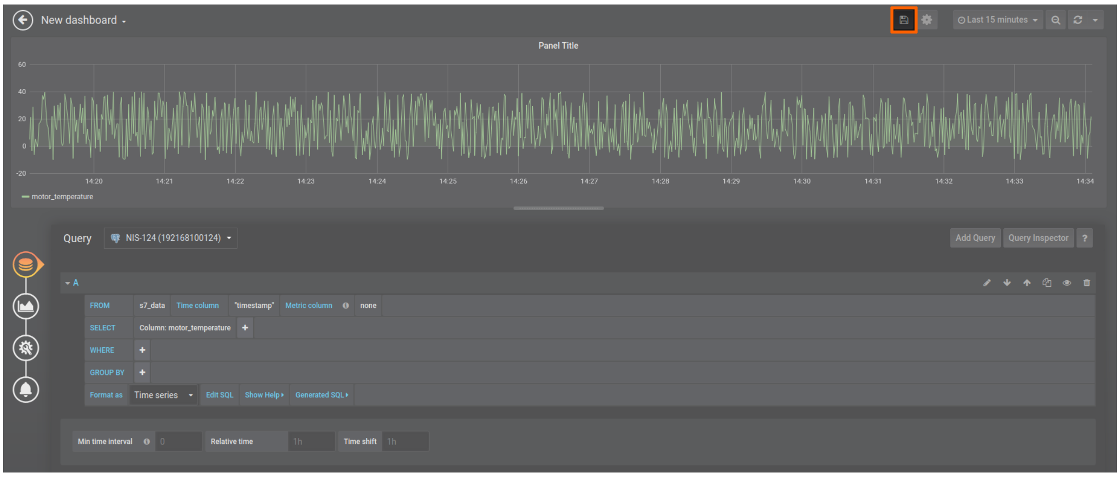


5. Fill in the following query information:

Setting	Value
FROM	`s7_data` **Time column**: "timestamp"
SELECT	Column: motor_temperature

Setting	Value
Format as	Time series

6. Select the save icon in the upper-right corner to save the dashboard.



Query NIS-124 (192168100124)

FROM s7_data Time column "timestamp" Metric column none

SELECT Column: motor_temperature

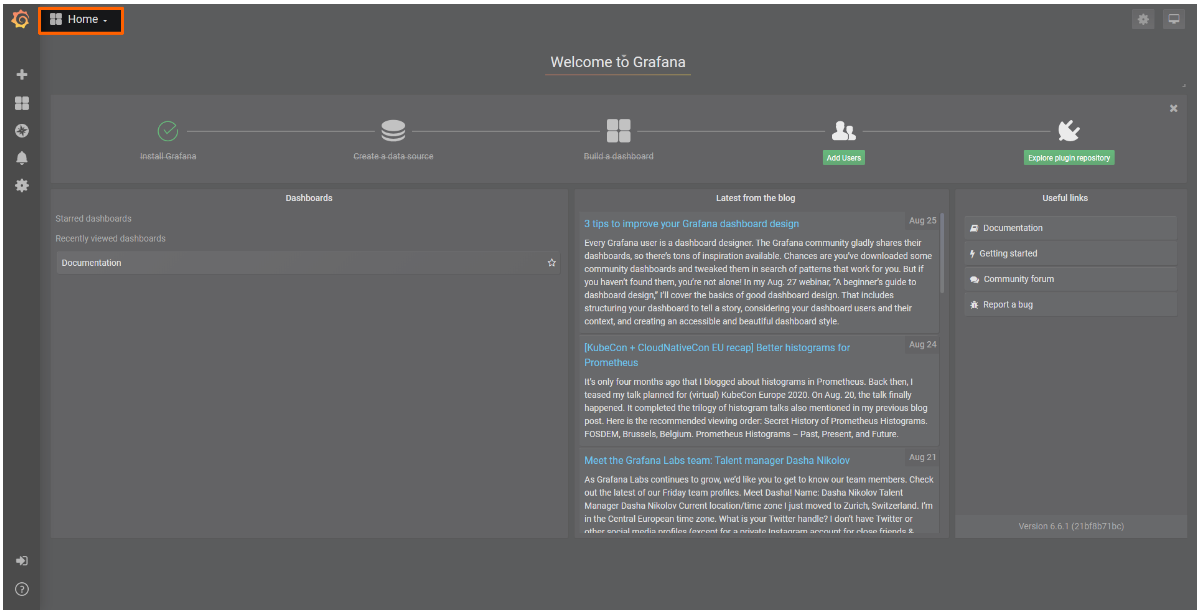
WHERE

GROUP BY

Format as Time series

Min time interval 0 Relative time 1h Time shift 1h

The dashboard can be accessed from the Grafana home menu.



3 tips to improve your Grafana dashboard design Aug 25

Every Grafana user is a dashboard designer. The Grafana community gladly shares their dashboards, so there's tons of inspiration available. Chances are you've downloaded some community dashboards and tweaked them in search of patterns that work for you. But if you haven't found them, you're not alone! In my Aug. 27 webinar, "A beginner's guide to dashboard design," I'll cover the basics of good dashboard design. That includes structuring your dashboard to tell a story, considering your dashboard users and their context, and creating an accessible and beautiful dashboard style.

[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

Meet the Grafana Labs team: Talent manager Dasha Nikolov Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends &

MQTT publisher to OPC UA Server at the node

In this example, temperature and humidity data is provided from a sensor simulation as an MQTT Publisher. The sensor simulation is running on the node. With the Data Services, the data is then provided as OPC UA Server on the node. A development PC with the UaExpert OPC UA Client is used to connect to the OPC UA Server on the node to read the data.

The instructions below cover the following steps:

- Provisioning an MQTT broker as a Docker workload
- Provisioning an MQTT Publisher simulation as a Docker workload
- Deploying the provisioned Docker workloads to the target node
- Configuring the Data Services Gateway on the node

- Reading data with the UaExpert OPC UA Client on a development PC

Provisioning and deploying the sensor simulation and the MQTT broker

In the instructions below two Docker workloads will be provisioned and deployed:

An MQTT broker must to be deployed to the node first in order for the sensor simulation to function. The EMQX MQTT broker is used in this example that can be downloaded from the Docker Hub registry.

Afterwards the temperature and humidity sensors simulation MQTT publisher is deployed. Download the **Data Services MQTT demo sensor** found under **Example Applications** from the Nerve Software Center. This is the Docker image that is required for provisioning the demo sensor as a Docker workload.

1. Log in to the Management System. Make sure that the user has the permissions to access the Data Services.

2. Provision a Docker workload for the EMQX MQTT broker by following Provisioning a Docker workload. This example uses **emqx-4.1.0** as the workload name. Use the following workload version settings:

Setting	Value
Name	Enter any name for the workload version.
Release name	Enter any release name.
DOCKER IMAGE	Select **From registry** and enter `emqx/emqx:v4.1.0`.
Container name	`emqx`
Network name	`host`

3. Provision a Docker workload for the sensor simulation by following Provisioning a Docker workload. Use the following workload version settings:

Setting	Value
Name	Enter any name for the workload version.
Release name	Enter any release name.
DOCKER IMAGE	Select **Upload** to add the Docker image of the temperature sensor simulation that has been downloaded before.
Container name	`ttt-mqtt-demo-sensor-nerve-dp-2.1.1`
Network name	`host`

4. Deploy both provisioned Docker workloads above by following Deploying a workload.

> **NOTE**
>
> Remember the node name and serial number of the target node. They are needed for the JSON configuration.

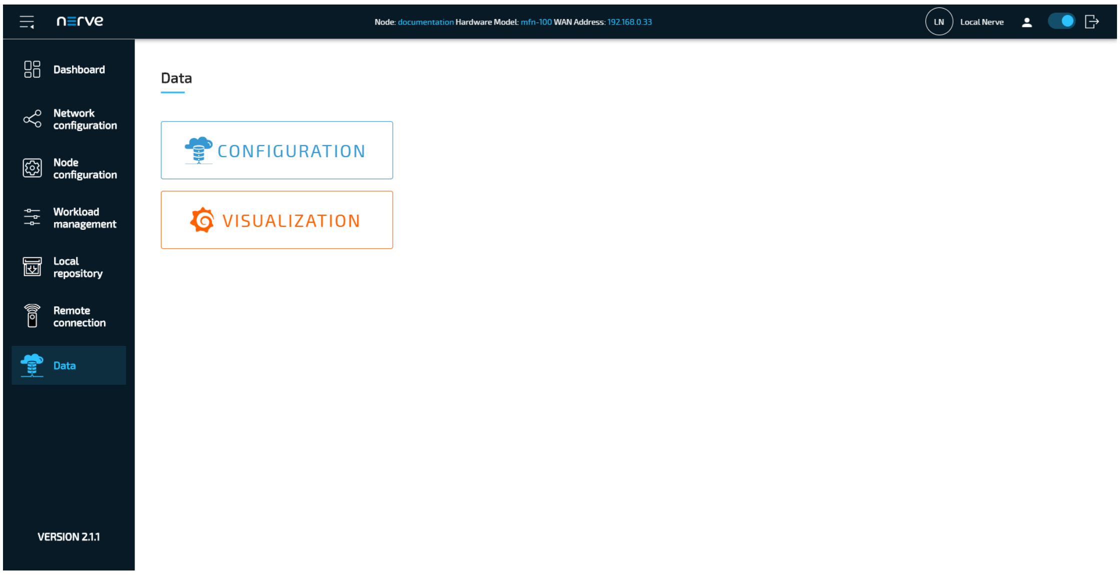
Configuring the Data Services Gateway on the node

The input data is a subscription to temperature and humidity variables on the MQTT broker. An OPC UA Server is set up as data output. The input and output are linked in connections.

1. Access the Local UI on the node. This is Nerve Device specific. Refer to the table below for device specific links to the Local UI. The login credentials to the Local UI can be found in the customer profile.

Nerve Device	Physical port	Local UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

2. Select **Data** in the navigation on the left.

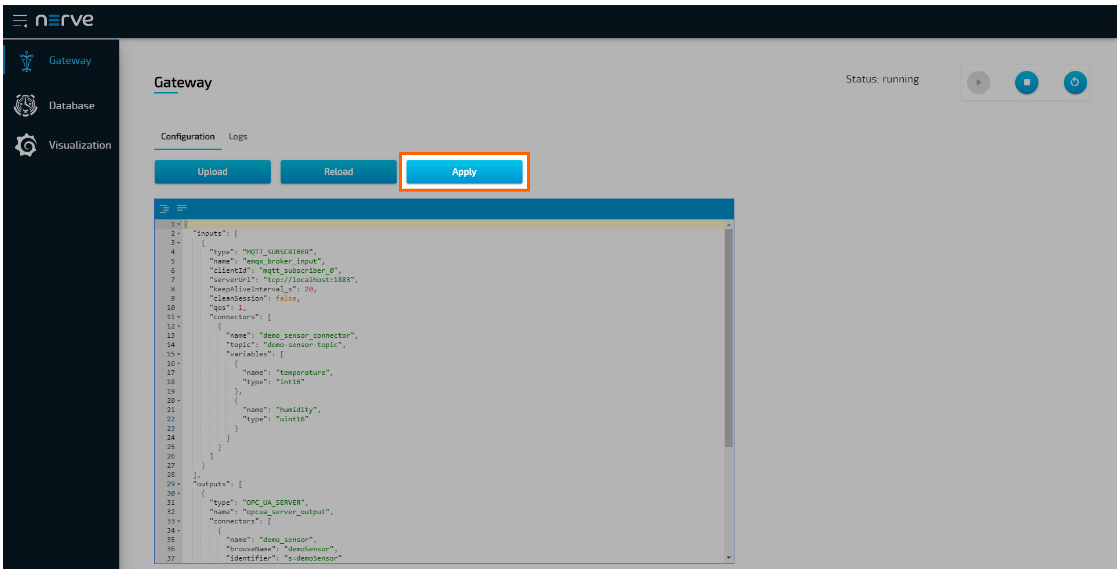


3. Select **CONFIGURATION**.

4. Enter the following configuration into the configuration editor of the Gateway:

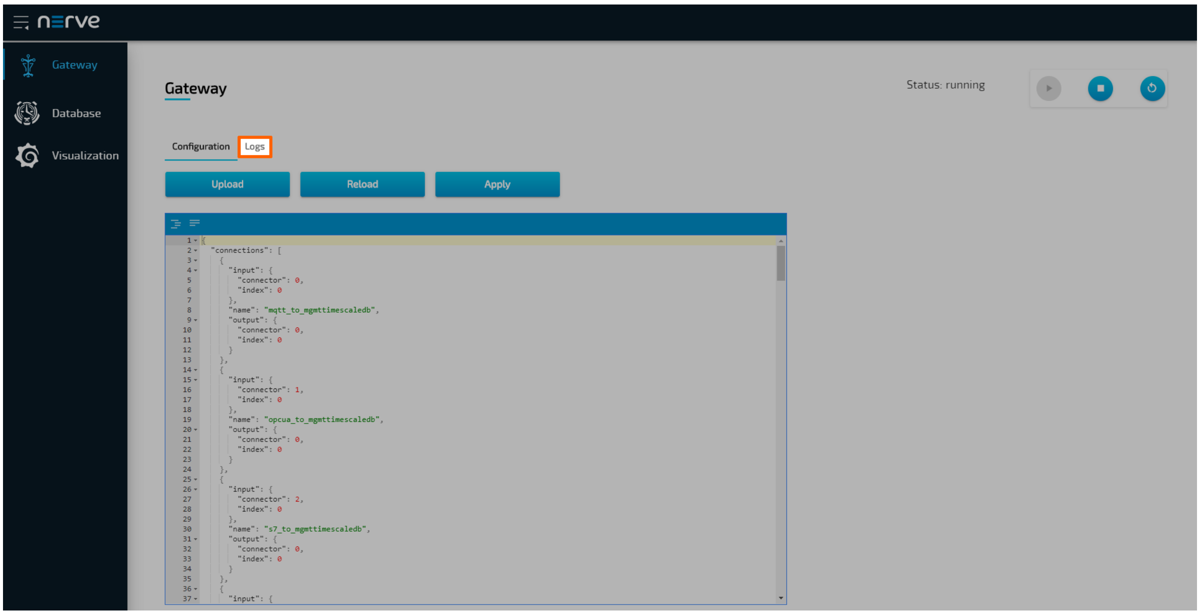
```
{
  "inputs": [
    {
      "type": "MQTT_SUBSCRIBER",
      "name": "emqx_broker_input",
      "clientId": "mqtt_subscriber_0",
      "serverUrl": "tcp://localhost:1883",
      "keepAliveInterval_s": 20,
      "cleanSession": false,
      "qos": 1,
      "connectors": [
        {
          "name": "demo_sensor_connector",
          "topic": "demo-sensor-topic",
          "variables": [
            {
              "name": "temperature",
              "type": "int16"
            },
            {
              "name": "humidity",
              "type": "uint16"
            }
          ]
        }
      ]
    }
  ],
  "outputs": [
    {
      "type": "OPC_UA_SERVER",
      "name": "opcua_server_output",
      "connectors": [
        {
          "name": "demo_sensor",
          "browseName": "demoSensor",
          "identifier": "s=demoSensor"
        }
      ]
    }
  ],
  "connections": [
    {
      "name": "mqtt_publisher_TO_opcua_server",
      "input": {
        "index": 0,
        "connector": 0
      },
      "output": {
        "index": 0,
        "connector": 0
      }
    }
  ]
}
```

5. Select **Apply** to save the Gateway configuration. The Gateway will restart automatically.



```
{
  "inputs": [
    {
      "type": "MQTT_SUBSCRIBER",
      "name": "emqx_broker_input",
      "clientId": "mqtt_subscriber_0",
      "serverUrl": "tcp://localhost:1883",
      "keepAliveInterval_s": 20,
      "cleanSession": false,
      "qos": 1,
      "connectors": [
        {
          "name": "demo_sensor_connector",
          "topic": "demo-sensor-topic",
          "variables": [
            {
              "name": "temperature",
              "type": "int16"
            },
            {
              "name": "humidity",
              "type": "uint16"
            }
          ]
        }
      ]
    }
  ],
  "outputs": [
    {
      "type": "OPC_UA_SERVER",
      "name": "opcua_server_output",
      "connectors": [
        {
          "name": "demo_sensor",
          "browseName": "demoSensor",
          "identifier": "s=demoSensor"
```

To verify that the Gateway accepted the configuration and works as expected, select the **Logs** tab in the Gateway UI and check for errors.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

Reading data with the UaExpert OPC UA Client

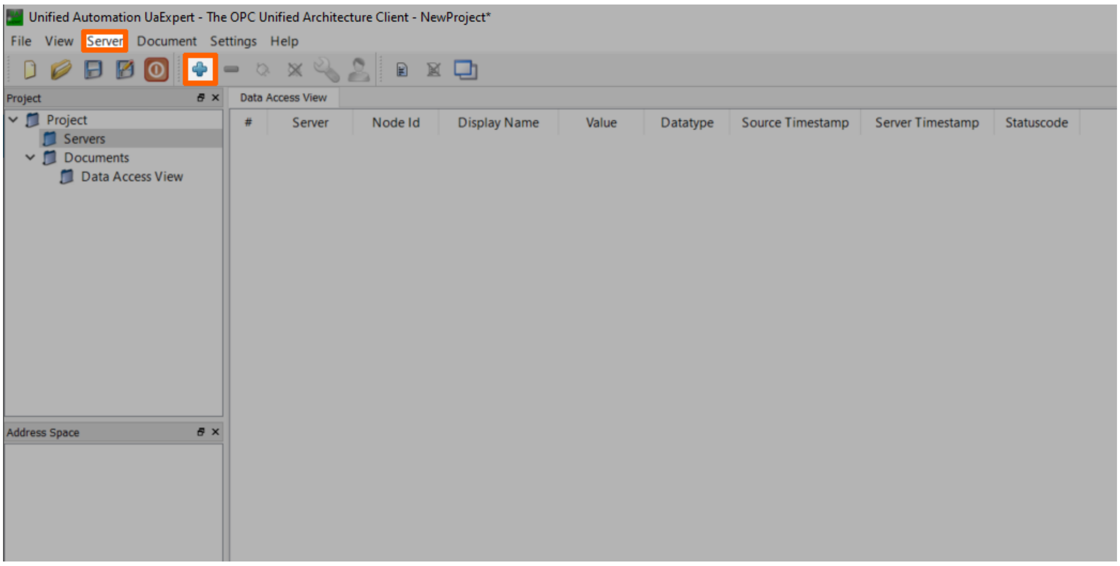
The temperature and humidity simulation data can be checked visually using a third party application. This example uses the UaExpert OPC UA Client. Add a new server with the address `opc.tcp://172.20.2.1:4840` (here port 4840 is specified as the default of the Gateway) and connect to it. The desired variables are located in the address space, and by adding them to the data access view, their values can be observed.

NOTE

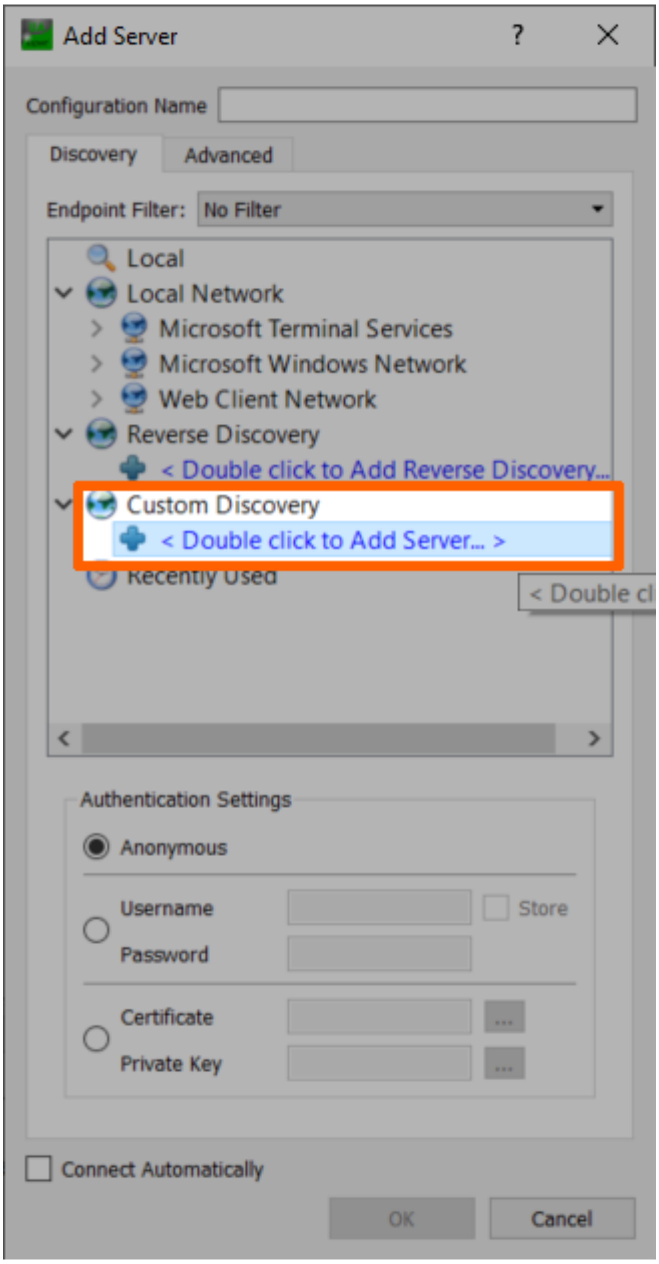
In order to download the UaExpert OPC UA Client, it is required to create a free account at unified-automation.com.

Before continuing, make sure to follow the UaExpert documentation through the first steps with the UaExpert client. Afterwards follow the instructions below:

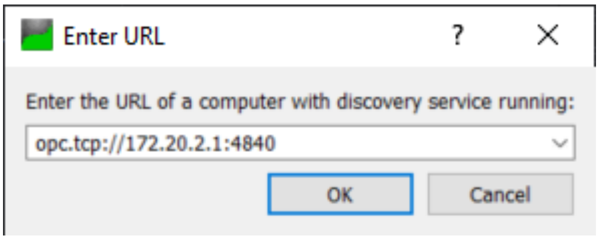
1. Select **Server > Add...** in the main menu or select the plus icon in the toolbar.



2. Select **Double click to Add Server...** under **Custom Discovery**.



3. Enter `opc.tcp://172.20.2.1:4840`. The new server now appears under **Custom Discovery**.

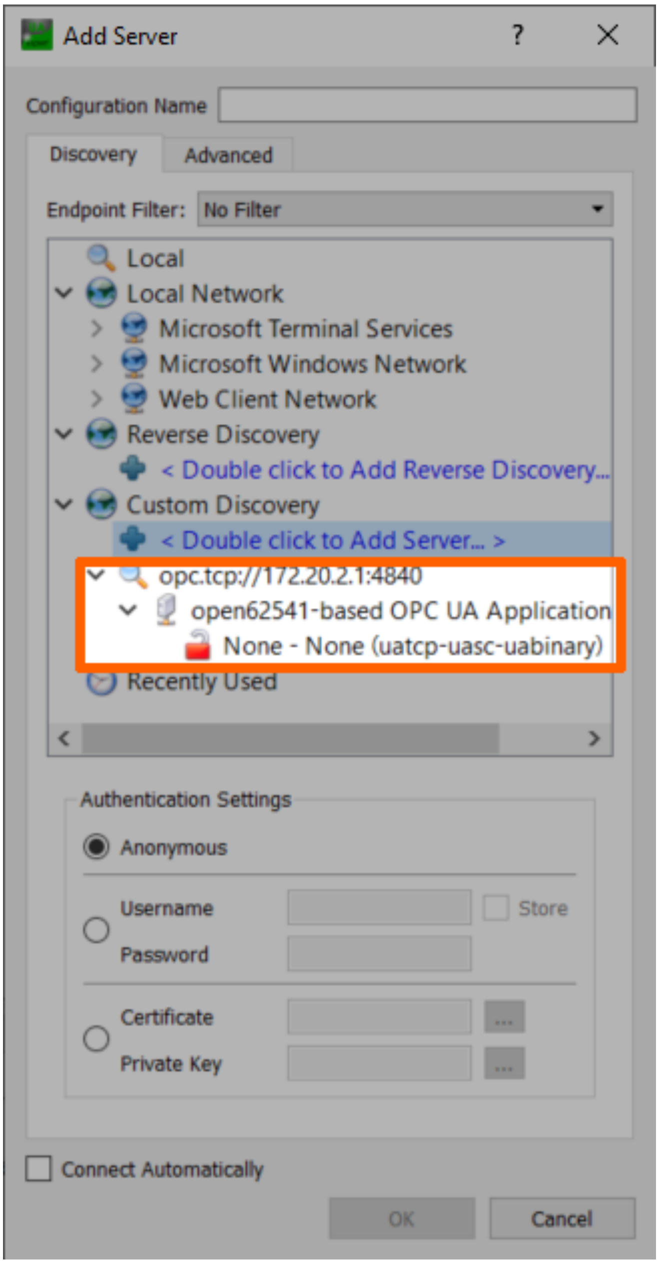


Enter URL

Enter the URL of a computer with discovery service running:

opc.tcp://172.20.2.1:4840

4. Expand the arrow next to the left of the new server.



Endpoint Filter: No Filter

opc.tcp://172.20.2.1:4840

open62541-based OPC UA Application

None - None (uatcp-uasc-uabinary)

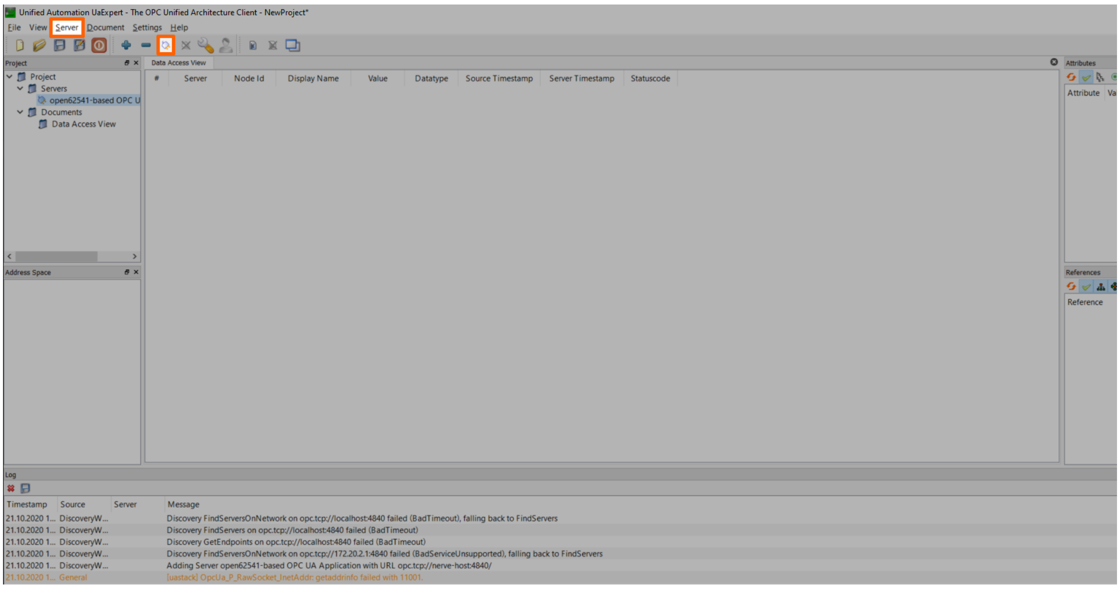
● Anonymous

□ Connect Automatically

5. Double-click the **None** security policy. The server configuration now appears in the **Project** panel under **Servers** on the left side.

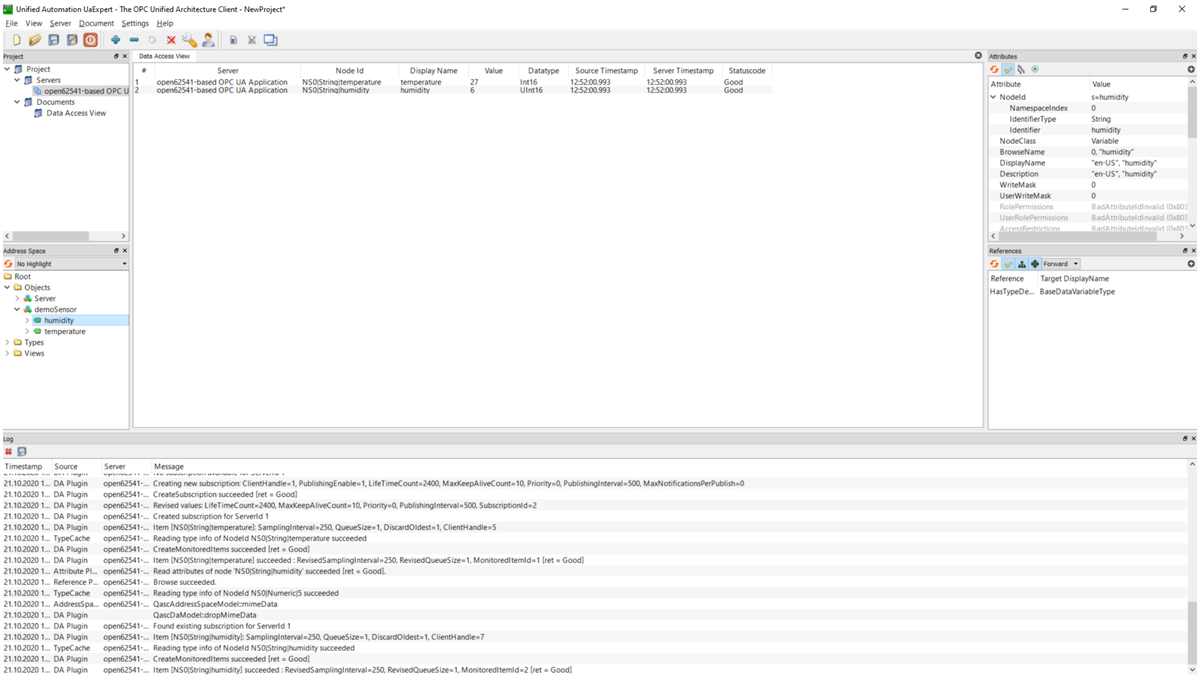
6. Select the server configuration in the **Project** panel.

7. Select **Server > Connect** in the main menu or select the **Connect Server** icon in the toolbar to establish a connection to the server.



```
Timestamp      Source        Server  Message
21.10.2020 1... DiscoveryW...        Discovery FindServersOnNetwork on opc.tcp://localhost:4840 failed (BadTimeout), falling back to FindServers
21.10.2020 1... DiscoveryW...        Discovery FindServers on opc.tcp://localhost:4840 failed (BadTimeout)
21.10.2020 1... DiscoveryW...        Discovery GetEndpoints on opc.tcp://localhost:4840 failed (BadTimeout)
21.10.2020 1... DiscoveryW...        Discovery FindServersOnNetwork on opc.tcp://172.20.2.1:4840 failed (BadServiceUnsupported), falling back to FindServers
21.10.2020 1... DiscoveryW...        Adding Server open62541-based OPC UA Application with URL opc.tcp://nerve-host:4840/
21.10.2020 1... General              [uastack] OpcUa_P_RawSocket_InetAddr: getaddrinfo failed with 11001.
```

All connection steps are logged in the **Log** panel at the bottom. Once connected, the OPC UA Server's address space appears in the **Address Space** panel on the left. Expand **Objects > demoSensor** in the **Address Space** panel on the left and drag-and-drop the temperature and humidity variables to the **Data Access View** in the middle to monitor their values.



#	Server	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	open62541-based OPC UA Application	NS0|String|temperature	temperature	27	Int16	12:52:00.993	12:52:00.993	Good
2	open62541-based OPC UA Application	NS0|String|humidity	humidity	6	UInt16	12:52:00.993	12:52:00.993	Good

Attribute	Value
NodeId	s=humidity
NamespaceIndex	0
IdentifierType	String
Identifier	humidity
NodeClass	Variable
BrowseName	0, "humidity"
DisplayName	"en-US", "humidity"
Description	"en-US", "humidity"
WriteMask	0
UserWriteMask	0
RolePermissions	BadAttributeIdInvalid (0x803
UserRolePermissions	BadAttributeIdInvalid (0x803
AccessRestrictions	BadAttributeIdInvalid (0x803

Reference	Target DisplayName
HasTypeDe...	BaseDataVariableType

```
Timestamp       Source        Server        Message
21.10.2020 1... DA Plugin     open62541-... Creating new subscription: ClientHandle=1, PublishingEnable=1, LifeTimeCount=2400, MaxKeepAliveCount=10, Priority=0, PublishingInterval=500, MaxNotificationsPerPublish=0
21.10.2020 1... DA Plugin     open62541-... CreateSubscription succeeded [ret = Good]
21.10.2020 1... DA Plugin     open62541-... Revised values: LifeTimeCount=2400, MaxKeepAliveCount=10, Priority=0, PublishingInterval=500, SubscriptionId=2
21.10.2020 1... DA Plugin     open62541-... Created subscription for ServerId 1
21.10.2020 1... DA Plugin     open62541-... Item [NS0|String|temperature]: SamplingInterval=250, QueueSize=1, DiscardOldest=1, ClientHandle=5
21.10.2020 1... TypeCache     open62541-... Reading type info of NodeId NS0|String|temperature succeeded
21.10.2020 1... DA Plugin     open62541-... CreateMonitoredItems succeeded [ret = Good]
21.10.2020 1... DA Plugin     open62541-... Item [NS0|String|temperature] succeeded : RevisedSamplingInterval=250, RevisedQueueSize=1, MonitoredItemId=1 [ret = Good]
21.10.2020 1... Attribute Pl... open62541-... Read attributes of node 'NS0|String|humidity' succeeded [ret = Good].
21.10.2020 1... Reference P... open62541-... Browse succeeded.
21.10.2020 1... TypeCache     open62541-... Reading type info of NodeId NS0|Numeric|5 succeeded
21.10.2020 1... AddressSpa... open62541-... QascAddressSpaceModel::mimeData
21.10.2020 1... DA Plugin                   QascDaModel::dropMimeData
21.10.2020 1... DA Plugin     open62541-... Found existing subscription for ServerId 1
21.10.2020 1... DA Plugin     open62541-... Item [NS0|String|humidity]: SamplingInterval=250, QueueSize=1, DiscardOldest=1, ClientHandle=7
21.10.2020 1... TypeCache     open62541-... Reading type info of NodeId NS0|String|humidity succeeded
21.10.2020 1... DA Plugin     open62541-... CreateMonitoredItems succeeded [ret = Good]
21.10.2020 1... DA Plugin     open62541-... Item [NS0|String|humidity] succeeded : RevisedSamplingInterval=250, RevisedQueueSize=1, MonitoredItemId=2 [ret = Good]
```

Receiving data via MQTT for Analytics and Visualization

In this example, data from a sensor providing temperature and humidity as an MQTT Publisher is visualized and processed via the Data Services and the analytics element. Data will be displayed before and after processing.

The instructions below cover the following steps:

- Provisioning an MQTT broker as a Docker workload
- Provisioning an MQTT Publisher simulation as a Docker workload
- Deploying the provisioned Docker workloads to the target node
- Creating and provisioning an analytics app with the Nerve Data SDK
- Deploying the analytics app as a Docker workload
- Local Data Visualization of temperature data before and after processing

Provisioning and deploying the sensor simulation and the MQTT broker

In the instructions below two Docker workloads will be provisioned and deployed:

An MQTT broker must to be deployed to the node first in order for the sensor simulation to function. The EMQX MQTT broker is used in this example that can be downloaded from the Docker Hub registry.

After that the temperature and humidity sensors simulation MQTT publisher is deployed. Download the **Data Services MQTT demo sensor** found under **Example Applications** from the Nerve Software Center. This is the Docker image that is required for provisioning the demo sensor as a Docker workload.

1. Log in to the Management System. Make sure that the user has the permissions to access the Data Services.

2. Provision a Docker workload for the EMQX MQTT broker by following Provisioning a Docker workload. This example uses **emqx-4.1.0** as the workload name. Use the following workload version settings:

Setting	Value
Name	Enter any name for the workload version.
Release name	Enter any release name.
DOCKER IMAGE	Select **From registry** and enter `emqx/emqx:v4.1.0`.
Container name	`emqx`
Network name	`host`

3. Provision a Docker workload for the sensor simulation by following Provisioning a Docker workload. Use the following workload version settings:

Setting	Value
Name	Enter any name for the workload version.
Release name	Enter any release name.
DOCKER IMAGE	Select **Upload** to add the Docker image of the temperature sensor simulation that has been downloaded before.
Container name	`ttt-mqtt-demo-sensor-nerve-dp-2.1.1`
Network name	`host`

4. Deploy both provisioned Docker workloads above by following Deploying a workload.

Preparing the Nerve Data SDK

The Nerve Data SDK is required for working with analytics apps. They are created, built and provisioned with it. Download the **Nerve Data SDK** found under **Nerve Tools** from the Nerve Software Center. Refer to Data analytics for more information.

Creating and provisioning an analytics app

Before working with the SDK, make sure that the Conda environment is active. If the Conda environment is active it will be displayed in parentheses in front. The default name of the Conda environment is `nerve-dp-analytics`. Activate the Conda environment by entering the following command:

```
source miniconda/bin/activate <environmentname>
```

The Conda environment automatically deactivates after a restart so it needs to be activated whenever it is used.

1. Enter the following command to create an analytics app. `demo_sensor_analytics_app` is the name used for this example:

   ```
   nerve-analytics create demo_sensor_analytics_app .
   ```

2. Enter `cd demo_sensor_analytics_app` to navigate to the newly created folder.

3. Edit the `demo_sensor_analytics_app.py` file and insert the following code:

   ```
   import signal
   import sys

   from nerve_dp_analytics.stream.inputs.input_zeromq import Stream_Input_Zeromq
   from nerve_dp_analytics.batch.outputs.output_timescaledb import Batch_Output_Times

   running = True

   def sig_hdlr(signal, frame):
       global running
       running = False

       if siz:
           try:
               siz.clear()
           except Exception as e:
               print(e)

       if bot:
           try:
               bot.clear()
           except Exception as e:
               print(e)

       print('Exiting...')
       sys.exit(0)

   # catch CTRL+C
   signal.signal(signal.SIGINT, sig_hdlr)

   def celsius_to_kelvin(value):
       return value + 273.15
   ```

```
def normalize_humidity(value):
    return value / 100

try:
    siz = Stream_Input_Zeromq('demo-sensor-analytics-app-siz',
                              host='172.20.10.1',
                              port=5555,
                              topic='demo-sensor-topic')

    bot = Batch_Output_Timescaledb('demo-sensor-analytics-app-bot',
                                   table_name='demo_sensor_analyzed_data',
                                   vars={'temperature': 'real',
                                         'humidity': 'real'})

    while(running):
        try:
            data = siz.receive(Stream_Input_Zeromq.DTYPE_LIST)
            new_data = list()

            for d in data:
                nd = dict()

                nd['timestamp'] = d['timestamp']
                nd['temperature'] = celsius_to_kelvin(d['temperature'])
                nd['humidity'] = normalize_humidity(d['humidity'])

                new_data.append(nd)

            for nd in new_data:
                bot.send(nd)
        except Exception as e:
            print(e)
except Exception as e:
    print(e)
```

This analytics app receives data from the Gateway through the ZeroMQ Stream Input, which is a default way of transferring data between the Gateway and analytics. Processed data is stored in a TimescaleDB via the TimescaleDB Batch Output. When only `table_name` is provided for this output, the analytics write data into the default database of the node that has the node serial number as a name.

In this example, basic processing is done on the data provided by the demoSensor. Temperature data is converted from Celsius to Kelvin while humidity data is normalized to a range between 0 and 1.

> **NOTE**
>
> The ZeroMQ Publisher output of the Gateway must publish messages on `172.20.10.1` if analytics are running on the node in the `nerve-dp` Docker network. Consequently, the ZeroMQ Stream Input of the analytics must listen on the same IP address.

4. Edit the `Dockerfile` and insert the following:

```
FROM python:3.8.3-slim-buster

WORKDIR /nerve

COPY nerve_dp_analytics_api-1.0-py3-none-any.whl .
```

```
RUN pip install wheel nerve_dp_analytics_api-1.0-py3-none-any.whl

WORKDIR /

COPY demo_sensor_analytics_app.py .

CMD [ "python", "-u", "demo_sensor_analytics_app.py" ]
```

5. Enter the following command to build the Docker image containing the analytics app. `nerve-dp-2.1.1` is used as the name in this example:

   ```
   nerve-analytics build -t nerve-dp-2.1.1
   ```

6. Enter the following command to provision the analytics app as a Docker workload in the Management System:

   ```
   nerve-analytics provision -u https://<MS-URL> -n "Data Services Analytics - demoSe
   ```

   This will provision a Docker workload with the following settings:

Setting	Description
Workload name	Data Services Analytics - demoSensor App
Description	Docker container running a Nerve Data Services analytics app that processes temperature and humidity data.
Version name	nerve-dp-2.1.1
Release name	nerve-dp-2.1.1
CPU resource in percentage	1
Container name	analytics-demo-sensor-app
Network name	nerve-dp

   > **NOTE**
   >
   > Due to version differences, the workload is created with a maximum of 1% of allowed CPU usage. Change this setting to a value between 10 and 25.

All settings except **Container name** and **Network name** in the command above or in the Management System are suggestions and can be changed freely.

With the analytics app provisioned in the Management System, the app needs to be deployed to the node to analyze data coming from the demoSensor. Deploy the app to the node that has the demoSensor and the MQTT broker deployed by following Deploying a workload.

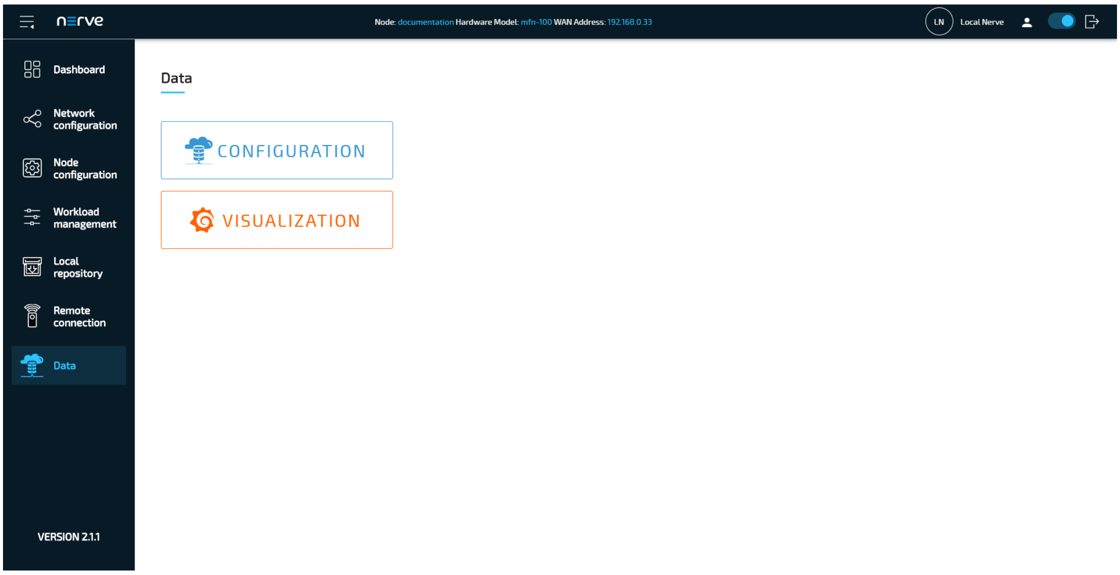
Configuring the Data Services Gateway on the node

Now that the demoSensor, the MQTT broker and the analytics app are deployed on the node, the Gateway instance on the node must be configured next in order for data to be processed and visualized.

1. Access the Local UI on the node. This is Nerve Device specific. Refer to the table below for device specific links to the Local UI. The login credentials to the Local UI can be found in the customer profile.

Nerve Device	Physical port	Local UI
MFN 100	P1	http://172.20.2.1:3333
Kontron KBox A-150-APL	LAN 1	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-150-APL chapter of the device guide.
Kontron KBox A-250	ETH 2	`<wanip>:3333` To figure out the IP address of the WAN interface, refer to Finding out the IP address of the device in the Kontron KBox A-250 chapter of the device guide.
Siemens SIMATIC IPC127E	X1 P1	http://172.20.2.1:3333
Siemens SIMATIC IPC427E	X1 P1	http://172.20.2.1:3333
Vecow SPC-5600-i5-8500	LAN 1	http://172.20.2.1:3333

2. Select **Data** in the navigation on the left.

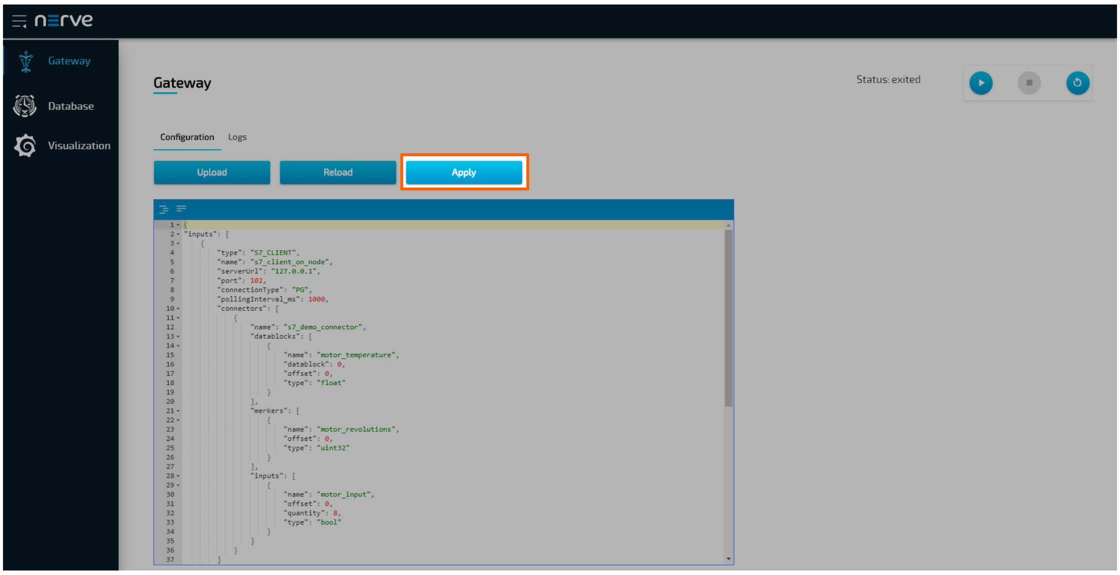


3. Select **CONFIGURATION**.

4. Enter the following configuration into the configuration editor of the Gateway:

```
{
  "inputs": [
    {
      "type": "MQTT_SUBSCRIBER",
      "name": "mqtt_subscriber",
      "clientId": "mqtt_subscriber_0",
      "serverUrl": "tcp://localhost:1883",
      "keepAliveInterval_s": 20,
      "cleanSession": false,
      "qos": 1,
      "connectors": [
        {
          "name": "mqtt_subscriber_connector_0",
          "topic": "demo-sensor-topic",
          "variables": [
            {
              "name": "temperature",
              "type": "int16"
            },
            {
              "name": "humidity",
              "type": "uint16"
            }
          ]
        }
      ]
    }
  ],
  "outputs": [
    {
      "type": "ZEROMQ_PUBLISHER",
      "name": "zeromq_publisher_0",
      "serverUrl": "tcp://172.20.10.1:5555",
      "connectors": [
        {
          "name": "zeromq_publisher_connector_0",
          "topic": "demo-sensor-topic",
          "timestampRequired": true,
          "timestampFormat": "unix_ns"
        }
      ]
    },
    {
      "type": "DB_TIMESCALE",
      "name": "timescaledb_0",
      "url": "<LOCAL>"
    }
  ],
  "connections": [
    {
      "name": "mqttsub_zmqpub_0",
      "input": {
        "index": 0,
        "connector": 0
      },
      "output": {
        "index": 0,
        "connector": 0
      }
    },
    {
      "name": "mqttsub_timescaledb_0",
      "input": {
        "index": 0,
        "connector": 0
      },
      "output": {
        "index": 1,
        "connector": 0
      }
    }
  ]
}
```

5. Select **Apply** to save the Gateway configuration. The Gateway will restart automatically.



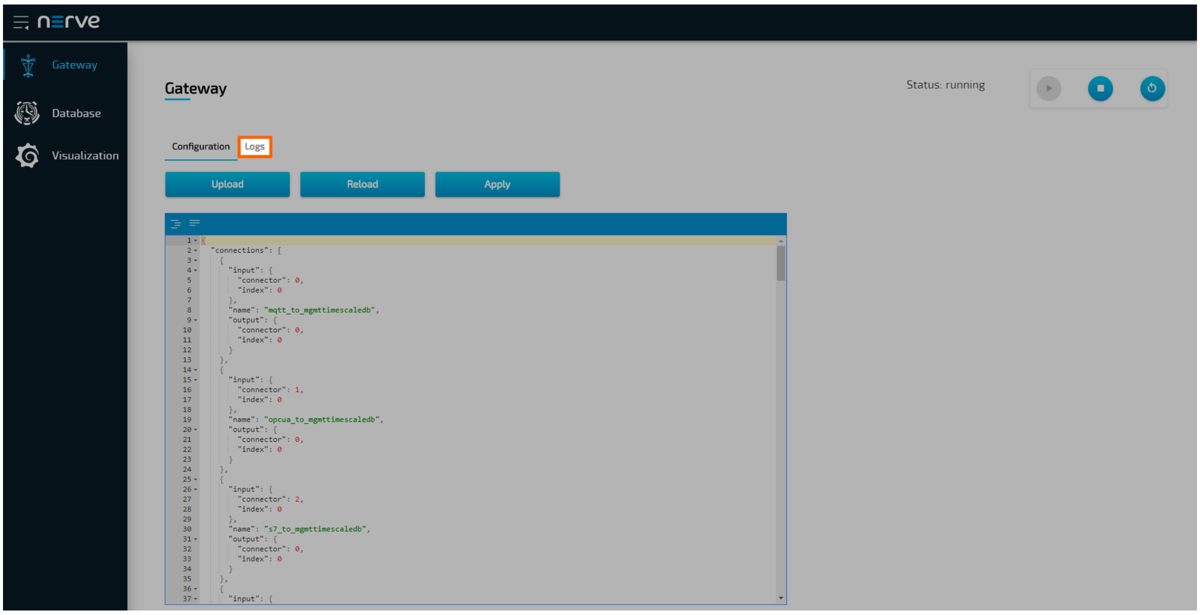
```
{
"inputs": [
    {
        "type": "S7_CLIENT",
        "name": "s7_client_on_node",
        "serverUrl": "127.0.0.1",
        "port": 102,
        "connectionType": "PG",
        "pollingInterval_ms": 1000,
        "connectors": [
            {
                "name": "s7_demo_connector",
                "datablocks": [
                    {
                        "name": "motor_temperature",
                        "datablock": 0,
                        "offset": 0,
                        "type": "float"
                    }
                ],
                "merkers": [
                    {
                        "name": "motor_revolutions",
                        "offset": 0,
                        "type": "uint32"
                    }
                ],
                "inputs": [
                    {
                        "name": "motor_input",
                        "offset": 0,
                        "quantity": 8,
                        "type": "bool"
                    }
                ]
            }
        ]
```

This configuration defines an MQTT Subscriber connection to a ZeroMQ Publisher. Upon receiving data from the `demo-sensor-topic` topic of the MQTT Subscriber, the Gateway forwards said data to the ZeroMQ Publisher. The ZeroMQ Publisher in turn publishes the data to the `demo-sensor-topic` ZeroMQ topic that the analytics app listens to. The configuration also defines a connection from the MQTT Subscriber to the TimescaleDB database, which means that the same data received at the MQTT Subscriber end is also written directly into the TimescaleDB.

NOTE

Note that both topics have the same name in this example. However, they are different as they are topics of two different protocols, MQTT and ZeroMQ.

To verify that the Gateway accepted the configuration and works as expected, select the **Logs** tab in the Gateway UI and check for errors.

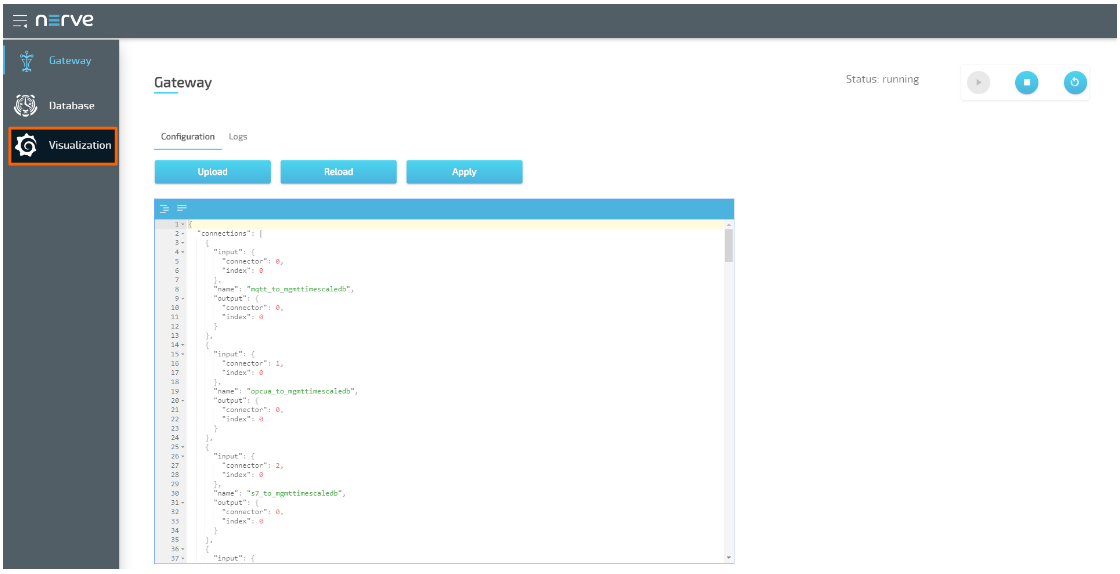


```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

Local data visualization at the node

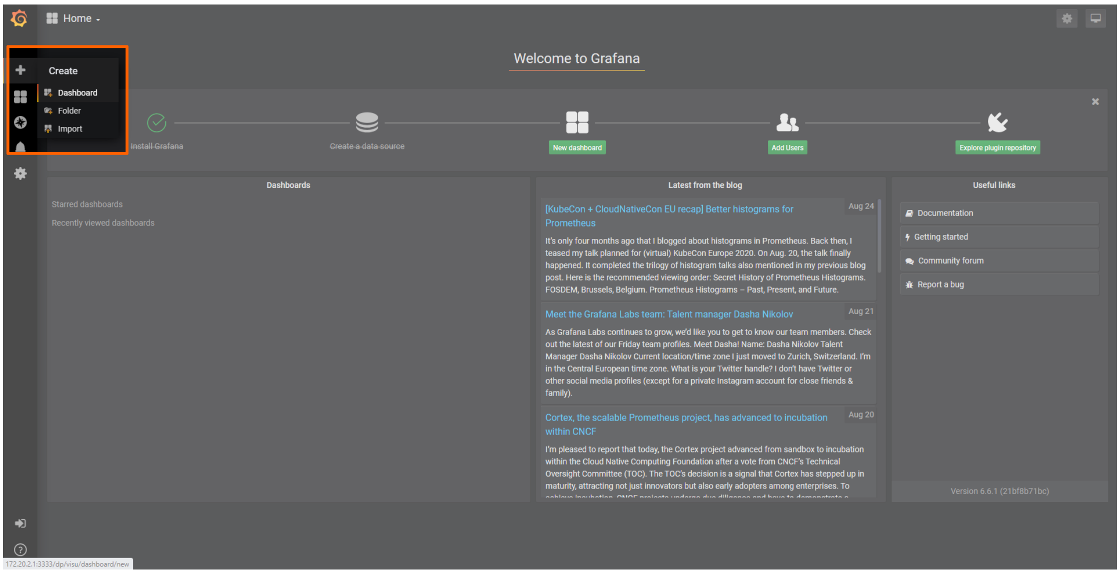
To visualize the data received by the Gateway and data processed by the analytics app, open the local data visualization element through the Data Services UI on the node. Two queries will be added in the instructions below.

1. Select **Visualization** in the navigation on the left. The Grafana UI will open.



```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

2. Select **+ > Dashboard** in the navigation on the left. A box will appear.



[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus — Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

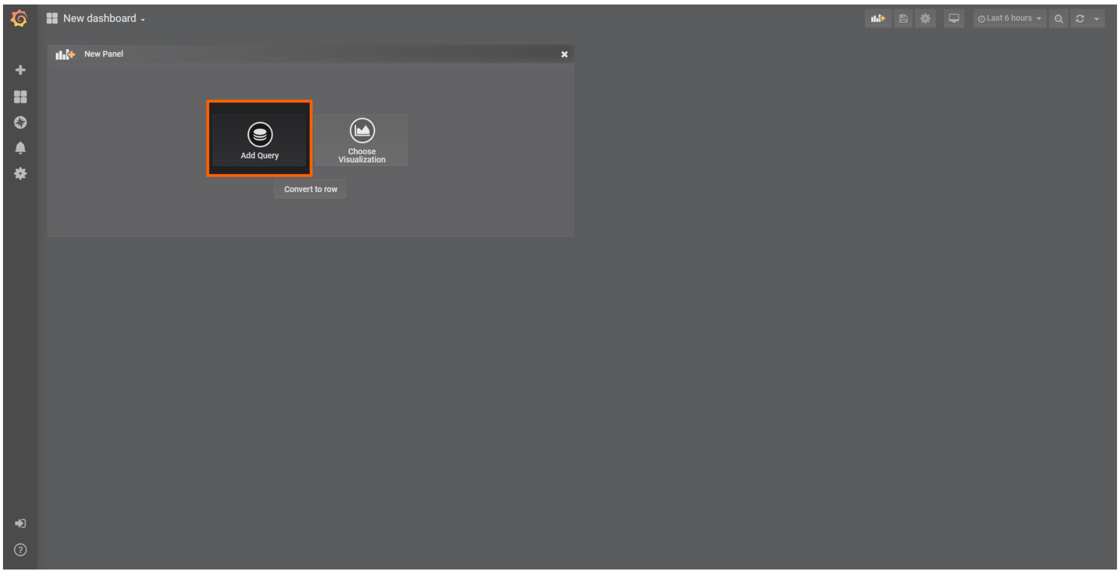
Meet the Grafana Labs team: Talent manager Dasha Nikolov — Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends & family).

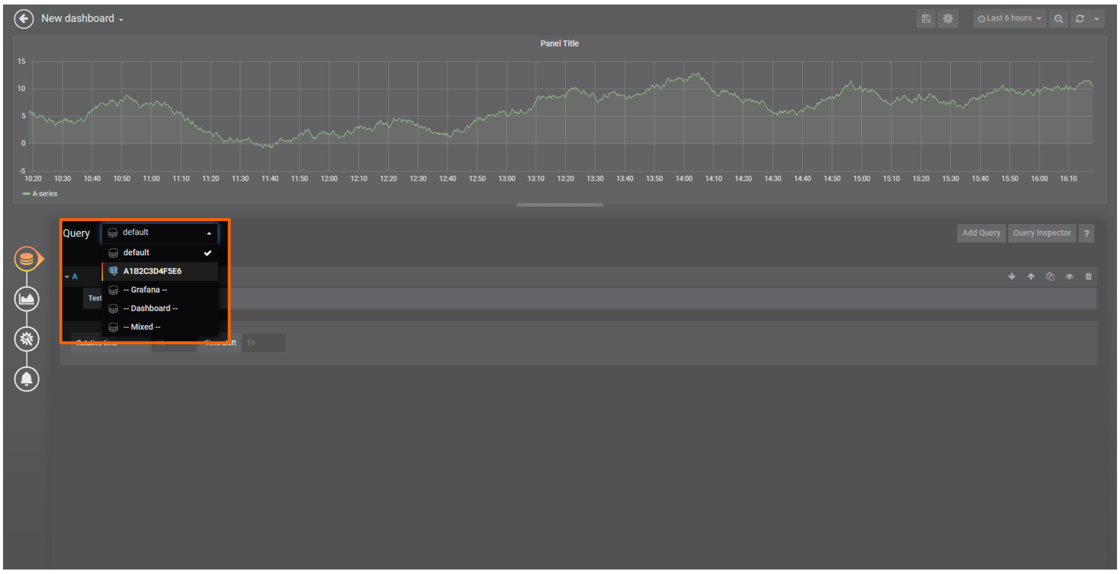
Cortex, the scalable Prometheus project, has advanced to incubation within CNCF — Aug 20

I'm pleased to report that today, the Cortex project advanced from sandbox to incubation within the Cloud Native Computing Foundation after a vote from CNCF's Technical Oversight Committee (TOC). The TOC's decision is a signal that Cortex has stepped up in maturity, attracting not just innovators but also early adopters among enterprises. To

3. Select **Add Query** in the **New Panel** box.



4. Select the data source from the drop-down menu. The name of the data source is the serial number of the node.



5. Fill in the following query information to add the temperature data from the MQTT Subscriber:

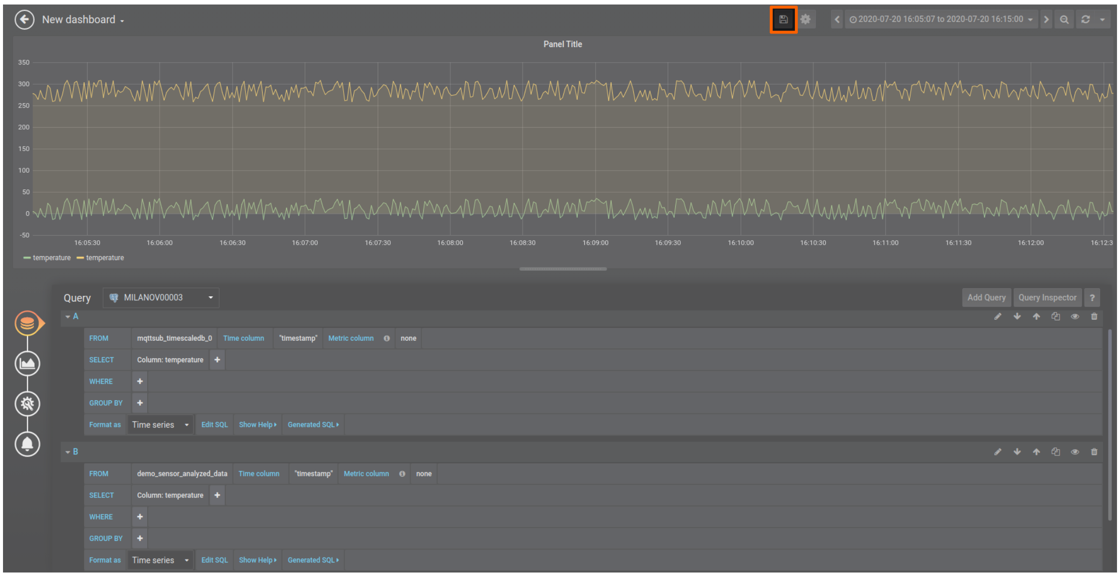
Setting	Value
FROM	`mqttsub_timescaledb_0` **Time column**: "timestamp"
SELECT	Column: temperature
Format as	Time series

6. Select **Add Query** to the right to add query B for temperature data analyzed by the analytics app.

7. Fill in the following query information:

Setting	Value
FROM	`demo_sensor_analyzed_data` **Time column**: "timestamp"
SELECT	Column: temperature
Format as	Time series

8. Select the save icon in the upper-right corner to save the dashboard.



Query MILANOV00003

A

FROM mqttsub_timescaledb_0 Time column "timestamp" Metric column none

SELECT Column: temperature

WHERE

GROUP BY

Format as Time series Edit SQL Show Help Generated SQL

B

FROM demo_sensor_analyzed_data Time column "timestamp" Metric column none

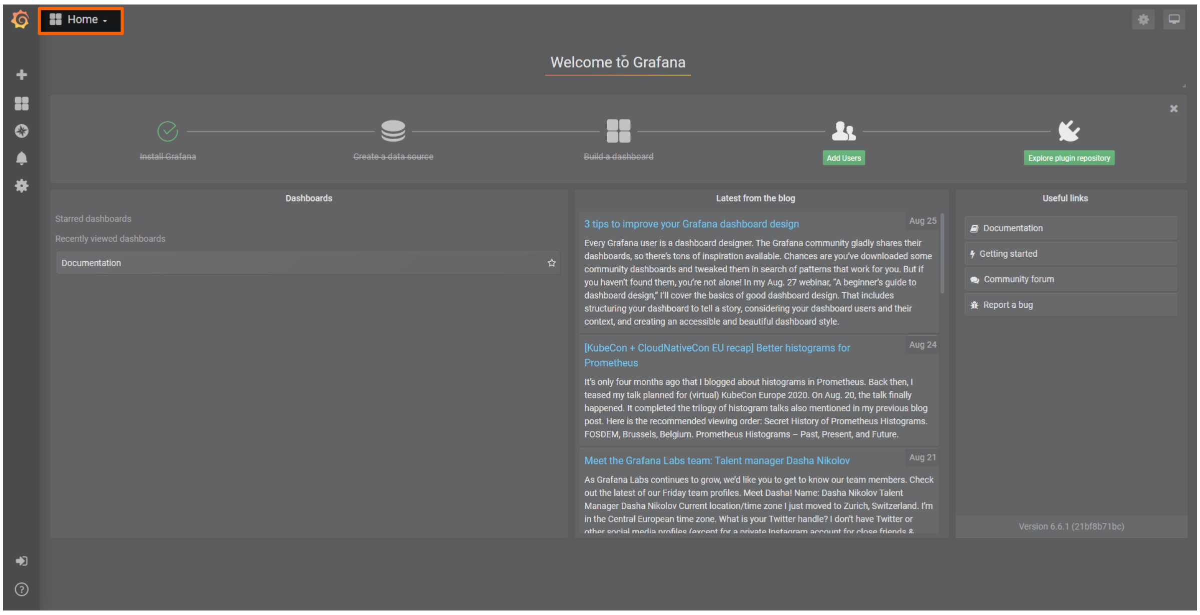
SELECT Column: temperature

WHERE

GROUP BY

Format as Time series Edit SQL Show Help Generated SQL

The dashboard can be accessed from the Grafana home menu.



3 tips to improve your Grafana dashboard design Aug 25

Every Grafana user is a dashboard designer. The Grafana community gladly shares their dashboards, so there's tons of inspiration available. Chances are you've downloaded some community dashboards and tweaked them in search of patterns that work for you. But if you haven't found them, you're not alone! In my Aug. 27 webinar, "A beginner's guide to dashboard design," I'll cover the basics of good dashboard design. That includes structuring your dashboard to tell a story, considering your dashboard users and their context, and creating an accessible and beautiful dashboard style.

[KubeCon + CloudNativeCon EU recap] Better histograms for Prometheus Aug 24

It's only four months ago that I blogged about histograms in Prometheus. Back then, I teased my talk planned for (virtual) KubeCon Europe 2020. On Aug. 20, the talk finally happened. It completed the trilogy of histogram talks also mentioned in my previous blog post. Here is the recommended viewing order: Secret History of Prometheus Histograms. FOSDEM, Brussels, Belgium. Prometheus Histograms – Past, Present, and Future.

Meet the Grafana Labs team: Talent manager Dasha Nikolov Aug 21

As Grafana Labs continues to grow, we'd like you to get to know our team members. Check out the latest of our Friday team profiles. Meet Dasha! Name: Dasha Nikolov Talent Manager Dasha Nikolov Current location/time zone I just moved to Zurich, Switzerland. I'm in the Central European time zone. What is your Twitter handle? I don't have Twitter or other social media profiles (except for a private Instagram account for close friends &

Node internal networking

This chapter explains how a user can connect workloads (VMs and Docker containers) to services and network ports of a node. In order to do this, it explains the internal networking concepts in detail. Most workloads will need to be connected to a network as networking is the main form of communication for workloads. They either want to connect to external servers or they are servers themselves, in which case they need to be made visible for their communication partners. The Nerve networking system enables both use cases.

The image below shows an example node consisting of the **host/domain-0** and the real-time VM running the CODESYS runtime (labeled **rtvm**). To further clarify the networking example it also has one Virtual Machine workload and two Docker workloads deployed. The virtual machine is depicted outside of the host and the Docker containers are depicted in the Docker network inside of the host. For the sake of the example, however, the workloads are not yet connected. This is done in the examples further below.

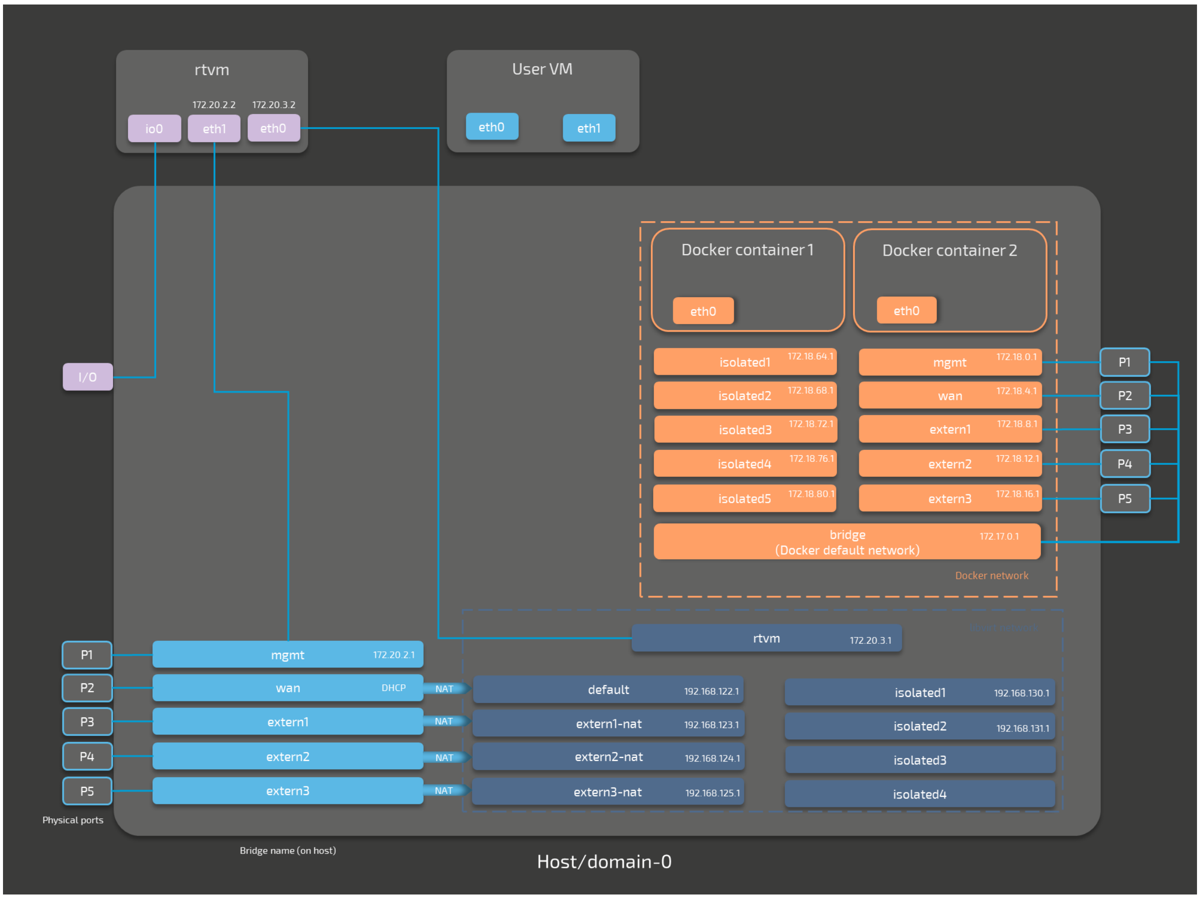
The physical ports **P1** to **P5** and **I/O** of the Nerve Device (the MFN 100 in this case) are displayed on the left, touching the large dark rectangle that represents the host. The light blue interfaces connected to them inside the host are Linux bridged interfaces displayed with their names on the host. Highlighted by a dark blue dashed frame is the libvirt network with NAT interfaces on the left and isolated interfaces on the right. Slightly above them is the **rtvm** interface for communication with the RTVM. The system is setup so that the interfaces connected to the physical ports can be reached by connecting through the physical ports.

Highlighted by an orange dashed frame is the Docker network including the default Docker network (the orange interface labeled **bridge**), as well as Docker network equivalents of the Linux bridged interfaces, with one additional isolated network in comparison. For the sake of easier representation, the physical ports **P1** to **P5** are duplicated to the right of the Docker network, again touching the host. This is done to show that Docker network interfaces can also be reached directly by connecting to the physical ports while also making sure to show that the libvirt network and the Docker network are separate from each other.

All interfaces colored in purple are related to the RTVM. Interfaces labeled **eth** are symbolic representations of interfaces that are used by virtual machines and Docker containers for communication with the Nerve Blue system. The actual used interfaces depend on the Docker container or virtual machine.

Connections are displayed in three ways. Blue lines are connections that are predefined by the system. Blue arrows are used between bridged interfaces and the libvirt network to indicate NAT. Further down below, green lines are used as example connections that can be done by the user.

As mentioned above, the image below represents the MFN 100. Refer to the device guide for information on the Nerve Device as the physical ports and the connection to their respective interfaces differ.



See the table below for more information on the interfaces, their usage and their IP ranges.

Legend	
Physical Ports	The physical ports are device dependent. They are included here for clarification of the image above. The MFN 100 is used as an example. Refer to the device guide for information on the specific hardware model of the Nerve Device. • **P1 to P5** Ethernet ports of the MFN 100. Note that P5 is an SFP port. • **I/O** While also an Ethernet port, this port is reserved for communication of the RTVM with the fieldbus.
Bridged Interfaces	• **mgmt** This interface is mainly used for the configuration of the system. It is connected to the **eth1** interface of the RTVM. • **wan** This interface is designated for internet connection and configured as DHCP client by default. • **extern1, extern2, extern3** These interfaces can be used for customer specific external connections.

NAT Interfaces	If a deployed virtual machine uses one of the predefined NAT interfaces, the IP address of the respective interface is assigned by a DHCP server with a subnet mask of `255.255.255.0`. The DHCP pool contains the upper half of the respective address space, e.g. `192.168.122.128` to `192.168.122.254`.
	• **default**
	This network is the NAT interface of the **wan** bridged interface with an IP address in the range from `192.168.122.128` to `192.168.122.254`.
	• **extern1-nat, extern2-nat, extern3-nat**
	These networks are the NAT interfaces of the **extern1**, **extern2** and **extern3** bridged interfaces. The IP addresses of these interfaces are in the following ranges:
	◦ **extern1-nat**
	`192.168.123.128` to `192.168.123.254`
	◦ **extern2-nat**
	`192.168.124.128` to `192.168.124.254`
	◦ **extern3-nat**
	`192.168.125.128` to `192.168.125.254`

Legend	
Isolated Interfaces	Isolated interfaces can be used to allow communication between two virtual machines. These networks cannot communicate outside of the system. • **isolated1** This interface has a DHCP server, meaning that a VM can configure a static or dynamic IP address in the range from `192.168.130.2` to `192.168.130.254`. • **isolated2** This interface has a DHCP server, meaning that a VM can configure a static or dynamic IP address in the range from `192.168.131.2` to `192.168.131.254`. • **isolated3** This network does not have a DHCP server. • **isolated4** This network does not have a DHCP server.

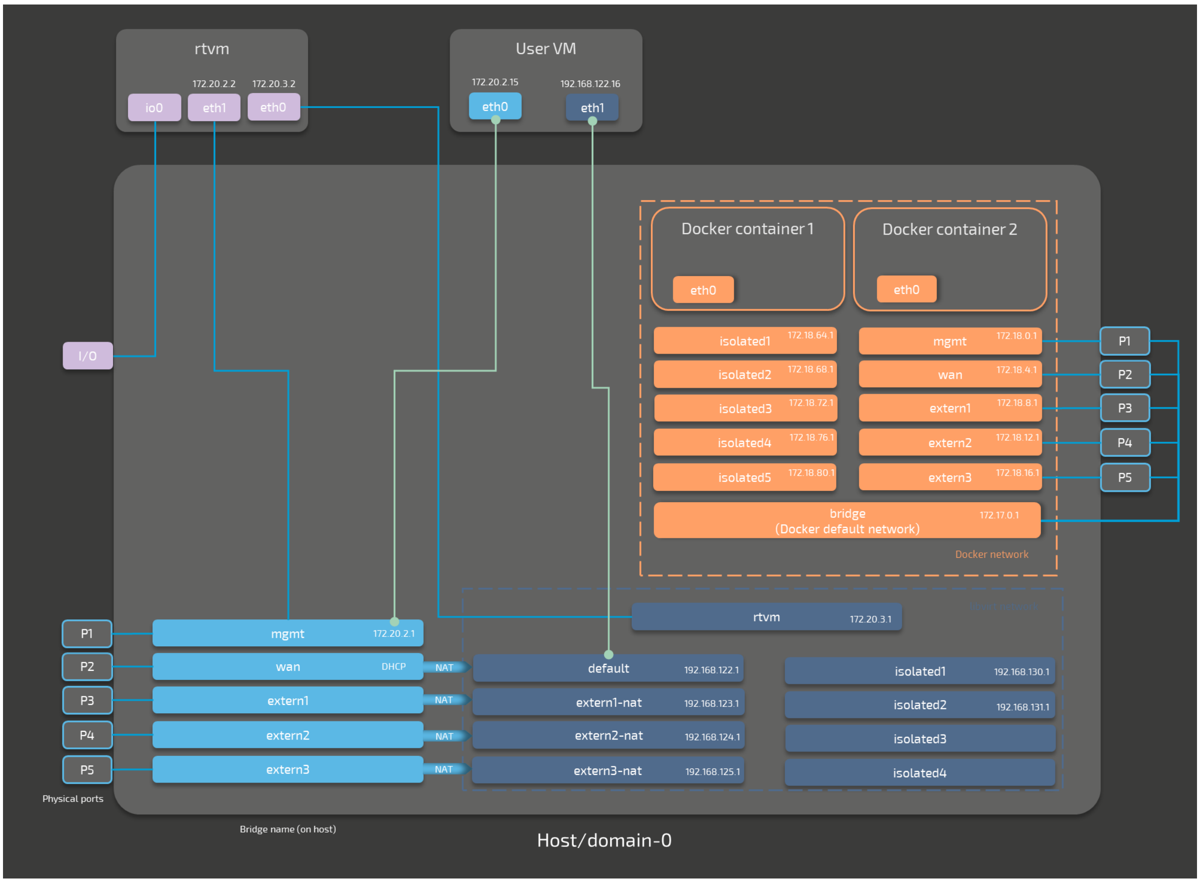
Docker network	The Docker network includes the default Docker network **bridge** and the Docker network equivalents of the Linux bridged interfaces, with one additional isolated network in comparison.
	• **bridge**
	This is the default Docker network designated for communication with Docker. Interfaces of Docker containers that are connected to the gateway receive IP addresses in the default Docker network, ranging from `172.17.0.2` to `172.17.0.244`. By default, this interface can be reached through any physical port (excluding **io0**).
	• **mgmt, wan, extern1, extern2, extern3**
	These interfaces are the Docker network equivalents of the bridged interfaces. They can be reached through the physical ports **P1** to **P5**. The IP addresses of these interfaces are in the following ranges:
	◦ **mgmt**
	`172.18.0.2` to `172.18.0.254`
	◦ **wan**
	`172.18.4.2` to `172.18.4.254`
	◦ **extern1**
	`172.18.8.2` to `172.18.8.254`
	◦ **extern2**
	`172.18.12.2` to `172.18.12.254`
	◦ **extern3**
	`172.18.16.1` to `172.18.16.254`
	• **isolated1, isolated2, isolated3, isolated4, isolated5**
	Similar to the bridged interfaces, these interfaces cannot be reached outside of the system. The IP addresses of these interfaces are in the following ranges:
	◦ **isolated1**
	`172.18.64.2` to `172.18.64.254`
	◦ **isolated2**
	`172.18.68.2` to `172.18.68.254`
	◦ **isolated3**
	`172.18.72.2` to `172.18.72.254`
	◦ **isolated4**
	`172.18.76.2` to `172.18.76.254`
	◦ **isolated5**
	`172.18.80.2` to `172.18.80.254`

Legend	
Other interfaces	Other interfaces that can be used as a NAT network but without port forwarding. These interfaces do not communicate outside of the system. • **rtvm** Use this interface for communication with the RTVM. • **eth** These interfaces are symbolic representations of interfaces that are used by virtual machines and Docker containers for communication with the Nerve Blue system. The actual interfaces used depend on the Docker container or virtual machine. The only exceptions are **eth0** and **eth1** for the RTVM as they are always defined with these names. • **io0** This interface is defined for communication between the I/O port and the RTVM.
Connections	• **Blue lines** Blue lines signify connections that are predefined by the Nerve Blue system. • **Green lines** Green lines are example connections that could be made by a user to connect virtual machines or Docker containers to the network. • **Blue arrows** Blue arrows indicate a NAT between bridged interfaces and the libvirt network.

The following sections are conceptual explanations. Workloads are attached to internal networks during the provisioning process. Refer to the provisioning chapters (Virtual Machine workloads and Docker workloads) in the user guide on how to provision workloads.

Attaching virtual machines to a network

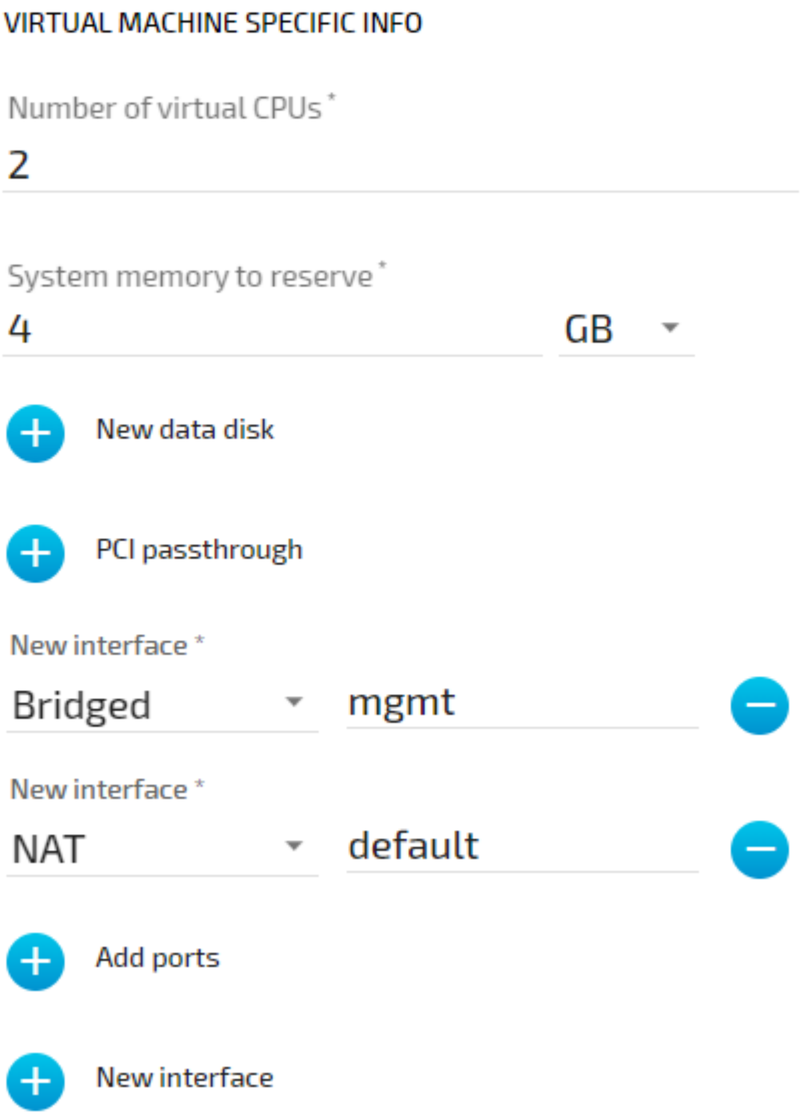
Virtual machine networking is comparable to installing a network card in the virtual machine and attaching it to the network with the network name given in the network drawing. For this example, there are two "network cards" installed in a user deployed virtual machine. They are located in the **User VM** and are labeled **eth0** and **eth1** in this example. Green lines indicate a user established connection.



There are two connections established here: **eth0** of the **User VM** is connected to the **mgmt** bridged interface for communication with the RTVM inside of the system. **eth1** of the **User VM** is connected to the **default** NAT interface for an internet connection protected by NAT on **P2** of the Nerve Device. Both interfaces have IP addresses in the designated ranges. `172.20.2.15` for **eth0** was manually configured in the virtual machine and `192.168.122.16` for **eth1** was assigned by the DHCP server.

Settings example

To achieve the functionality above, configure the interfaces of the Virtual Machine workload the following way during the provisioning process in the Management System:



VIRTUAL MACHINE SPECIFIC INFO

Number of virtual CPUs *: 2

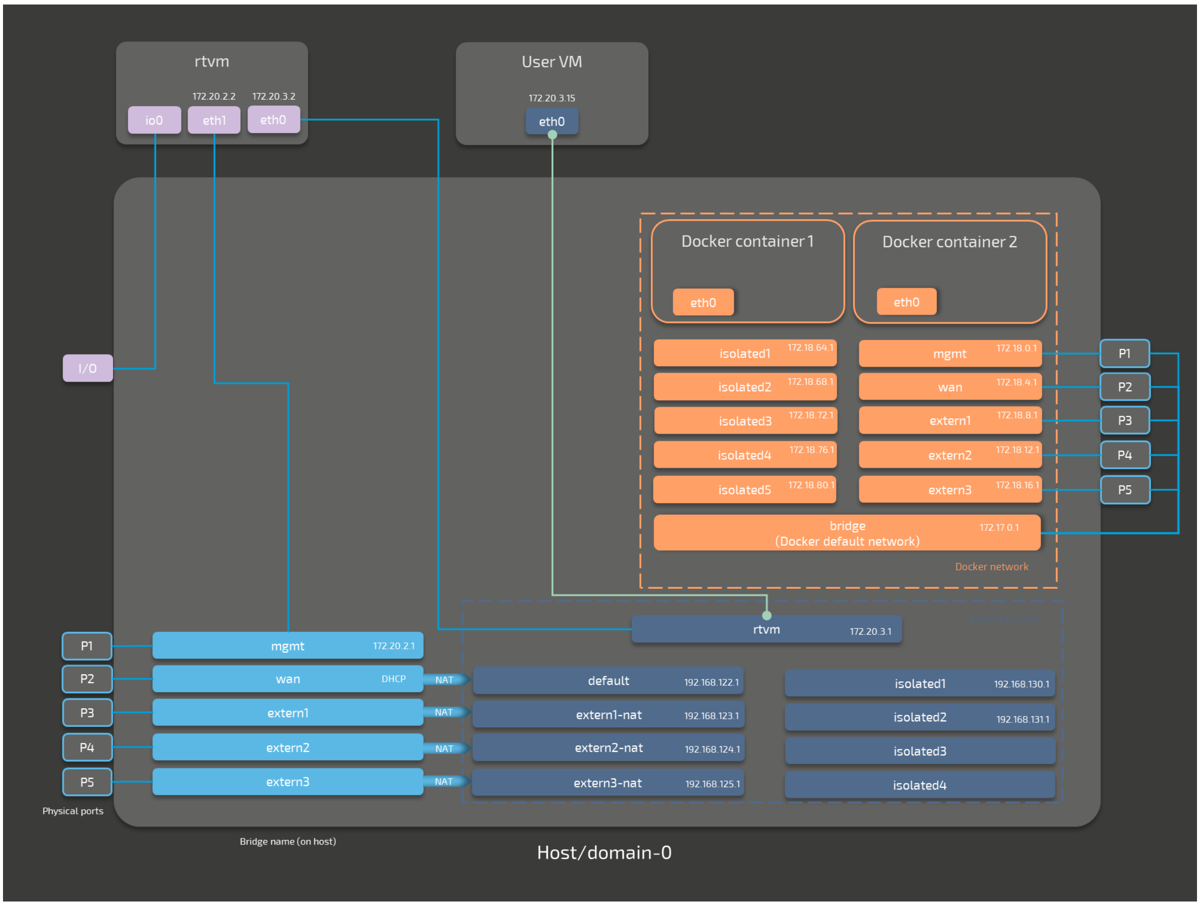
System memory to reserve *: 4 GB

New interface *: Bridged mgmt

New interface *: NAT default

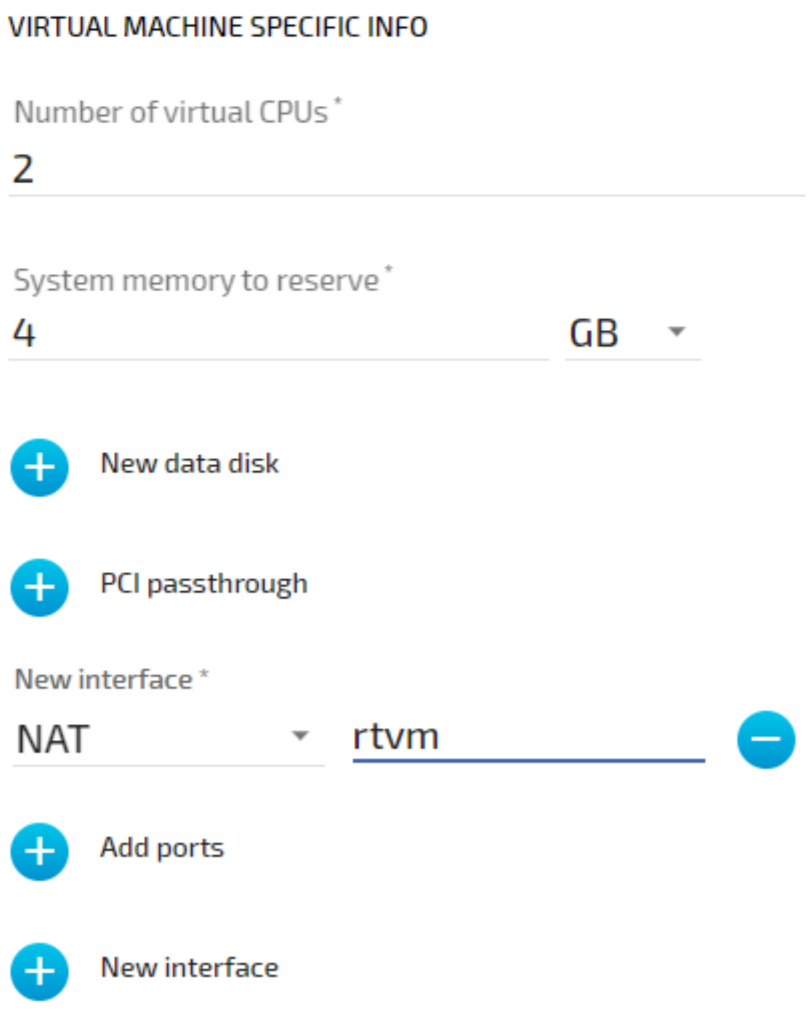
Communication of a virtual machine with the RTVM

A virtual machine can communicate with the RTVM by connecting an interface to the bridged interface **rtvm**. In the example below this is done with the interface **eth0** of the **User VM** that has the IP address `172.20.3.15`. The IP address was manually configured.



Settings example

To achieve the functionality above, configure the interfaces of the Virtual Machine workload the following way during the provisioning process in the Management System:



VIRTUAL MACHINE SPECIFIC INFO

Number of virtual CPUs *
2

System memory to reserve *
4 GB

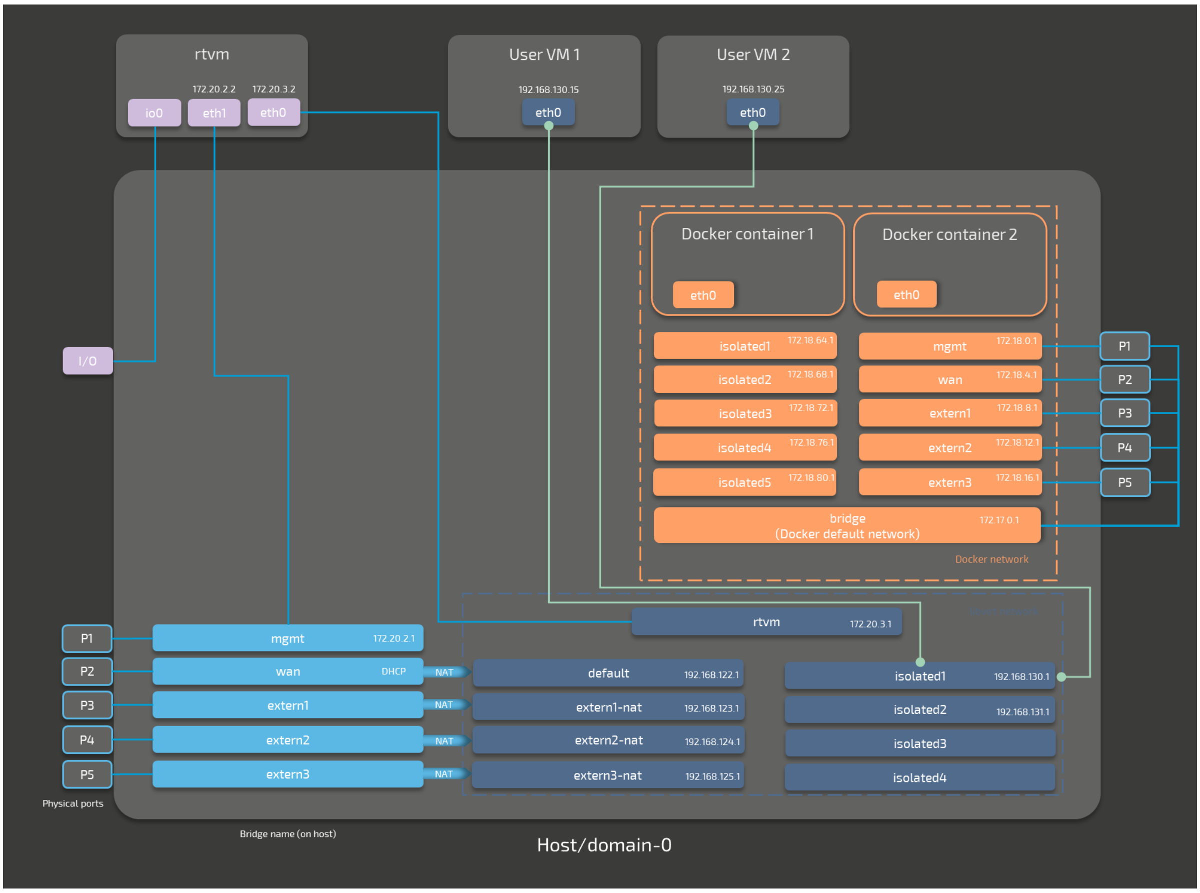
New data disk

PCI passthrough

New interface *
NAT rtvm

Communication of two virtual machines through isolated networks

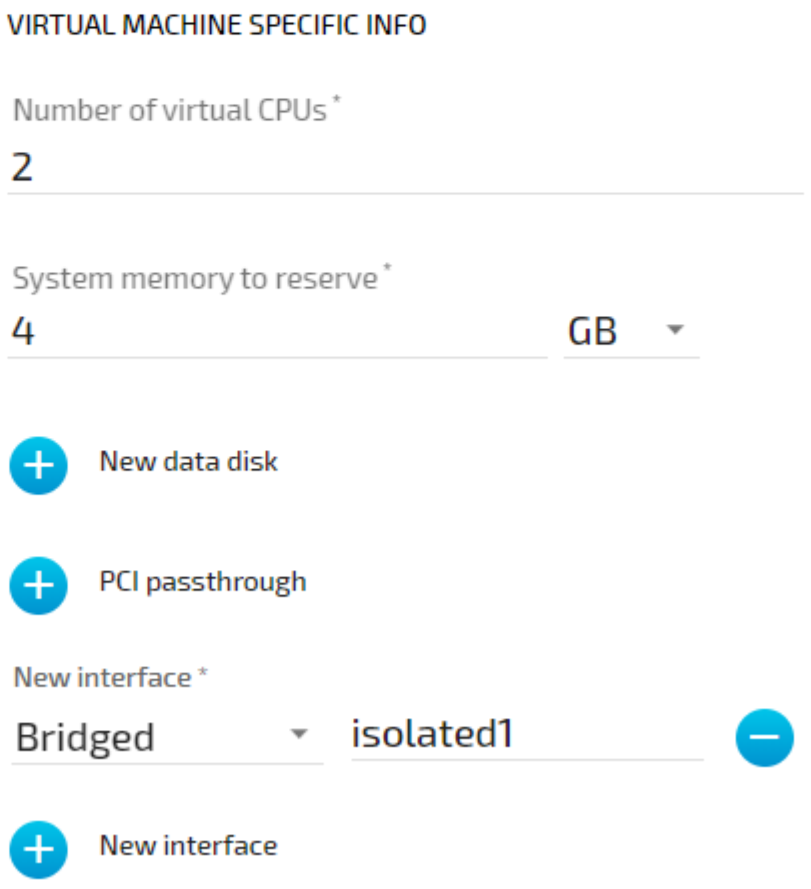
Nerve Blue offers isolated network interfaces for communication of workloads inside of the system. These interfaces do not communicate outside of the system. They can be used to establish communication between two Virtual Machines.



Both virtual machines have a "network card" installed. **User VM 1** is connected to the **isolated1** interface through **eth0** and **User VM 2** is connected through its interface **eth0** to the same network interface, **isolated1**. Each interface has been assigned an IP address by a DHCP server in the designated range: `192.168.130.15` for **eth0** of **User VM1** and `192.168.130.25` for **eth0** of **User VM 2**.

Settings example

To achieve the functionality above, configure the interfaces of both Virtual Machine workloads the following way during the provisioning process in the Management System:



VIRTUAL MACHINE SPECIFIC INFO

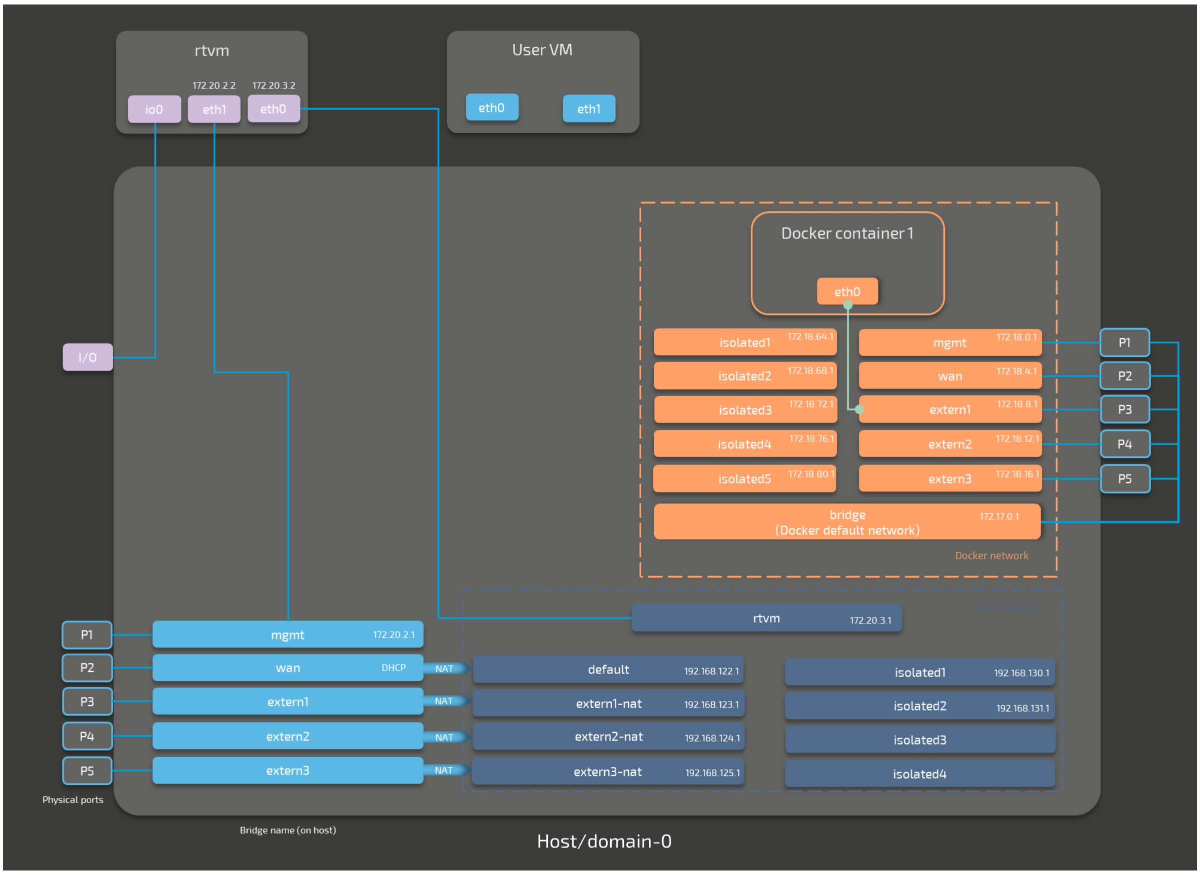
Number of virtual CPUs *
2

System memory to reserve *
4 GB

New interface *
Bridged isolated1

Communication of a Docker container outside of the system

For Docker containers the situation is different. Docker containers can be attached to the Docker default network or respective Docker network interfaces to access other parts of the system or communicate outside of the system. The Docker default network is called **bridge** and has the IP address `172.17.0.1` assigned. This is interface is available on all physical ports (here **P1** to **P5**). For this example, the Docker container will be connected to the **extern1** interface. In order to make a server accessible for other workloads, map the port and protocol of the Docker container to the outside by specifying the network name, here **extern1**, during workload provisioning in the port mapping section.



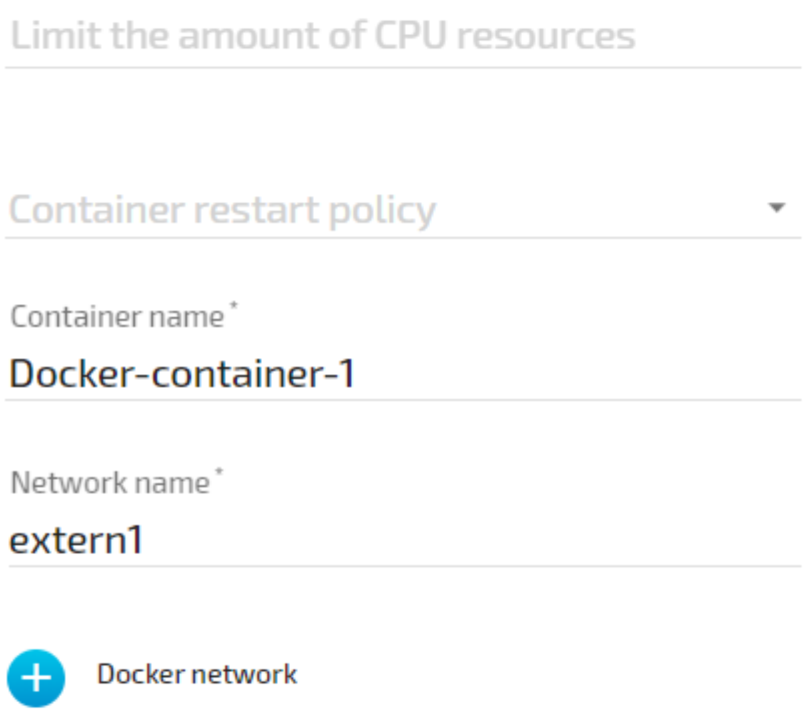
The Docker container is connected to the **extern1** interface in the Docker network and available at **P3** outside of the system at an IP address in the range from `172.18.8.2` to `172.18.8.254`.

NOTE

Note that the Docker default network **bridge** is always defined as an interface by default. That means all deployed Docker workloads can be reached through the **bridge** interface.

Settings example

To achieve the functionality above, configure the Docker network name the following way during the provisioning process in the Management System:



Limit the amount of CPU resources

Container restart policy

Container name *
Docker-container-1

Network name *
extern1

List of reserved TCP/UDP ports

In general, Nerve Blue reserves the port range 47200 — 47399 on both TCP and UDP for internal usage. The following list states ports that are reserved in version 2.1.

Port	Interface	Protocol	Reserved for
22	none	TCP	SSH daemon
3333	172.20.2.1	TCP	Local UI
47200	127.0.0.1	TCP/UDP	System Log
47201	127.0.0.1	UDP	Filebeat
47300	127.0.0.1	TCP	Local MQTT broker
47301	127.0.0.1	TCP	Local MQTT broker

First steps with CODESYS

NOTE

This chapter uses the MFN 100 as an example. However, the data below is valid for all Nerve Devices except the Kontron Kbox A-150-APL and the Kontron Kbox A-250, as CODESYS application development is not possible with these devices.

This chapter will give an introduction on how to start working with the integrated soft PLC in the MFN 100. First, some configuration and installation of files and libraries are required.

NOTE

- Download the CODESYS Development System V3 from store.codesys.com for this chapter.
  We recommend version 3.5 SP14 (32 bit) or newer.
- Connect the workstation to the console port **P1** of the MFN 100.

Installing the device descriptions

After downloading and installing the CODESYS Development System on the workstation, install the device description of the MFN 100 in the CODESYS Development System. The device description has the following filename:

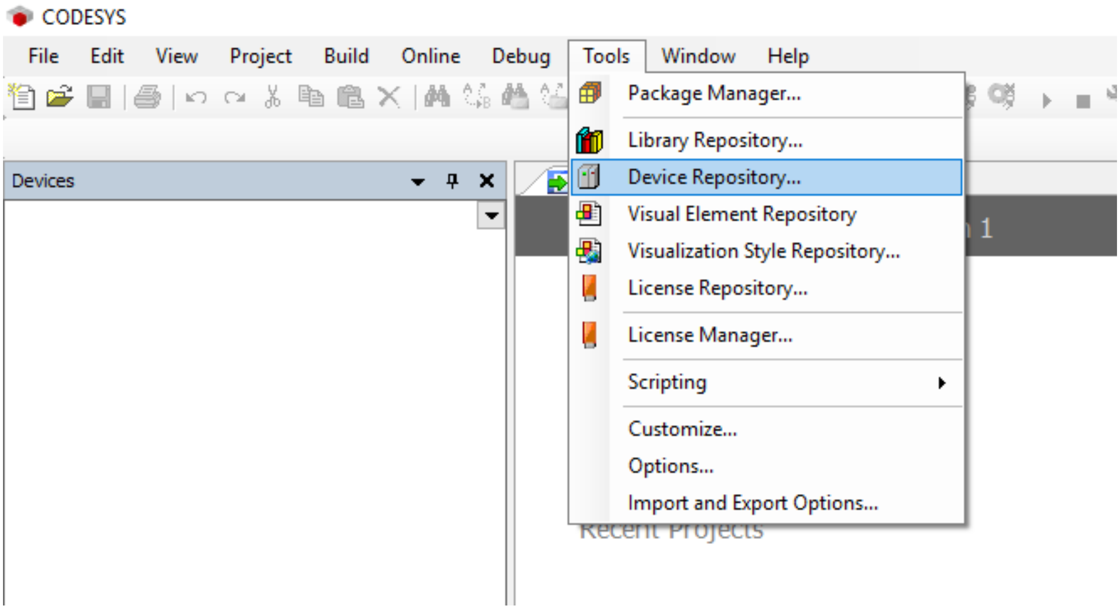
- Nerve_MFN_100_V3.5.XX.X.devdesc.xml

NOTE

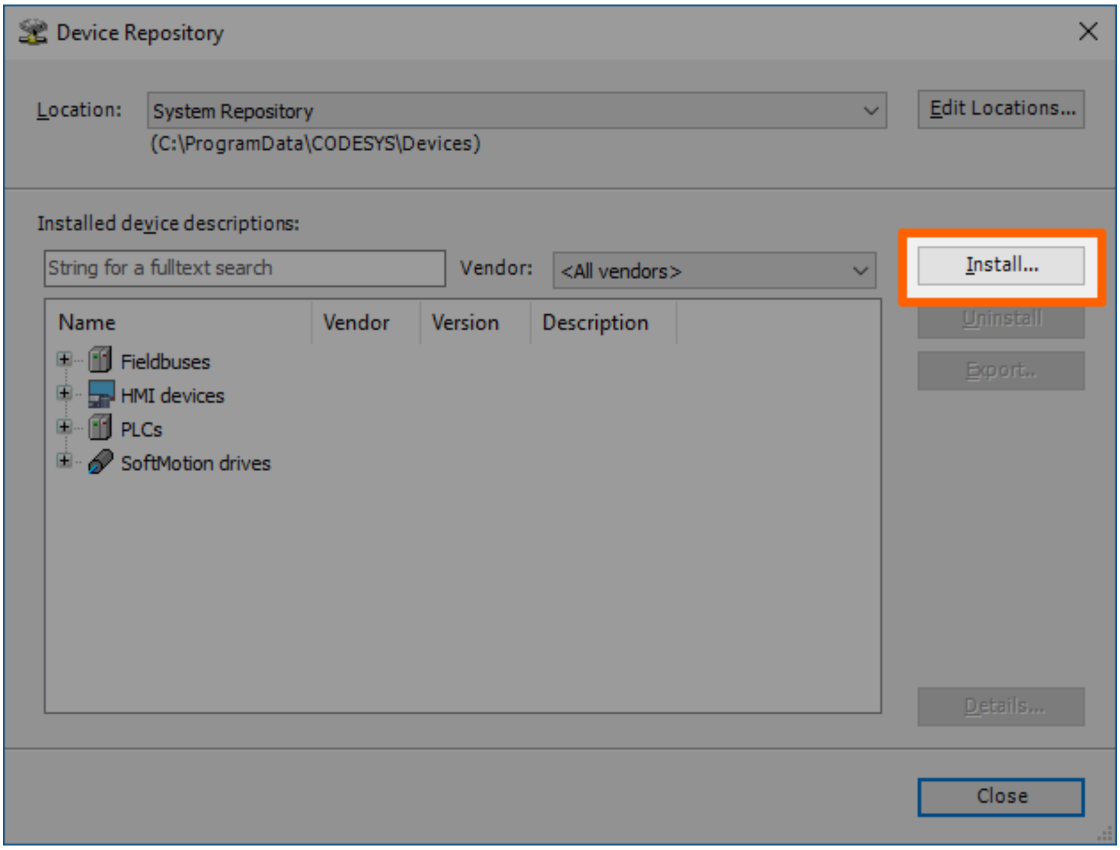
XX.X stands for the current version of the CODESYS Development System

The device description of the MFN 100 is available at the Nerve Software Center. Remember where the device description is saved for the following steps.

1. Start the CODESYS Development System.
2. Go to **Tools > Device Repository**.



3. Click **Install**.



Location: System Repository (C:\ProgramData\CODESYS\Devices)

Installed device descriptions:

String for a fulltext search

Vendor: <All vendors>

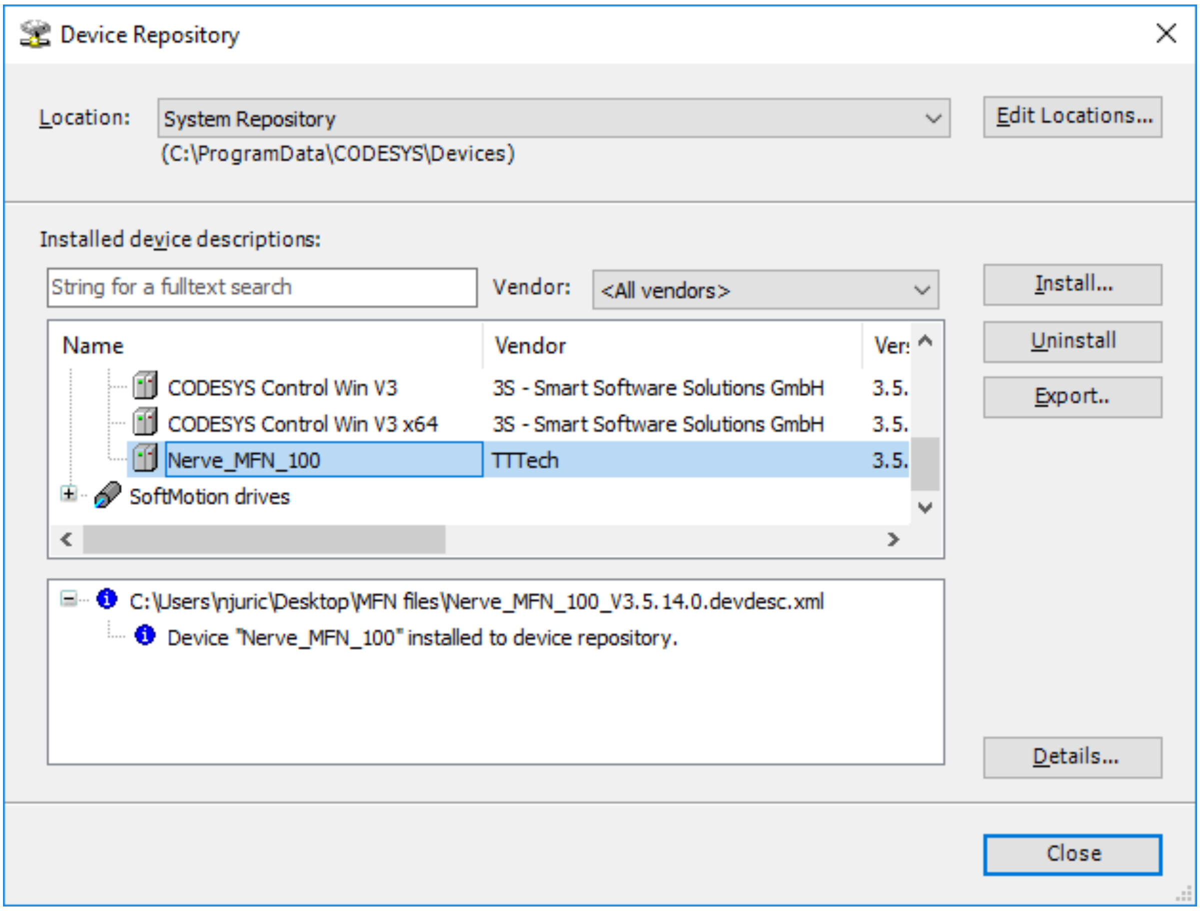
4. Go to the directory of the previously downloaded device description.

5. Select the device description of the MFN 100.

   The device description will look like this: `Nerve_MFN_100_V3.5.XX.X.devdesc.xml`

6. Click **Open**.

When the installation was successful, the MFN 100 will appear in the list of device descriptions in the middle of the window.



Location: System Repository (C:\ProgramData\CODESYS\Devices)

Name	Vendor	Ver:
CODESYS Control Win V3	3S - Smart Software Solutions GmbH	3.5.
CODESYS Control Win V3 x64	3S - Smart Software Solutions GmbH	3.5.
Nerve_MFN_100	TTTech	3.5.
SoftMotion drives		

C:\Users\njuric\Desktop\MFN files\Nerve_MFN_100_V3.5.14.0.devdesc.xml

Device "Nerve_MFN_100" installed to device repository.

After installing the device description, work with the CODESYS Development System can be started. However, libraries and device descriptions of generic devices might be missing so that the CODESYS Development System can work properly. The following chapters cover the download process.

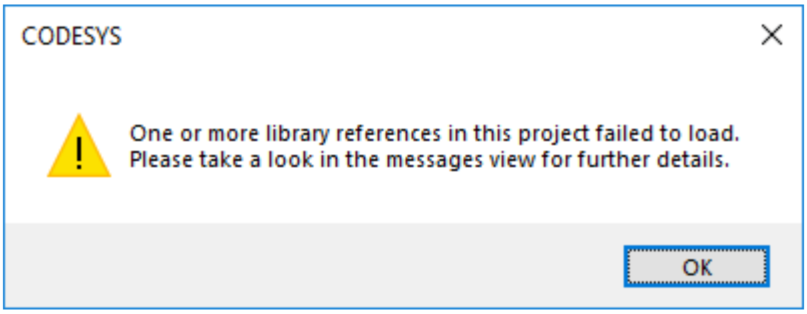
NOTE

The device description might need to be updated if this is not the first time working with MFN 100 and the CODESYS Development System:

1. Follow the steps above to install the newest device description.
2. Right-click **Device (Nerve_MFN_100)** on the left side.
3. Select **Update Device....**
4. Select the current device description in the new window.
5. Click **Update Device** in the lower-right.

Downloading missing libraries

The error message for missing libraries might appear when opening or creating a CODESYS project. The CODESYS Development System identifies the missing libraries automatically but the following process might need to be repeated a few times.

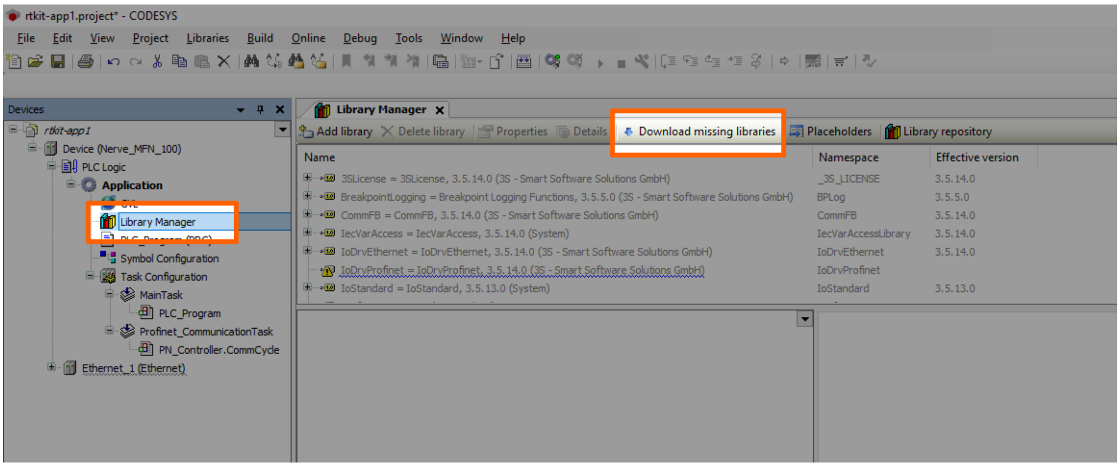


CODESYS

One or more library references in this project failed to load.
Please take a look in the messages view for further details.

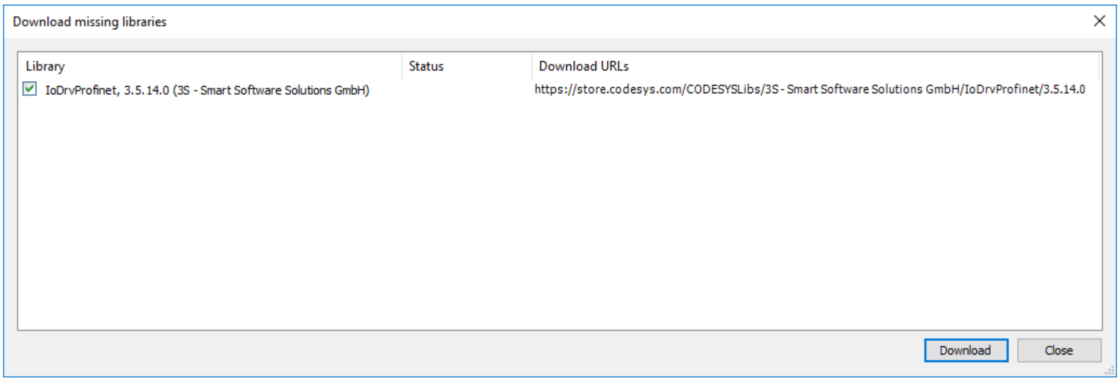
1. Open or create a CODESYS project.
2. If the error message about missing libraries appears, click **OK**.
3. Double-click **Library Manager** in the tree view on the left.

4. Click **Download missing libraries**.



Name	Namespace	Effective version
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.14.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CommFB = CommFB, 3.5.14.0 (3S - Smart Software Solutions GmbH)	CommFB	3.5.14.0
IecVarAccess = IecVarAccess, 3.5.14.0 (System)	IecVarAccessLibrary	3.5.14.0
IoDrvEthernet = IoDrvEthernet, 3.5.14.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	3.5.14.0
IoDrvProfinet = IoDrvProfinet, 3.5.14.0 (3S - Smart Software Solutions GmbH)	IoDrvProfinet	
IoStandard = IoStandard, 3.5.13.0 (System)	IoStandard	3.5.13.0

5. Click **Download** in the new window.



Download missing libraries

Library	Status	Download URLs
☑ IoDrvProfinet, 3.5.14.0 (3S - Smart Software Solutions GmbH)		https://store.codesys.com/CODESYSLibs/3S - Smart Software Solutions GmbH/IoDrvProfinet/3.5.14.0

6. Click **Close** when the download is finished.

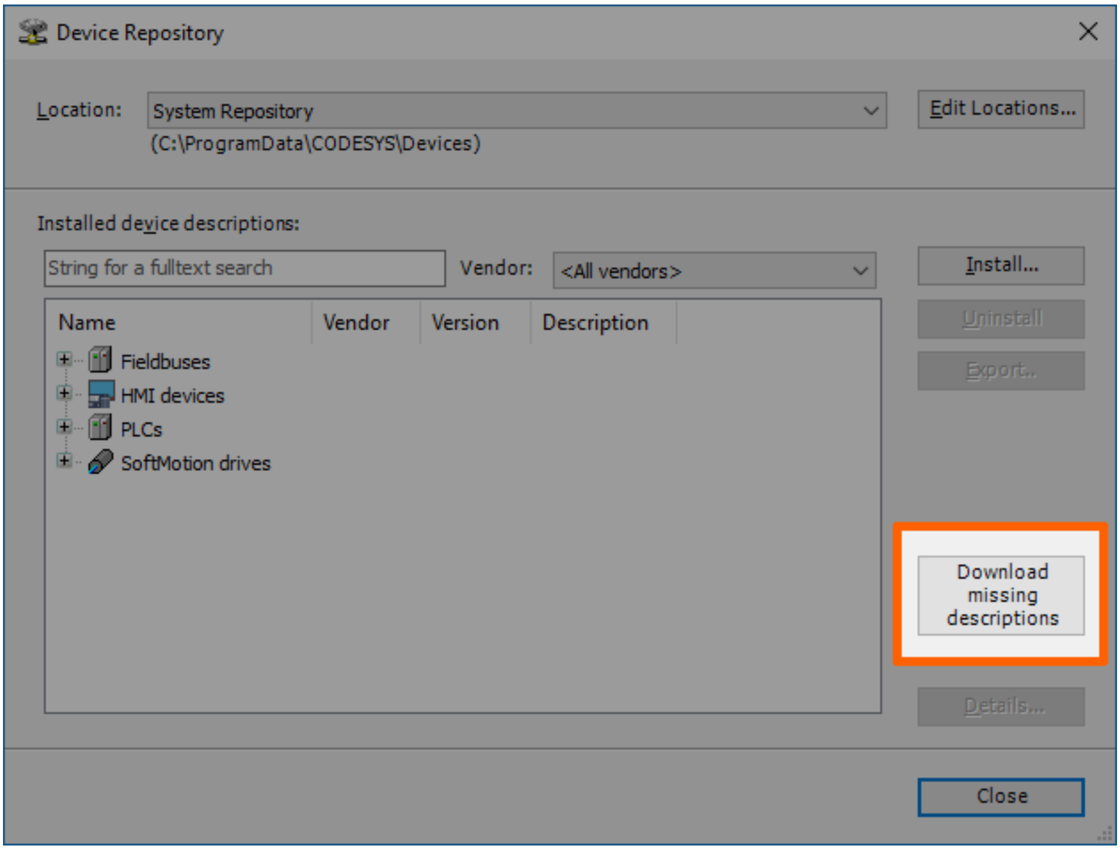
7. Repeat steps 3 to 5 until no more libraries appear in the download window.

Downloading missing device descriptions

Apart from the device description for the MFN 100 that have been installed manually before, device descriptions of generic devices may be missing for the CODESYS Development System to function as intended. The CODESYS Development System will identify the missing device descriptions automatically but this time it will not generate an error message unless a CODESYS application is being loaded into the MFN 100.

1. Click **Tools > Device Repository**.

2. Click **Download missing descriptions**.



NOTE

The button for downloading missing descriptions will not appear if no device descriptions of generic devices are missing. Close the window and continue if that is the case.

3. Click **Download** in the new window.

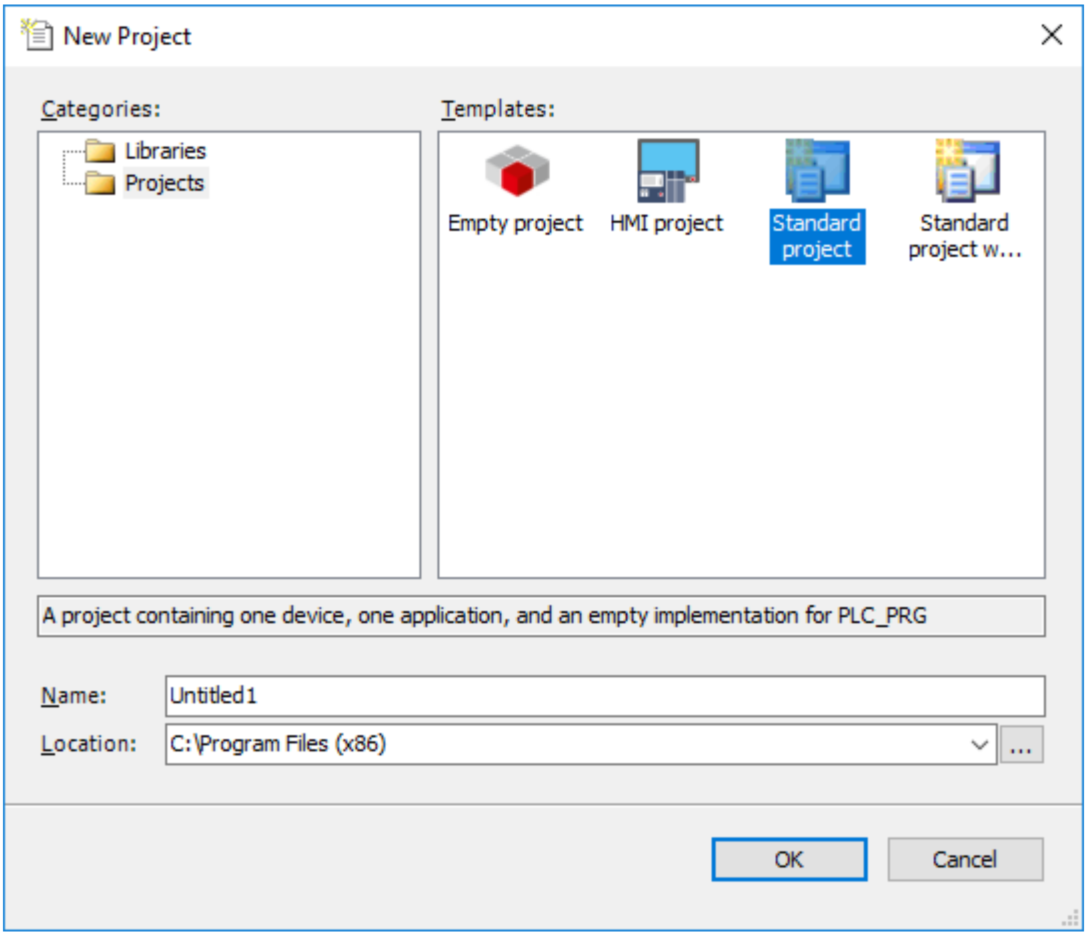
4. Click **Close** when the download is finished.

Creating a new CODESYS project

This example shows how to create a new project in the CODESYS Development System. The easiest way to get started is to create a **Standard project**.

1. Start CODESYS
2. Go to **File > New Project**.
3. Click **Standard project** on the right side among the templates.
4. Enter a name for the project.
5. Choose a **Location** where the project will be saved.

6. Click **OK** to save the project.



New Project

Categories:

- Libraries
- Projects

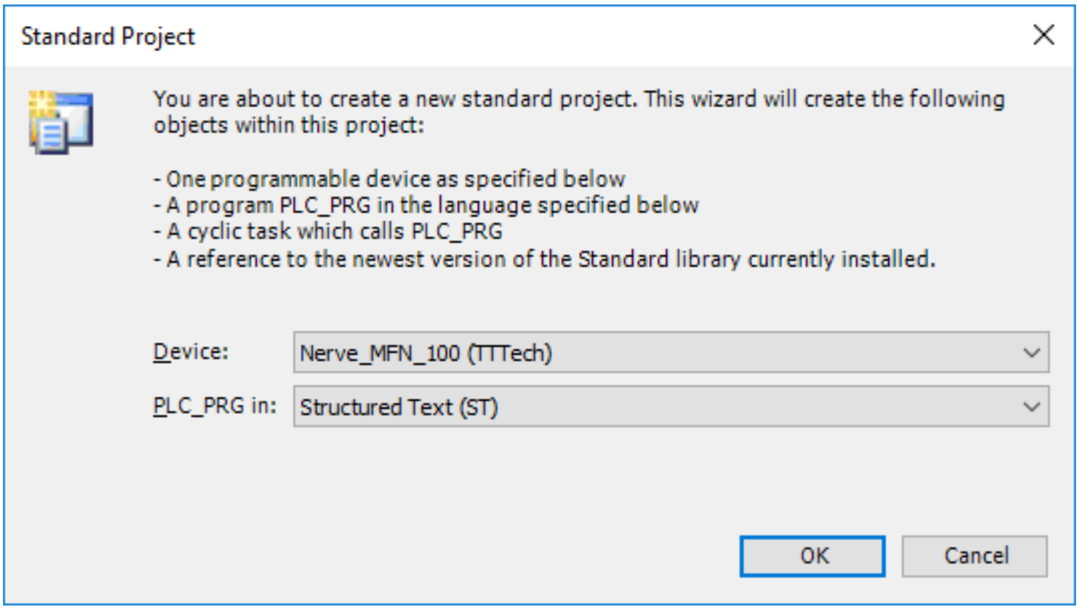
Templates:

A project containing one device, one application, and an empty implementation for PLC_PRG

Name: Untitled1

Location: C:\Program Files (x86)

7. Select **Nerve_MFN_100 (TTTech)** as the device.



Standard Project

You are about to create a new standard project. This wizard will create the following objects within this project:

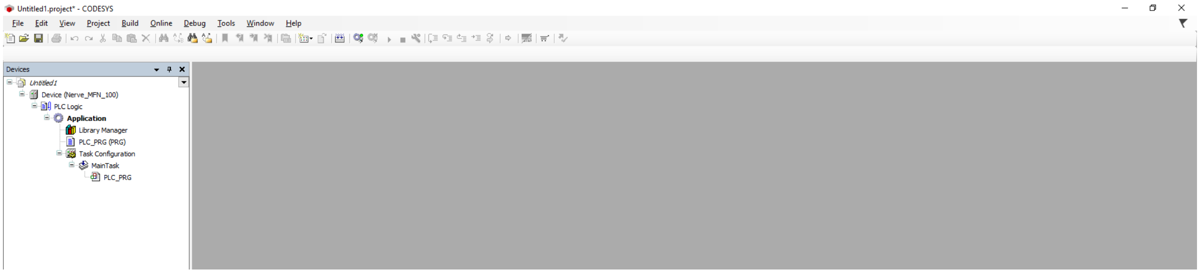
- One programmable device as specified below
- A program PLC_PRG in the language specified below
- A cyclic task which calls PLC_PRG
- A reference to the newest version of the Standard library currently installed.

Device: Nerve_MFN_100 (TTTech)

PLC_PRG in: Structured Text (ST)

8. Click **OK**.

The result is an empty project that is open in the main view of CODESYS.

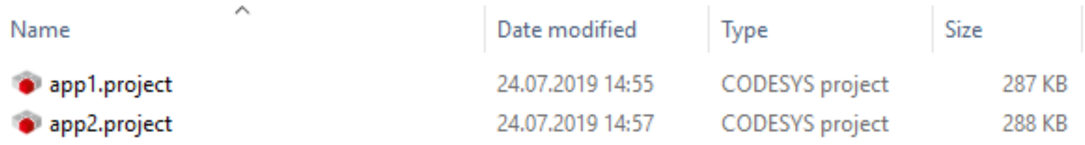


Working with the default applications

To work with existing applications first, modify the default applications `app1.project` and `app2.project`. They have been sent as part of the delivery.

1. Start CODESYS.
2. Go to **File > Open Project**.
3. Select the location where the default applications are saved.

4. Select the application to work with.



Name	Date modified	Type	Size
app1.project	24.07.2019 14:55	CODESYS project	287 KB
app2.project	24.07.2019 14:57	CODESYS project	288 KB

5. Click **Open**.

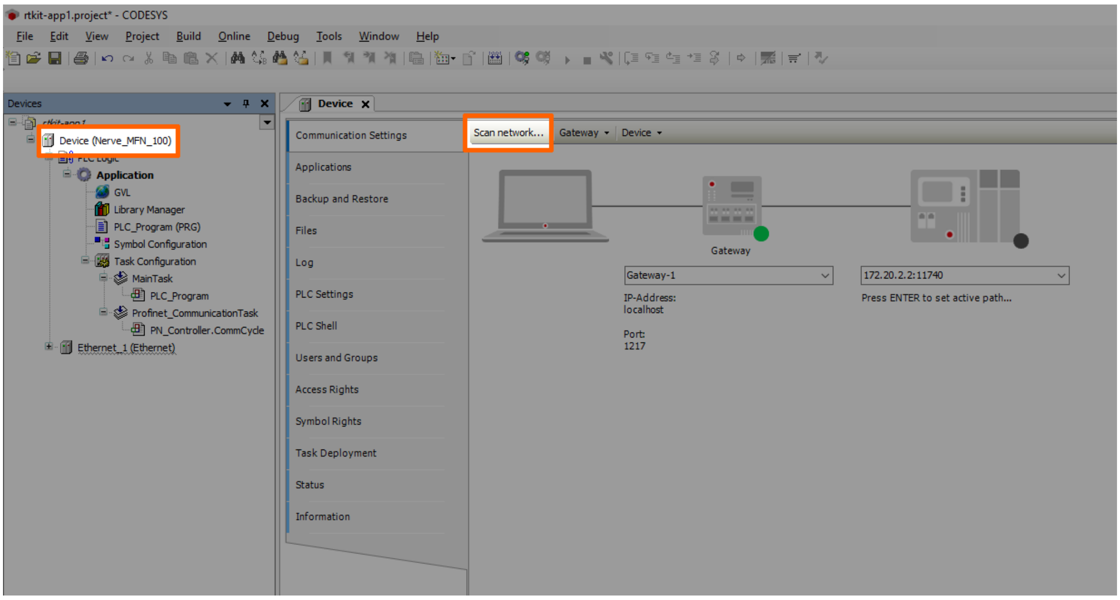
If the default applications are opened for the first time, some libraries and device descriptions will be missing. Follow the instructions above to see how to download the missing files.

Connecting to the MFN 100

Before downloading CODESYS applications to the MFN 100, make sure that the device description of the MFN 100 is installed in the CODESYS Development System.

1. Open or create a CODESYS project.
2. Double-click **Device (Nerve_MFN_100)** in the tree view on the left.

3. Go to **Communication Settings > Scan network...**.



Gateway-1

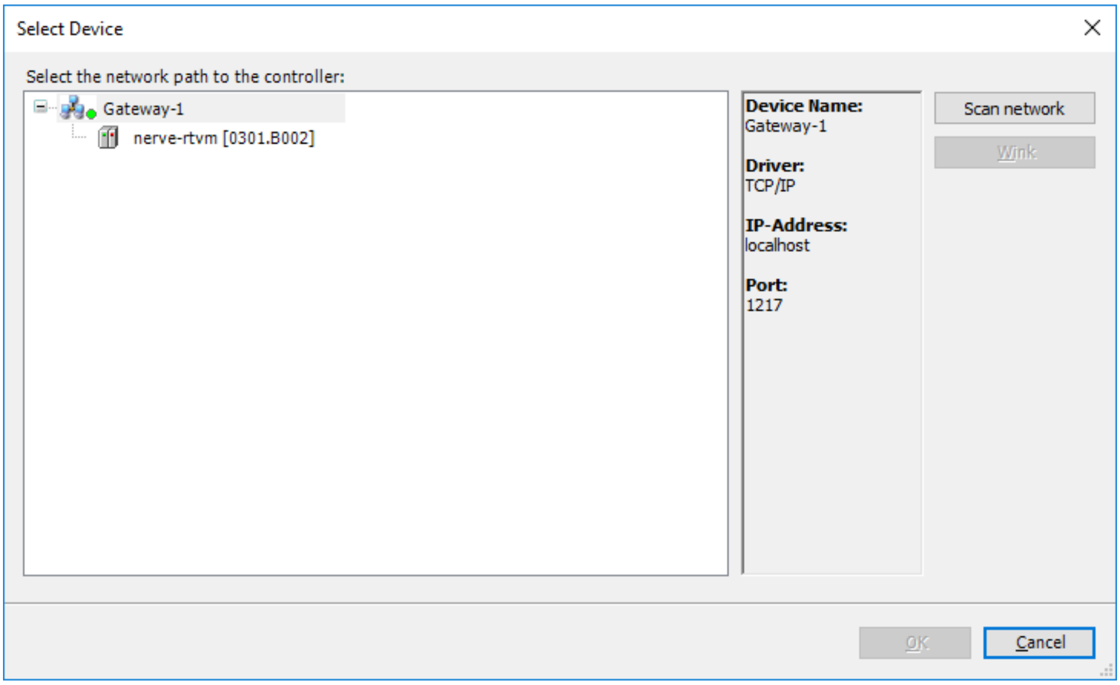
IP-Address:
localhost

Port:
1217

172.20.2.2:11740

Press ENTER to set active path...

4. Select the MFN 100 (here **nerve-rtvm [XXXX.XXXX]**) in this window.



Select the network path to the controller:

Device Name:
Gateway-1

Driver:
TCP/IP

IP-Address:
localhost

Port:
1217

NOTE

When more than one network is active on the workstation, it sometimes happens that the MFN 100 cannot be found. Continue reading if the MFN 100 does not appear in this window.

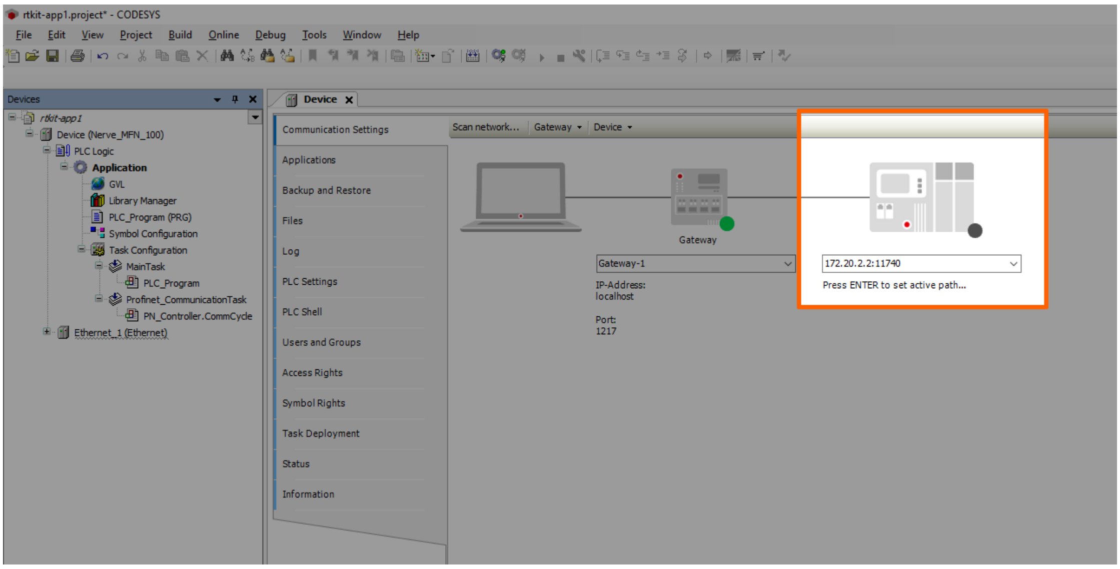
5. Click **OK**.

Typically the MFN 100 will be found automatically. If the MFN 100 cannot be found, enter the IP address and port of the CODESYS runtime manually.

1. Double-click **Device (Nerve_MFN_100)** in the tree view on the left.

2. Go to **Communication Settings** in the middle of the window.

3. Enter `172.20.2.2:11740` in the text box under the device on the right.



Gateway-1

IP-Address:
localhost

Port:
1217

172.20.2.2:11740

Press ENTER to set active path...

4. Press Enter.

The CODESYS Development System is now connected to the MFN 100 and applications can be downdloaded into the CODESYS runtime.

Downloading an application to the MFN 100

CODESYS applications can be loaded directly into the MFN 100. However, before downloading an application into the MFN 100 it needs to be free of errors.

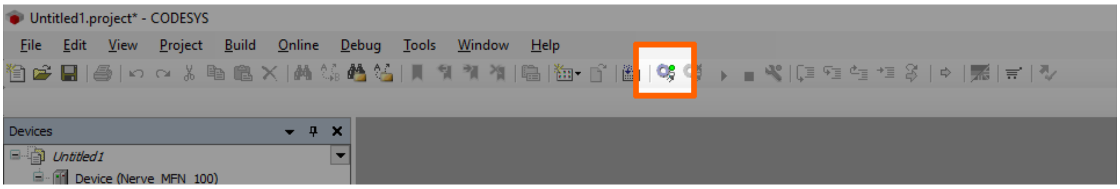
The process of downloading an application is slightly different if an entirely new application is downloaded into the MFN 100 or if an application is being updated that has already been downloaded into the MFN 100. If updating an application that has been downloaded to the MFN 100 before, continue with Downloading an Updated Application to the MFN 100 further down below.

Downloading a new application to the MFN 100

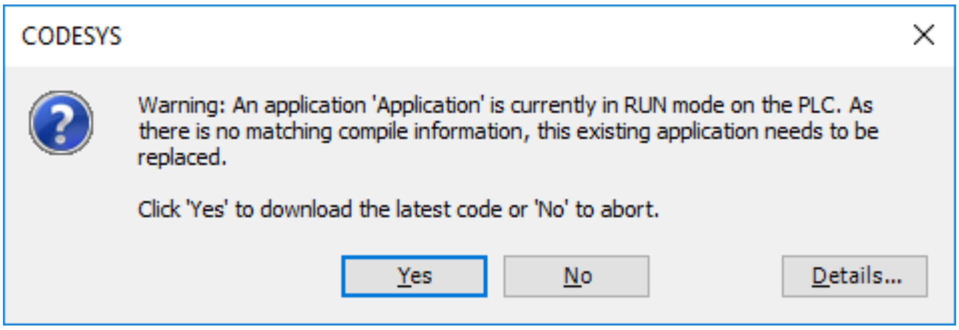
Once a project has been created and programming is finished, the CODESYS application can be downloaded to the MFN 100 directly.

1. Open the CODESYS project to load into the MFN 100.

2. Click the **Login** symbol in the CODESYS menu bar.


Untitled1.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
Untitled1
Device (Nerve_MFN_100)

3. Click **Yes** in the pop-up window.

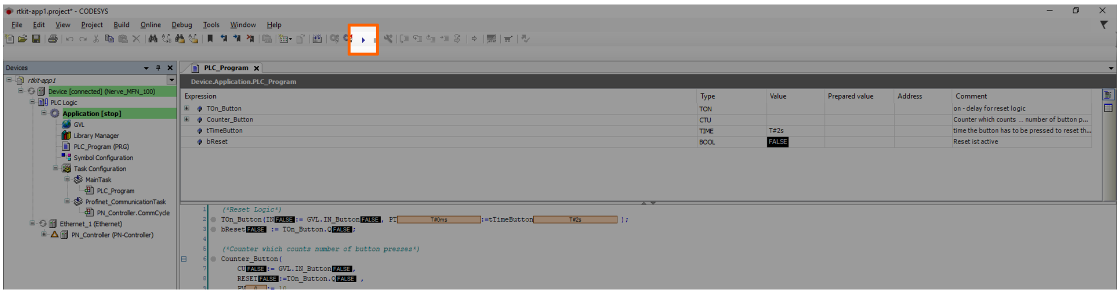


CODESYS

Warning: An application 'Application' is currently in RUN mode on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

4. The application is stopped now. Click the **Play** symbol in the CODESYS menu bar.



rtkit-app1.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
rtkit-app1
Device [connected] (Nerve_MFN_100)
PLC Logic
Application [stop]
GVL
Library Manager
PLC_Program (PRG)
Symbol Configuration
Task Configuration
MainTask
PLC_Program
Profinet_CommunicationTask
PN_Controller.CommCycle
Ethernet_1 (Ethernet)
PN_Controller (PN-Controller)
PLC_Program
Device.Application.PLC_Program

Expression	Type	Value	Prepared value	Address	Comment
TOn_Button	TON				on - delay for reset logic
Counter_Button	CTU				Counter which counts ... number of button p...
tTimeButton	TIME	T#2s			time the button has to be pressed to reset th...
bReset	BOOL	FALSE			Reset ist active

```
(*Reset Logic*)
TOn_Button(IN FALSE := GVL.IN_Button FALSE , PT  T#0ms  :=tTimeButton  T#2s  );
bReset FALSE  := TOn_Button.Q FALSE ;

(*Counter which counts number of button presses*)
Counter_Button(
    CU FALSE := GVL.IN_Button FALSE ,
    RESET FALSE :=TOn_Button.Q FALSE  ,
    PV  0  := 10
```

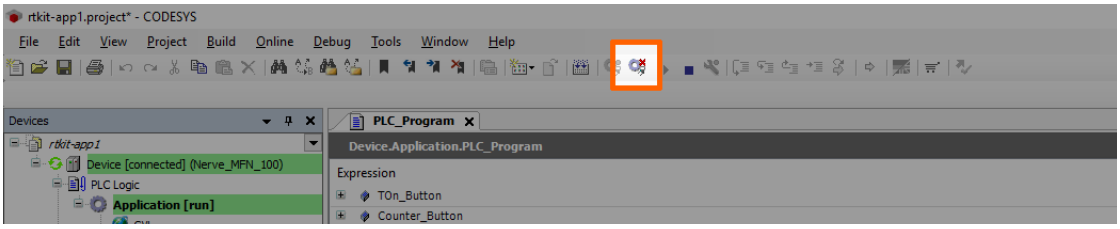
The application is now loaded to the MFN 100.

Downloading an updated application to the MFN 100

If updating an application after loading it into the MFN 100, it needs to be downloaded into the MFN 100 again. The download process is slightly different from downloading a new application into the MFN 100.

1. Stop the CODESYS application that has been loaded into the MFN 100 through the Local UI.

2. Click the **Logout** button in the CODESYS toolbar.

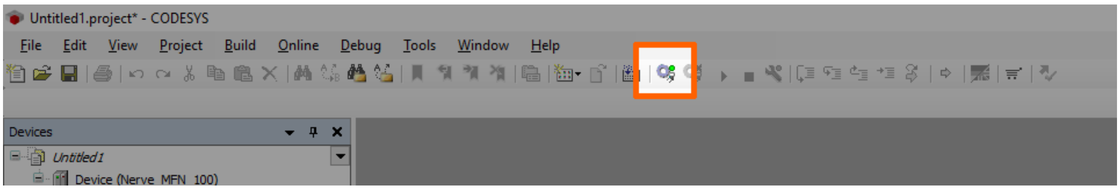


rtkit-app1.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
rtkit-app1
Device [connected] (Nerve_MFN_100)
PLC Logic
Application [run]
PLC_Program
Device.Application.PLC_Program
Expression
TOn_Button
Counter_Button

3. Expand **Device (Nerve_MFN_100) > PLC Logic > Application**.

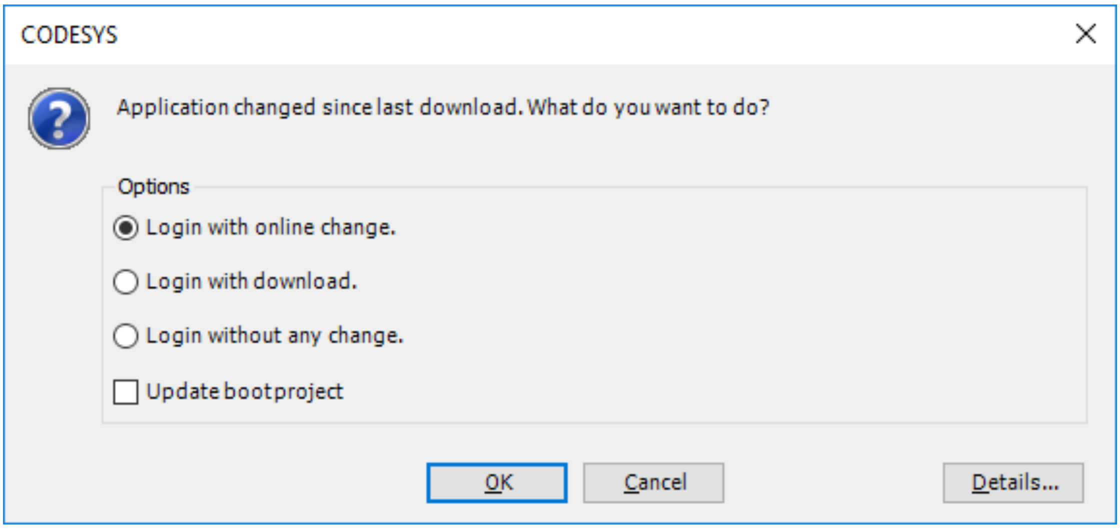
4. Double-click **PLC_Program (PRG)**.
5. Perform changes.

6. Click the **Login symbol** in the CODESYS menu bar.



Untitled1.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
Untitled1
Device (Nerve_MFN_100)

7. In the pop-up window, select one of the options.



CODESYS

Application changed since last download. What do you want to do?

Options

● Login with online change.

○ Login with download.

○ Login without any change.

□ Update bootproject

Item	Description
Login with online change.	The updated application will be loaded into the MFN 100. Variable values will not be reset. If the application was running before, it will be running after the download.
Login with download.	The updated application will be loaded into the MFN 100. Variable values will be reset. The application is stopped.
Login without any change.	The updated application will not be loaded into the MFN 100 but the code will keep the changes.

8. Click **OK**.

The application is now loaded to the MFN 100.

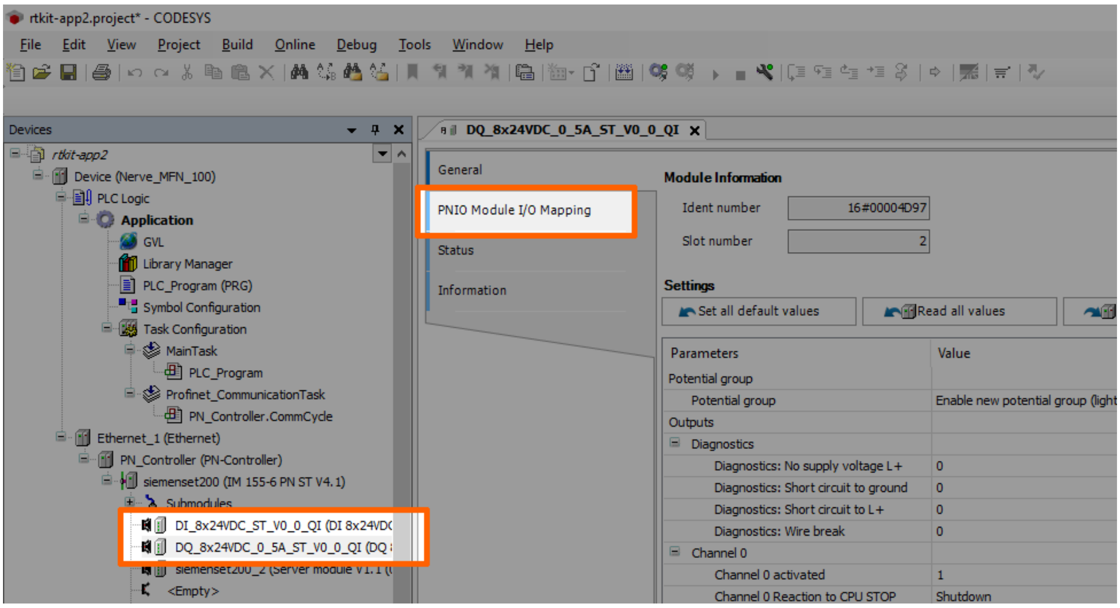
NOTE

For more help with programming PLC applications in the CODESYS Development System go to help.codesys.com.

Allocating variables to inputs or outputs

After connecting new sensors and actuators, assign variables to the I/O channel in CODESYS.

1. Open a CODESYS project.
2. Expand **Device (Nerve_MFN_100) > PLC Logic > Ethernet_1 > PN_Controller > siemenset200 (IM 155-6 PN ST V4.1)** in the tree structure on the left.
3. Double-click **DI_8x24VDC_ST_V0_0_QI (...)** for digital inputs.
   Double-click **DQ_8x24VDC_0_5A_ST_V0_0_QI (...)** for digital outputs.
4. Select **PNIO Module I/O Mapping**.


rtkit-app2.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
rtkit-app2
Device (Nerve_MFN_100)
PLC Logic
Application
GVL
Library Manager
PLC_Program (PRG)
Symbol Configuration
Task Configuration
MainTask
PLC_Program
Profinet_CommunicationTask
PN_Controller.CommCycle
Ethernet_1 (Ethernet)
PN_Controller (PN-Controller)
siemenset200 (IM 155-6 PN ST V4.1)
Submodules
DI_8x24VDC_ST_V0_0_QI (DI 8x24VDC
DQ_8x24VDC_0_5A_ST_V0_0_QI (DQ
siemenset200_2 (Server module V1.1 (
<Empty>
DQ_8x24VDC_0_5A_ST_V0_0_QI
General
PNIO Module I/O Mapping
Status
Information
Set all default values
Read all values

Module Information

Ident number: 16#00004D97

Slot number: 2

Settings

Parameters	Value
Potential group	
Potential group	Enable new potential group (light
Outputs	
Diagnostics	
Diagnostics: No supply voltage L+	0
Diagnostics: Short circuit to ground	0
Diagnostics: Short circuit to L+	0
Diagnostics: Wire break	0
Channel 0	
Channel 0 activated	1
Channel 0 Reaction to CPU STOP	Shutdown

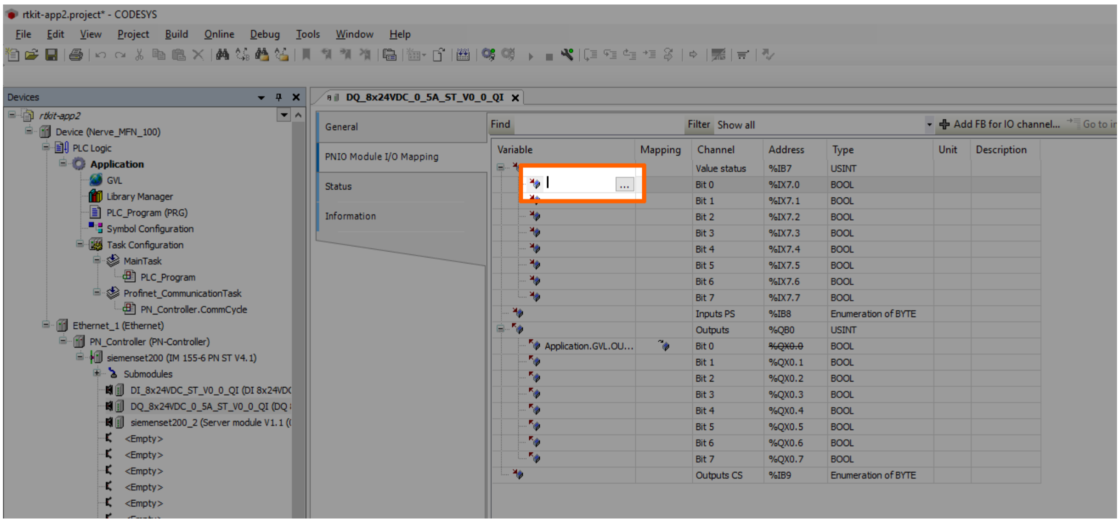
5. Fully expand the tree view.

6. Double-click the variable slot to assign.

NOTE

The inputs in this view do not match the physical inputs of the I/O module on the kit. The inputs here go from 0 to 7. The physical inputs go from 1 to 8. Therefore input 0 in this view represents the physical input 1 on the I/O module. This also applies to outputs.

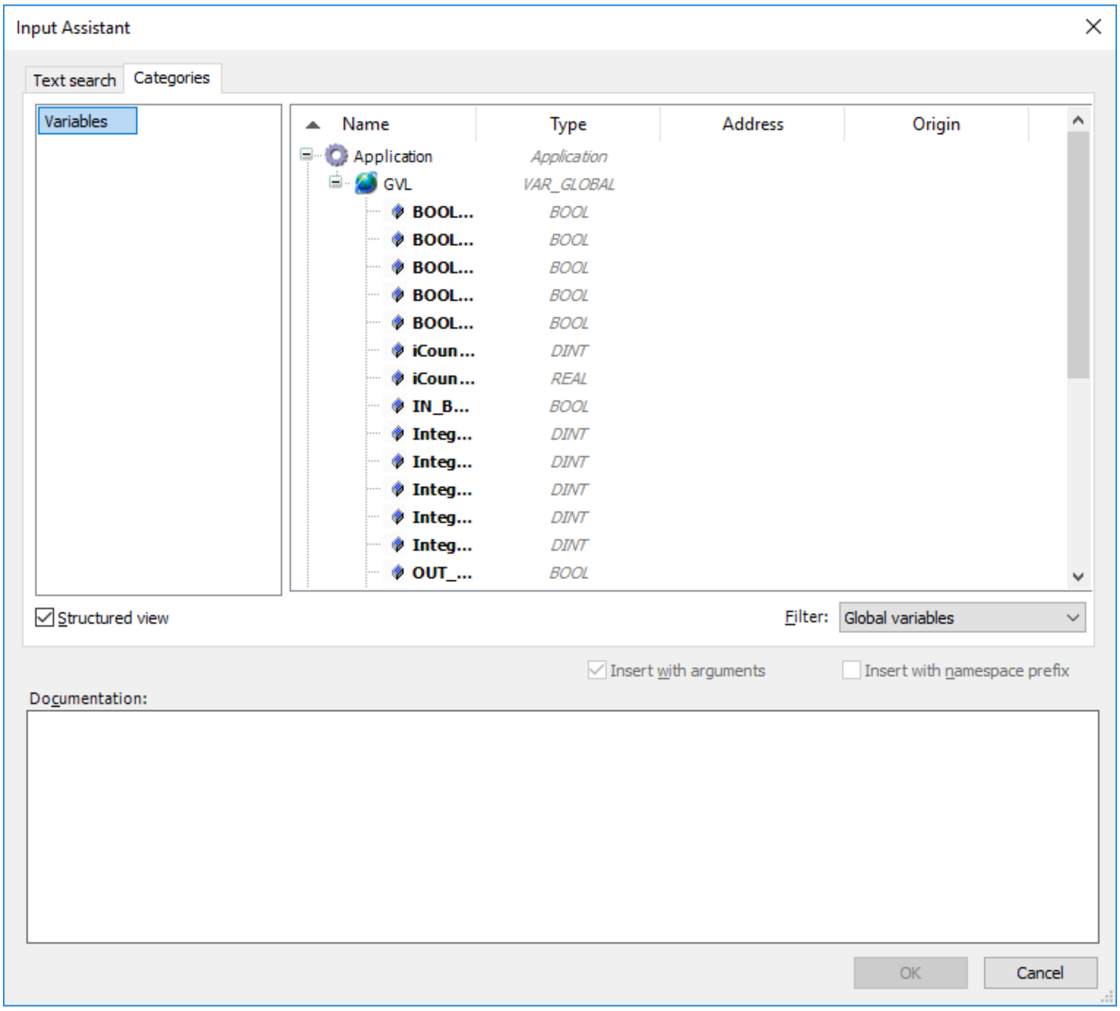
7. Click the three dots next to the variable slot.



Filter: Show all

Variable	Mapping	Channel	Address	Type	Unit	Description
		Value status	%IB7	USINT		
I …		Bit 0	%IX7.0	BOOL		
		Bit 1	%IX7.1	BOOL		
		Bit 2	%IX7.2	BOOL		
		Bit 3	%IX7.3	BOOL		
		Bit 4	%IX7.4	BOOL		
		Bit 5	%IX7.5	BOOL		
		Bit 6	%IX7.6	BOOL		
		Bit 7	%IX7.7	BOOL		
		Inputs PS	%IB8	Enumeration of BYTE		
		Outputs	%QB0	USINT		
Application.GVL.OU...		Bit 0	~~%QX0.0~~	BOOL		
		Bit 1	%QX0.1	BOOL		
		Bit 2	%QX0.2	BOOL		
		Bit 3	%QX0.3	BOOL		
		Bit 4	%QX0.4	BOOL		
		Bit 5	%QX0.5	BOOL		
		Bit 6	%QX0.6	BOOL		
		Bit 7	%QX0.7	BOOL		
		Outputs CS	%IB9	Enumeration of BYTE		

8. Expand **Application > GVL** in the new window.



Name	Type	Address	Origin
Application	Application		
GVL	VAR_GLOBAL		
BOOL...	BOOL		
BOOL...	BOOL		
BOOL...	BOOL		
BOOL...	BOOL		
BOOL...	BOOL		
iCoun...	DINT		
iCoun...	REAL		
IN_B...	BOOL		
Integ...	DINT		
Integ...	DINT		
Integ...	DINT		
Integ...	DINT		
Integ...	DINT		
OUT_...	BOOL		

☑ Structured view

Filter: Global variables

☑ Insert with arguments

☐ Insert with namespace prefix

Documentation:

9. Select the variable to assign.

> **NOTE**
>
> Make sure to select a variable of the same type as the input, i.e., a **BOOL** variable for a **BOOL** input or output.

10. Click **OK**.

Use the assigned variables to read data from connected sensors or to control actuator functionality.

> **NOTE**
>
> For more help with programming PLC applications in the CODESYS Development System go to help.codesys.com.

Enabling retain variables

`NerveCodesysRetainVar` is a library for storage and restoration of retain variables with prevention of data loss in case of system crash (e.g. power outage). To prevent data loss, the retain variables are stored in two file copies under `/opt/data/var/lib/nerve-codesys/PlcLogic/`.

The retain variables library (`NerveCodesysRetainVar.compiled-library`) and an example project showcasing typical use (`demo.project`) are available at the Nerve Software Center. Follow the instructions below to install and import the retain variables library into a project.

Installing and importing the NerveCodesysRetainVar library

Download the retain variables library (`NerveCodesysRetainVar.compiled-library`) from the Nerve Software Center first before following the instructions below.

1. Open or create a CODESYS project.
2. Select **Tools > Library Repository** in the toolbar.
3. Click **Install...** in the new window.
4. Navigate to the folder containing the retain variables library and select the `NerveCodesysRetainVar.compiled-library` file.
5. Select **OK**.

The library will appear in the **Application** element after it has been installed. Close the library repository window by selecting **Close** in the lower right. Next the library needs to import into the project.

1. Double-click **Library Manager** in the tree view on the left.
2. Select **Add library** in the **Library Manager** tab that opened on the right.
3. Enter `NERVE Retain` in the search field to search for the library.
4. Select the **NERVE Retain Variables Library**.
5. Select **OK**.

The **Nerve Retain Variables Library** now appears in the **Library Manager** tab.

Example project and use

The example project (`demo.project`) is meant to show how the library functions can be used. It includes two variants of a simple program counting up multiple counters stored as permanent variables:

- **PLC_PRG1** stores and restores the retain variables just once based on a trigger activated by boolean flags (`store` and `restore`).
- **PLC_PRG2** shows a typical use case, where the retain variables are restored once during the initialization phase and then periodically stored every cycle after all the counters are updated.

The NerveCodesysRetainVar library provides two functions:

- `StoreRetains` to store retain variables
- `RestoreRetains` to restore retain variables

The namespace of the library is `NerveCodesysRetainVar`.

Functions have a return value of type `RTS_IEC_RESULT` defined in the **SysTypes Interfaces** library. It is required to either include this library or define the return variables as type `NerveCodesysRetainVar.RTS_IEC_RESULT`. The possible error codes are defined by CmpErrors2 Interfaces.

Note that `RestoreRetains` may fail with error code 0x32 when the application runs for the first time, before any retain variables were stored. This is because the files with stored retain variables do not exist yet.

During the startup of the device, the CODESYS application uses the function `RestoreRetains`. This loads the last stored state of retain variables.

During execution, the CODESYS application periodically saves the current state of retain variables with the function `StoreRetains`. This could be done in a periodically executed task or at certain points during calculations, when it is meaningful to store the current state.
In case of a sudden crash of the CODESYS application (e.g. power failure), the last saved state of the retain variables will be restored.

> **NOTE**
>
> The type of memory must be taken into consideration for periodical storing of retain variables. Writing too often for longer periods of time may result in damage to disks.

Integrating Nerve Blue into the build system

The Management System can be controlled with the API. In this version of the Management System API documentation, the focus lies on working with workloads. As a demonstration, download the `Nerve_API_2.1.2.zip` from the Nerve Software Center. The login credentials for the download can be found in the customer profile.

With the Python script it is demonstrated how to:

- provision CODESYS, Docker and Virtual Machine workloads
- deploy workloads
- undeploy workloads
- delete workloads

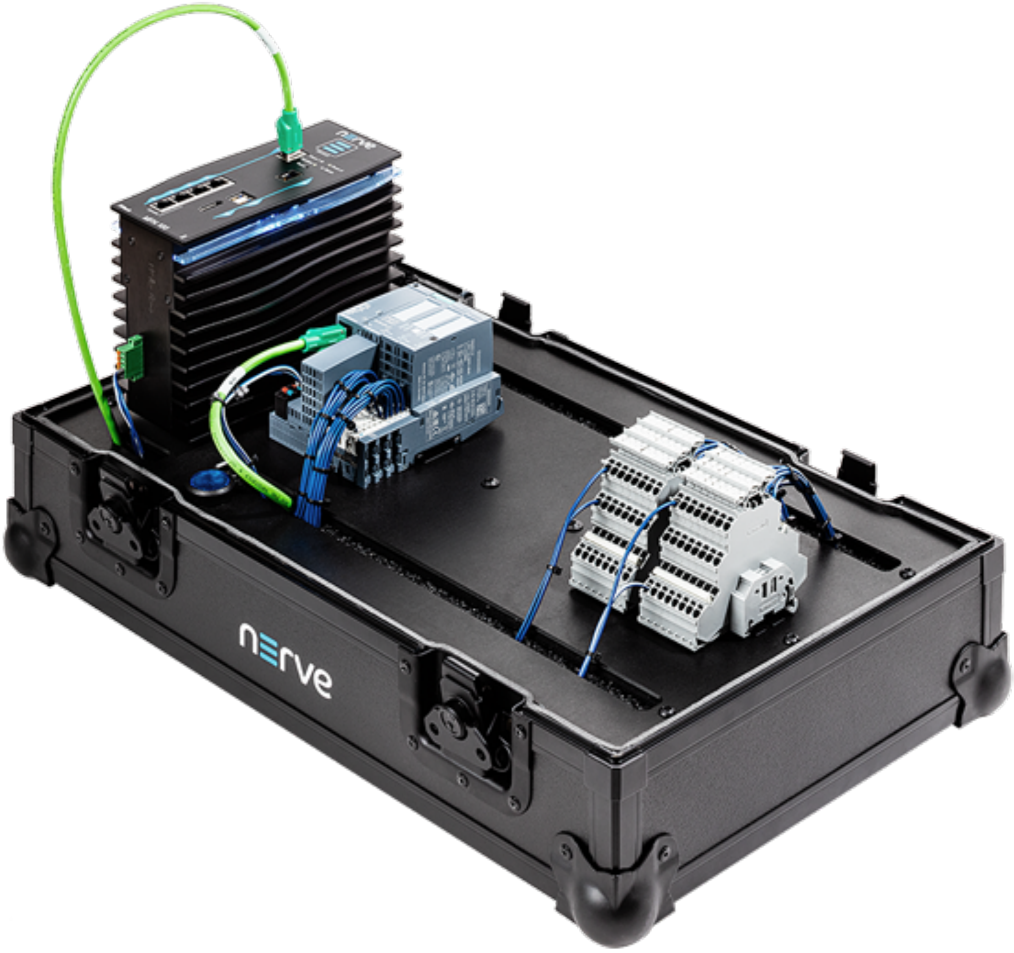
Full API documentation will be made available in future versions.

Release Notes

Nerve Blue

Nerve Blue Kit

Nerve Blue Kit



The Nerve Blue Kit is a ready-to-use hardware and software package, designed to support learning and training needs around digitalization. The kit provides everything needed to start collecting, storing and analyzing data, and can be expanded with custom sensors and IOs e.g. IO Link Master. Nerve Blue provides a virtualized software environment, enabling the implementation and remote management of multiple containers or virtual machines for specific use case. In addition, the kit can be used to connect to real-time data sources and update real-time control applications remotely via the Management System.

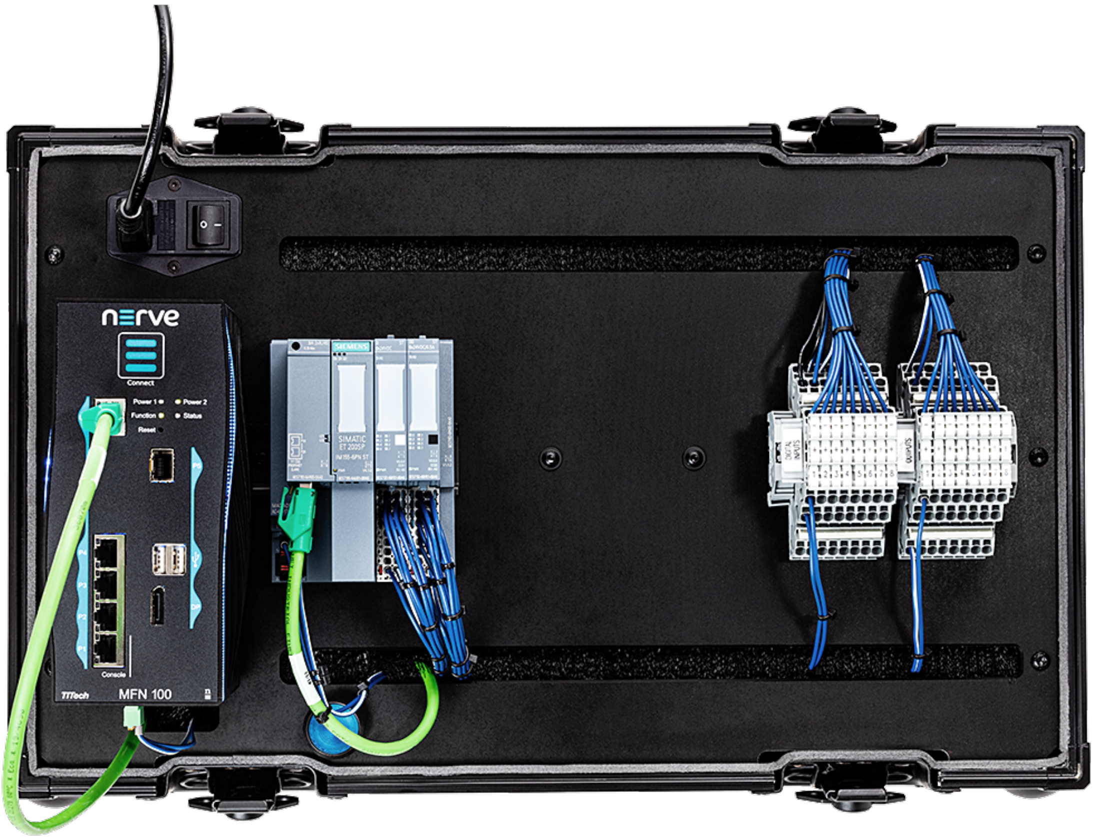
Key features

The Nerve Blue Kit is fully integrated in a robust, portable case suitable for lab use and demonstrations. Start with default control applications, then develop and deploy custom applications and extend the kit's functionality by adding sensors and actuators.

Plug and play	Extendable
• Fully configured Data Services for ingesting PROFINET, MQTT and OPC UA data • Two pre-installed CODESYS control applications for demo purposes • Pre-configured Grafana dashboard for visualization of data (available locally and in the cloud) • Workloads (CODESYS programs) available for deployment from the Management System to the Nerve Device	• Extra digital I/Os available for adding sensors to extend functionality • Data Services can be configured for new data streams including EtherCAT and Modbus TCP/IP • Customizable Grafana dashboard for visualization of new sensor data • Newly created workloads (virtual machines / containers / CODESYS programs) can be deployed from the Management System to the Nerve Device

Nerve Blue Kit contents

The contents of the kit are delivered in two separate boxes, a small box and a large box. The small box contains the MFN 100 including the Hardware Installation Guide and a mating connector. The large box has the remaining components of the kit: the mounting plate, the SIMATIC I/O module and the digital I/O terminal block.



A power cord and two network cables are also required to finish the setup.

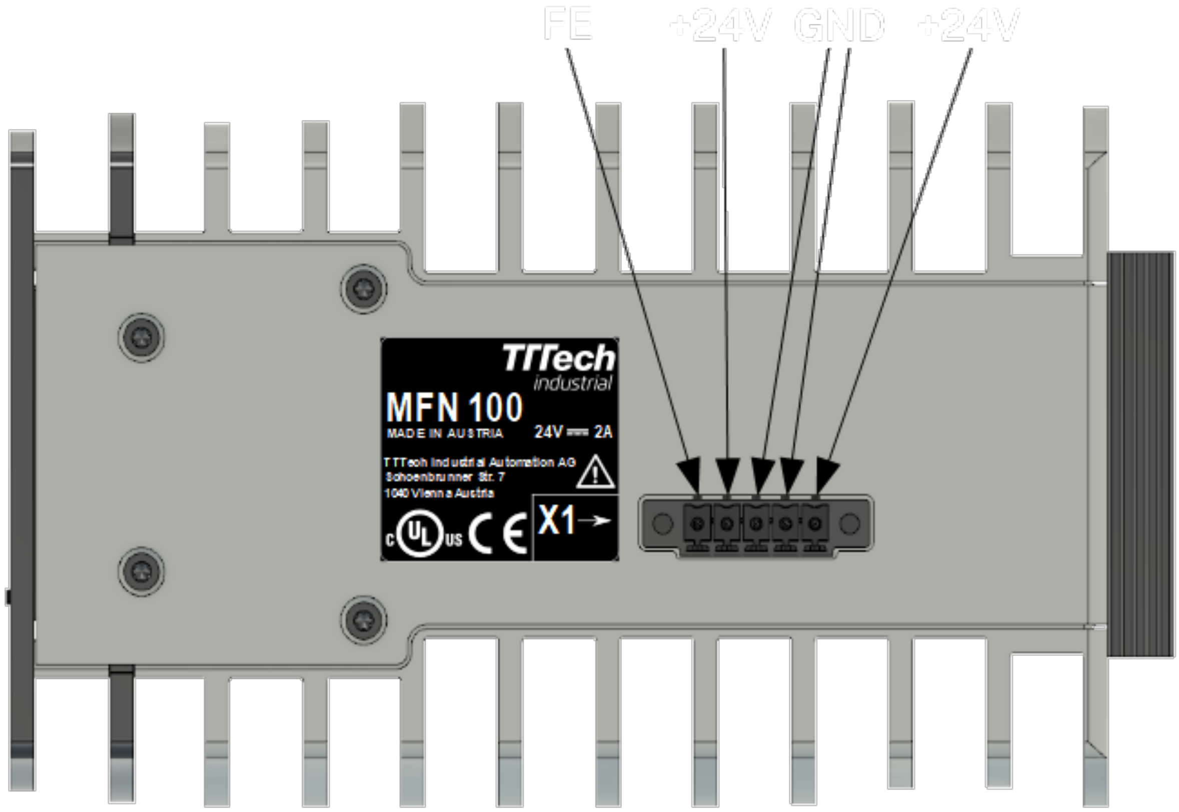
Content overview

Hardware	Software
• MFN 100-C64xx • SIMATIC ET200SP I/O system including: ◦ 1x SIMATIC ET 200SP bundle PROFINET IM, IM 155-6PN ST ◦ 1x SIMATIC ET 200SP 8x24V DC digital input ◦ 1x SIMATIC ET 200SP 8x24V DC digital output • Digital I/O terminal block • Illuminated push button • 24V DC Power supply • 1x Mating connector • 1x Network cable	• MFN 100-C64xx device description file • SIMATIC ET200SP I/O device description file • Two default CODESYS projects

The device descriptions and the default CODESYS applications can be downloaded from the Nerve Software Center. The login credentials for the Nerve Software Center can be found in the customer profile that has been sent as part of the delivery.

Power connectors overview

The power connectors are located at the bottom of the MFN 100 next to the label. There are two separate 24 V inputs, two GND inputs and one Functional Earth (FE) input. The inputs are fused internally. The fuse cannot be replaced by the user. The power supply inputs are protected against reverse polarity.



Pin	Description
1	Power supply line 2
2	GND
3	GND
4	Power supply line 1
5	Functional Earth (FE)

NOTE

The GND and FE pins (pins 2, 3, and 5) are electrically connected to the housing.

Setup and default operation

This section covers the hardware setup, explains the default operation of the kit and gives an introduction to the Management System.

Setting up the hardware

The MFN 100 is delivered in a separate box and has to be mounted on the kit. A mating connector is already connected to the power supply. Also, the SIMATIC ET 200SP I/O module is already connected to the I/O port of the MFN 100 with a network cable. In order to completely set up the kit, a power cord and two network cables are also required.

1. Mount the MFN 100 on the DIN rail on the left side of the kit.

   NOTE

   For help with mounting the MFN 100 on the DIN rail, refer to the Hardware Installation guide enclosed in the box of the MFN 100.

2. Plug the mating connector that is connected to the power supply into the bottom side of the MFN 100.
3. Connect port 2 of the MFN to a DHCP-enabled network with access to the Management System or internet access if the Management System is hosted by TTTech Industrial.

   NOTE

   Contact the IT administrator for help on how to allow external devices to connect to the network.

4. Connect the power cord to the power supply and to a power outlet. Make sure the power supply is turned off.
5. Push the button to switch on the power supply.

The MFN 100 will start after a few minutes and the blue light will turn on. All necessary services are initiated and after that, data is sent to the Management System.

Default operation

The Nerve Blue Kit comes with two default applications which can replicate scenarios found in industrial automation. The applications allow for data to be generated at the I/O, be acquired by the MFN 100 via Ethernet fieldbus, translated to OPC UA and be sent to the cloud for visualization. The data can be generated as a continuous flow throughout the cycle, as a regular data flow, or as irregular event based data flow. These different data flows mimic various types of industrial process and show the functionality of the Data Services from I/O to cloud.

The applications use the push button that is connected to the SIMATIC ET200 SP I/O module via the terminal block. Data is sent between the SIMATIC I/O module and the MFN 100 I/O port via PROFINET and the illumination of the button is controlled via the CODESYS soft PLC running on the MFN 100.

The two default CODESYS applications: **app1** and **app2** are used to alter flow of data from the I/O modules. Both apps control the illumination of the button and record two values: `iCountNumber` and `iCountButton`.

	app1	app2
iCountNumber (continuous data flow from the application to the Management System)	• Continuous count upwards from 0 to 1000. • Resets to 0 automatically when it has reached 1000.	• Continuous count downwards from 1000 to 0. • Resets to 1000 automatically when it has reached 0.
iCountButton (irregular and regular data flow from the application to the Management System)	• Counts the number of times that the button has been pressed (irregular data flow). • The push button lights up after 10 button presses. • The value resets to 0 when the button is held for two seconds. The light of the button goes out.	• Counts the duration that the button is held (regular data flow). • The counter continuously increases from 0. The push button lights up when the value reaches 25. • The value continuously decreases to 0 when the button is released.

In both application modes the calculated values are sent to the Management System and displayed in Grafana for visualization. The Management System allows users to deploy either of the default applications from the repository to the MFN 100. This demonstrates the ability to update CODESYS applications remotely and alter applications running on machines wherever they are in the world.

Beyond the two default applications, the kit can be used to develop custom CODESYS applications, which can be uploaded to the repository in the Management System and then deployed to the MFN 100. These applications can then be used in conjunction with corresponding actuators and/or sensors.

NOTE

The default application **app1** is loaded automatically during the initial startup of the MFN 100.

Connecting to the Management System

The Nerve Management System is a web-based service that permits management of Nerve Blue nodes that are registered.

NOTE

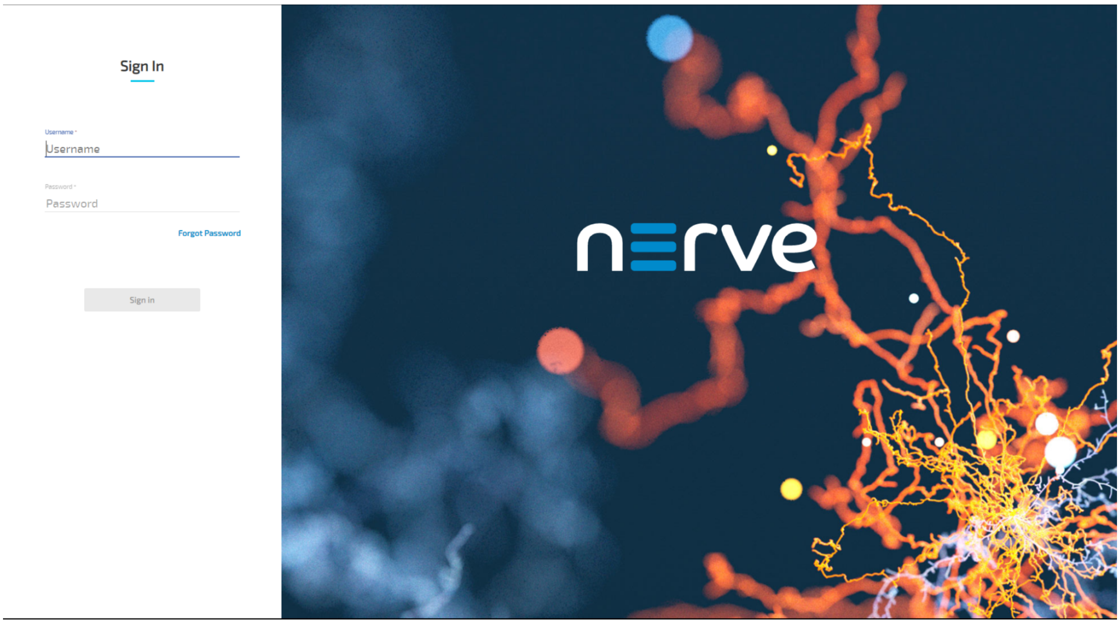
Google Chrome or Firefox Version 63 or later are recommended for the usage of the Management System.

Before connecting, make sure that the MFN 100 of the starter kit is connected to the network through port 2 and that an IP address has been assigned by the DHCP server. Contact the IT administrator for help with assigning an IP address.
The login credentials for the Management System are in the customer profile. The customer profile has been sent in form of a PDF as part of the delivery.

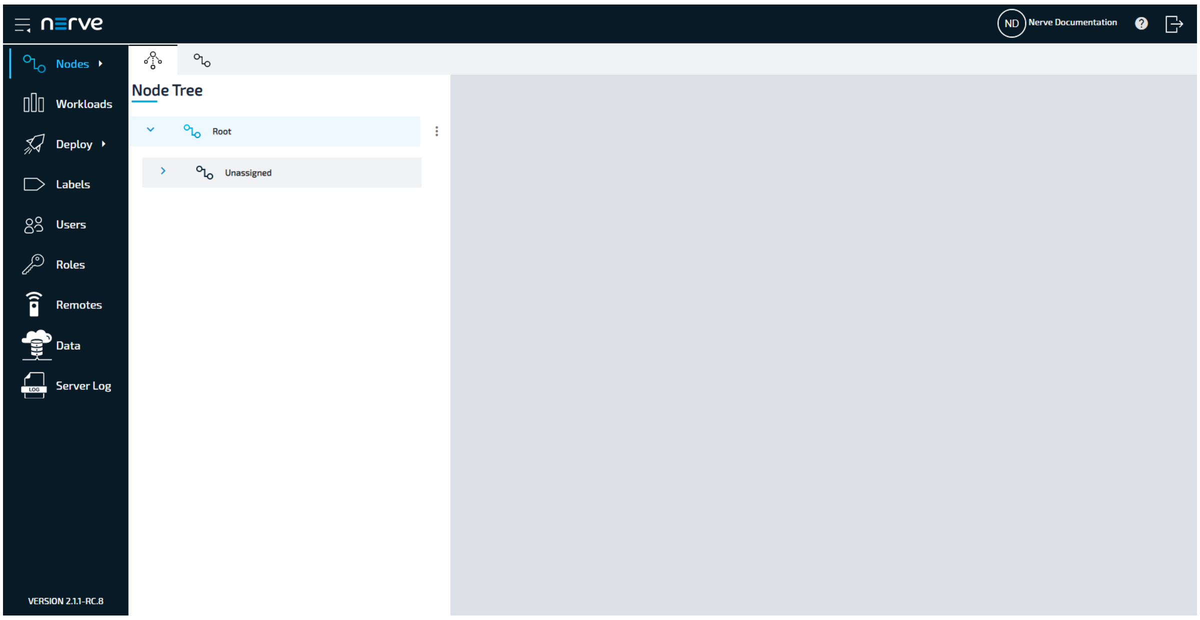
NOTE

If a customer profile has not been part of the delivery, contact a sales representative or TTTech Industrial customer support at support@tttech-industrial.com.

1. Go to the URL of the Management System in the customer profile.
2. Log in with the credentials provided in the customer profile.



The Management System will show the node tree by default.



One element in the node tree is already created at first login, containing the MFN 100 of the Nerve Blue Kit. All newly registered nodes will be located under **Root > Unassigned** by default.

Manage nodes, provision workloads and deploy workloads among other options from here. Refer to the user guide for more information on the Management System.

NOTE

Port 443 (HTTPS) and port 8883 (MQTTS) of the corporate firewall have to be open for communication between nodes and the Management System.

Moving a node from one tree element to another

Moving nodes in the node tree is possible by drag and drop. Make sure to create a new tree element before attempting to move a node.

1. Select **Nodes** in the navigation on the left.
2. Select the node tree tab

   on the right.
3. Expand the tree element of the node that will be moved. The default element is **Root > Unassigned**.
4. Choose the node to move.
5. Drag and drop the node to the newly created element. Elements expand automatically when they are hovered over.

6. Select **APPLY CHANGES (n)** in the upper-right corner of the node tree.

NOTE

`(n)` is a placeholder for the number of changes made to the node tree. For three performed changes, `(3)` will be displayed in the button above the node tree.

The node has now been moved to the new element. Note that a node cannot be moved back to **Unassigned** once it has been moved to another element.

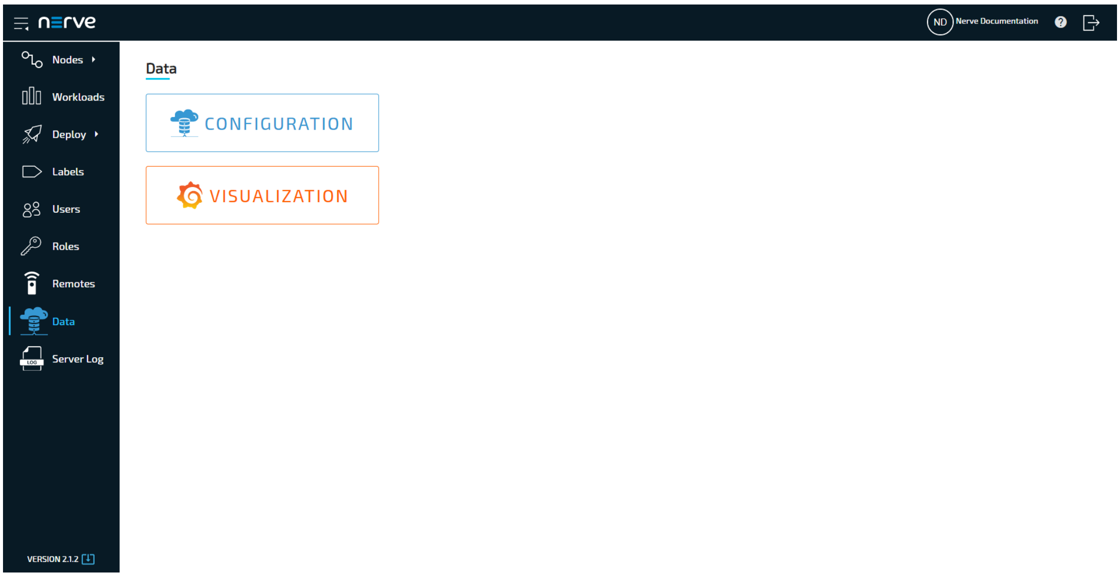
Viewing data in the Management System

Data that is being sent to the Management System can be viewed through the Grafana element in the Data Services instance of the Management System. The default values `iCountNumber` and `iCountButton` are set up by default. The URL of the Management System and the login credentials can be found in the customer profile. It has been sent in form of a PDF as part of the delivery.

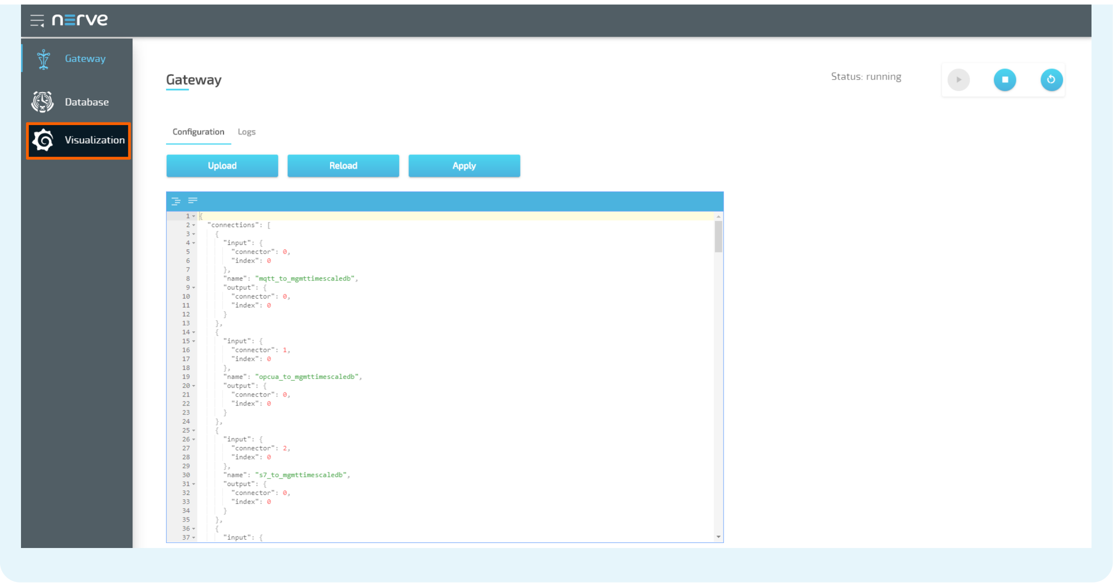
1. Log in to the Management System.
2. Select **Data** in the navigation on the left.

> **NOTE**
>
> If the menu item **Data** is not available, make sure the logged in user has the permission to access the Data Services. Refer to Assigning a role to a user for more information.

3. Select **VISUALIZATION**.

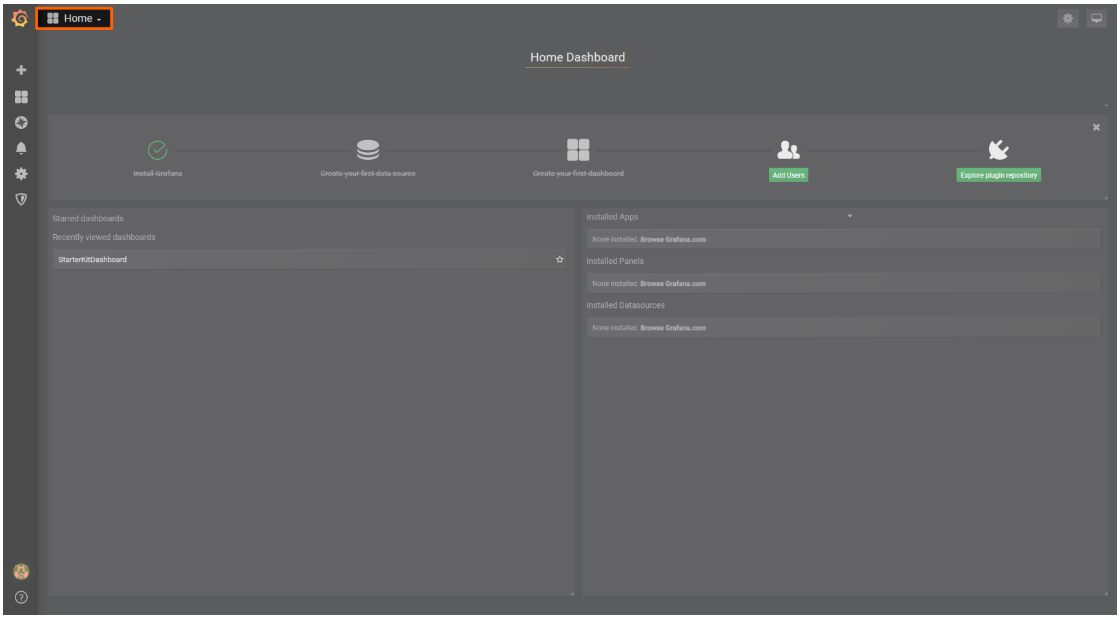


> **NOTE**
>
> The visualization element can also be reached from the Data Services UI. When in the Data Services UI, select **Visualization** in the navigation on the left and select **Open** to reach the Grafana UI.

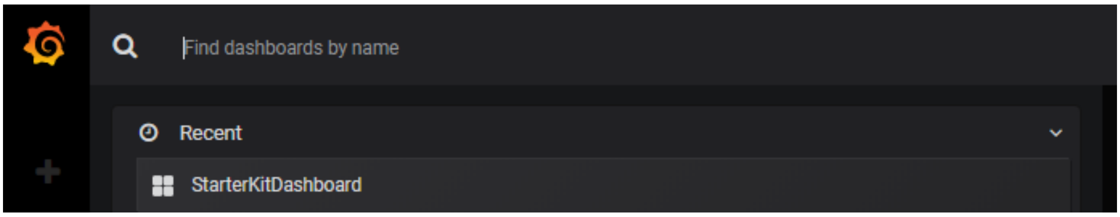


```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

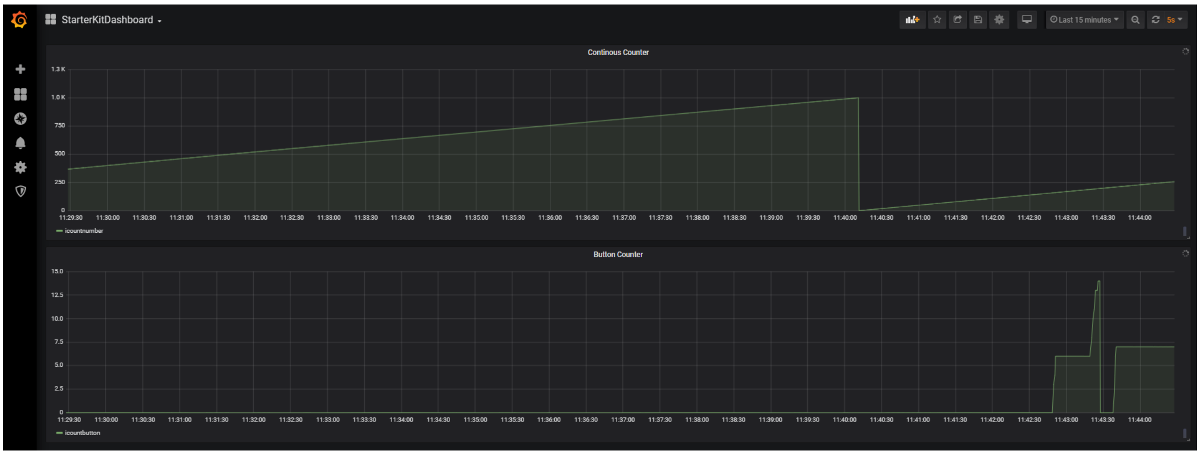
4. Select **Home** in the upper-left corner.


Home
Home Dashboard
~~Install Grafana~~
~~Create your first data source~~
~~Create your first dashboard~~
Add Users
Explore plugin repository
Starred dashboards
Recently viewed dashboards
StarterKitDashboard
Installed Apps
None installed. Browse Grafana.com
Installed Panels
None installed. Browse Grafana.com
Installed Datasources
None installed. Browse Grafana.com

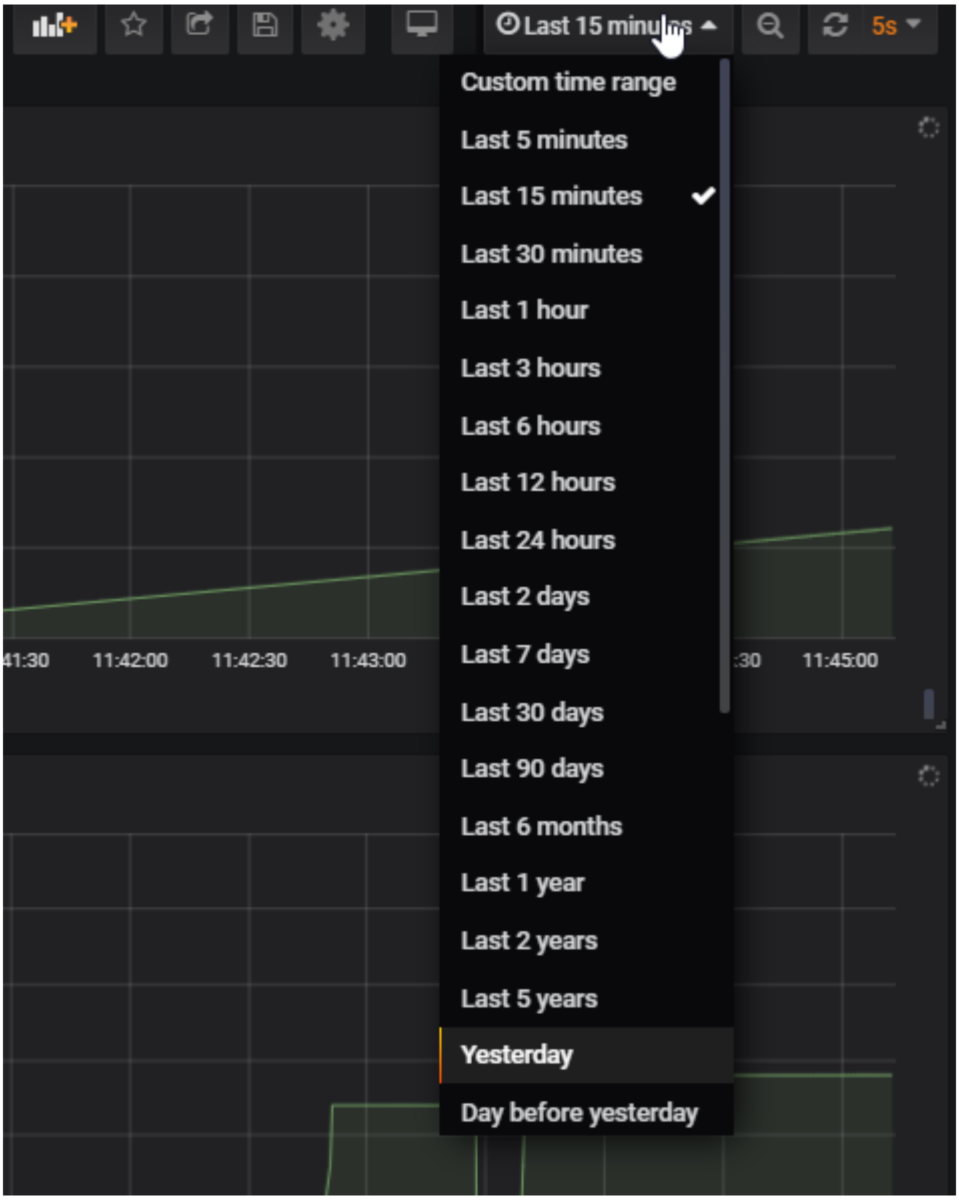
5. Select **Nerve Blue Kit** underneath the search bar.



This is the visualization of live data from the MFN 100. The calculated values, `iCountNumber` and `iCountButton`, that are generated in the default CODESYS applications **app1** and **app2** are displayed here by default.



The data automatically updates every 5 seconds. To change the update rate and the visible time range, access the settings by clicking the clock symbol in the upper-right.



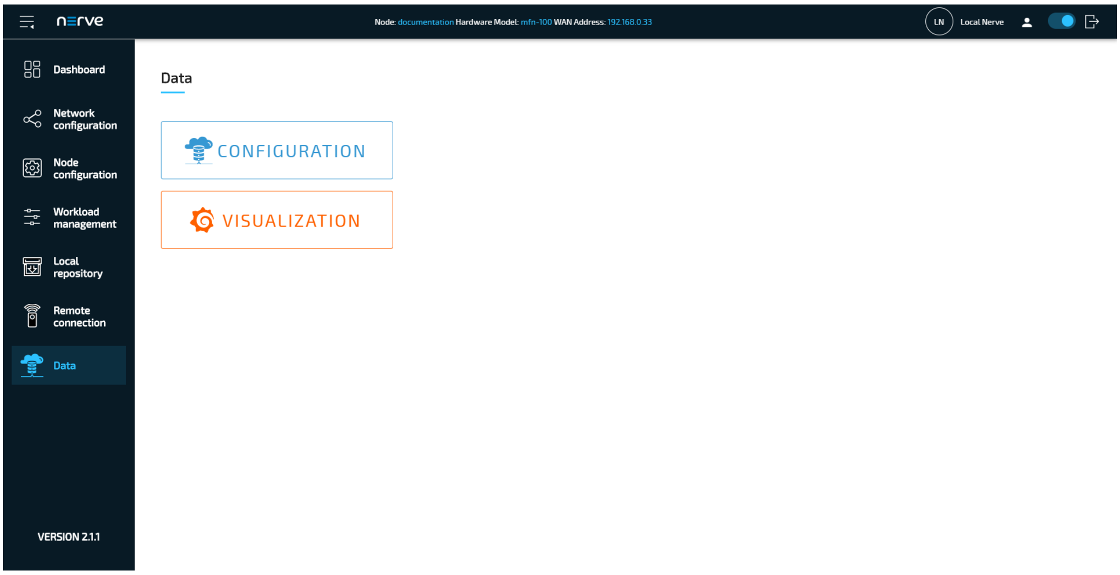
Viewing data on the Nerve Device

Data can also be viewed locally on the Nerve Device. The default values `iCountNumber` and `iCountButton` are set up by default.

1. Connect the workstation to the console port **P1** of the MFN 100.
2. Configure the network adapter through which the workstation is connected to the MFN 100 the following way:

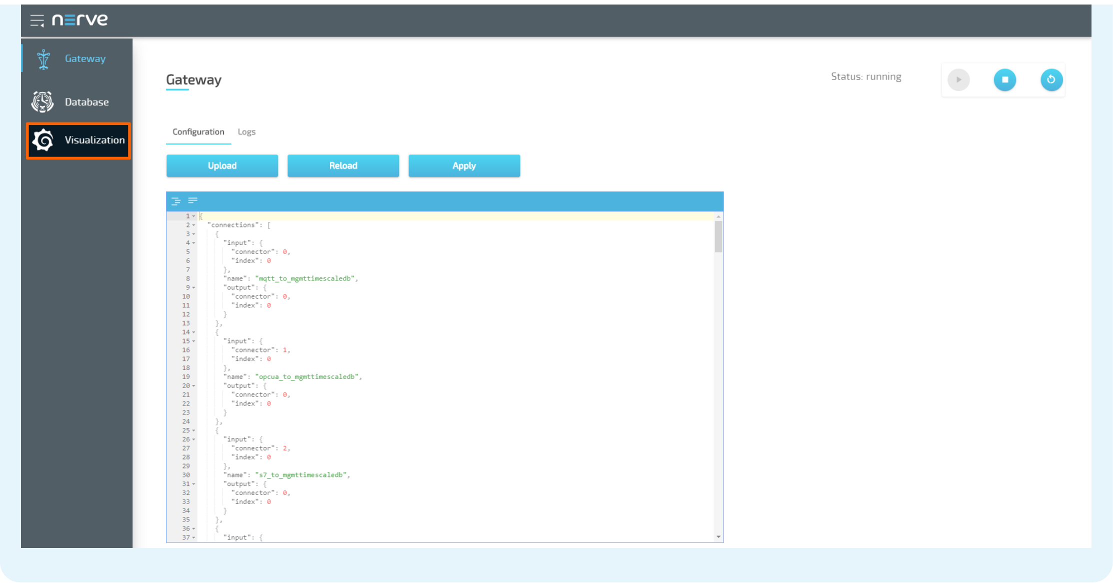
IP address	`172.20.2.90`
Subnet mask	`255.255.255.0`

3. Follow this link to reach the Local UI: http://172.20.2.1:3333.
4. Log in with the credentials from the customer profile.
5. Select **Data** in the navigation on the left.
6. Select **VISUALIZATION**.


Node: documentation Hardware Model: mfn-100 WAN Address: 192.168.0.33

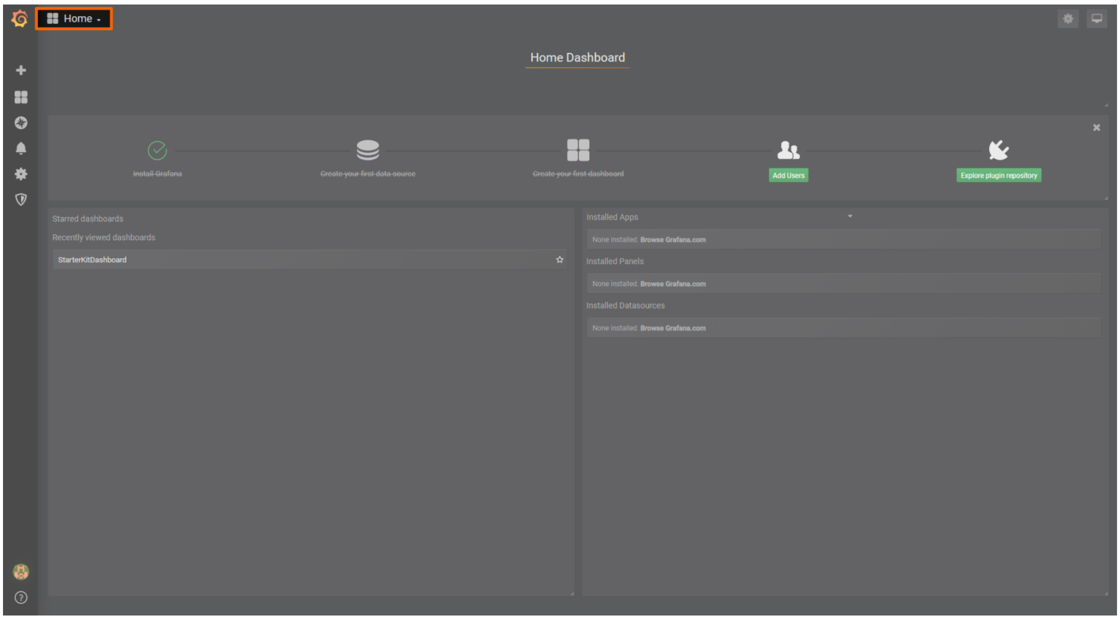
NOTE

The visualization element can also be reached from the Data Services UI. When in the Data Services UI, select **Visualization** in the navigation on the left and select **Open** to reach the Grafana UI.

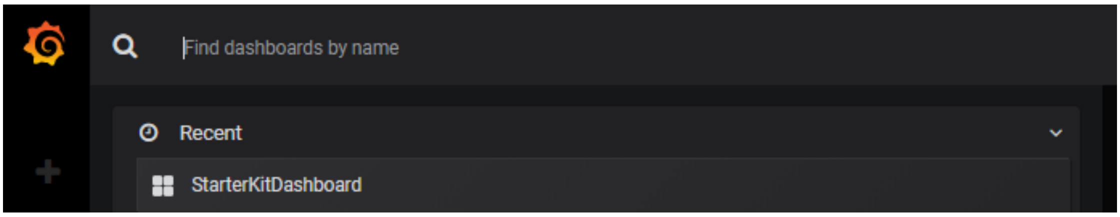


```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

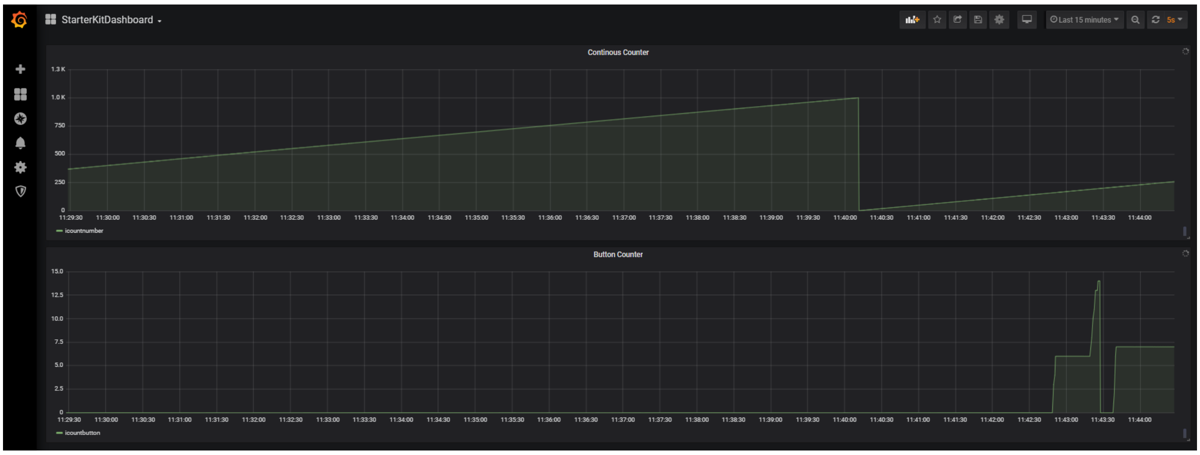
7. Select **Home** in the upper-left corner.


Home
Home Dashboard
~~Install Grafana~~
~~Create your first data source~~
~~Create your first dashboard~~
Add Users
Explore plugin repository
Starred dashboards
Recently viewed dashboards
StarterKitDashboard
Installed Apps
None installed. Browse Grafana.com
Installed Panels
None installed. Browse Grafana.com
Installed Datasources
None installed. Browse Grafana.com

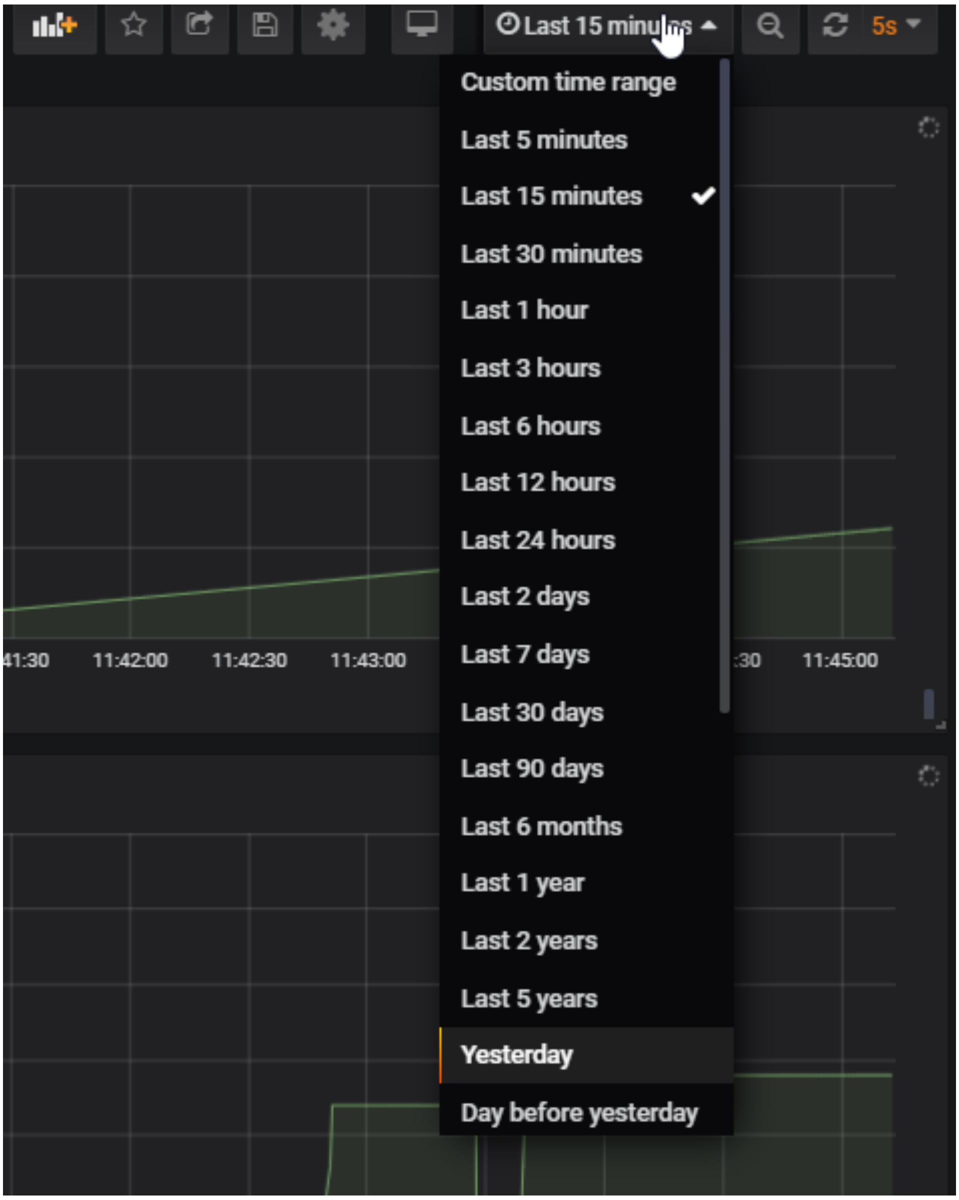
8. Select **Nerve Blue Kit** underneath the search bar.



This is the visualization of live data from the MFN 100. The calculated values, `iCountNumber` and `iCountButton`, that are generated in the default CODESYS applications **app1** and **app2** are displayed here by default.



The data automatically updates every 5 seconds. To change the update rate and the visible time range, access the settings by clicking the clock symbol in the upper-right.



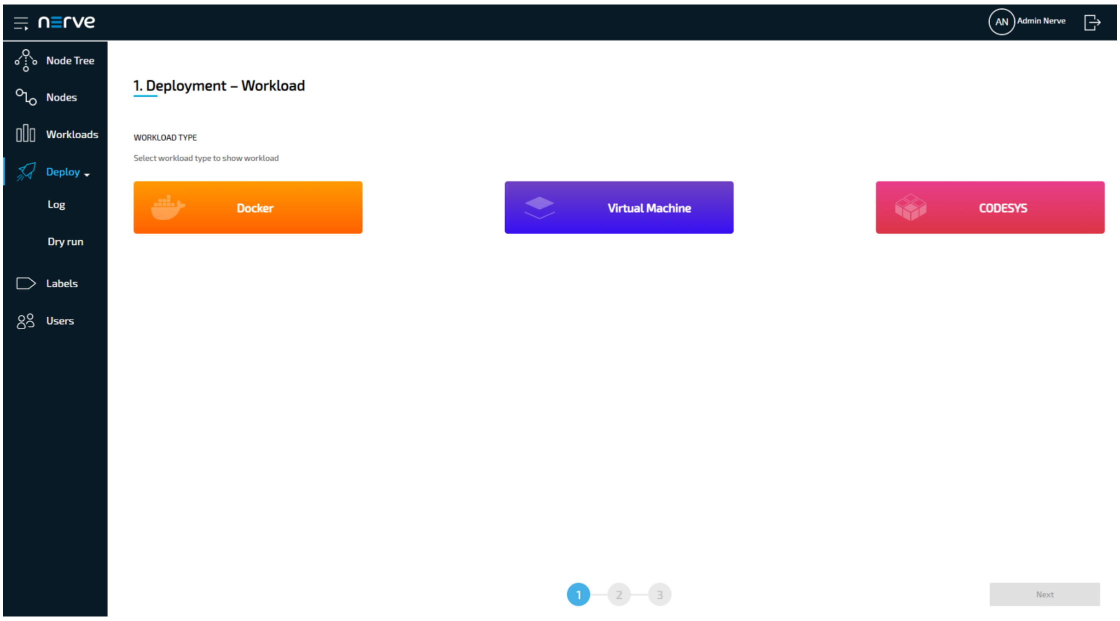
Downloading & uploading CODESYS applications from the Management System

With the kit CODESYS applications can be deployed from the Management System to the MFN 100. The two default CODESYS applications are already available in the Management System. Own applications can also be uploaded to the repository in the Management System and made ready for deployment.

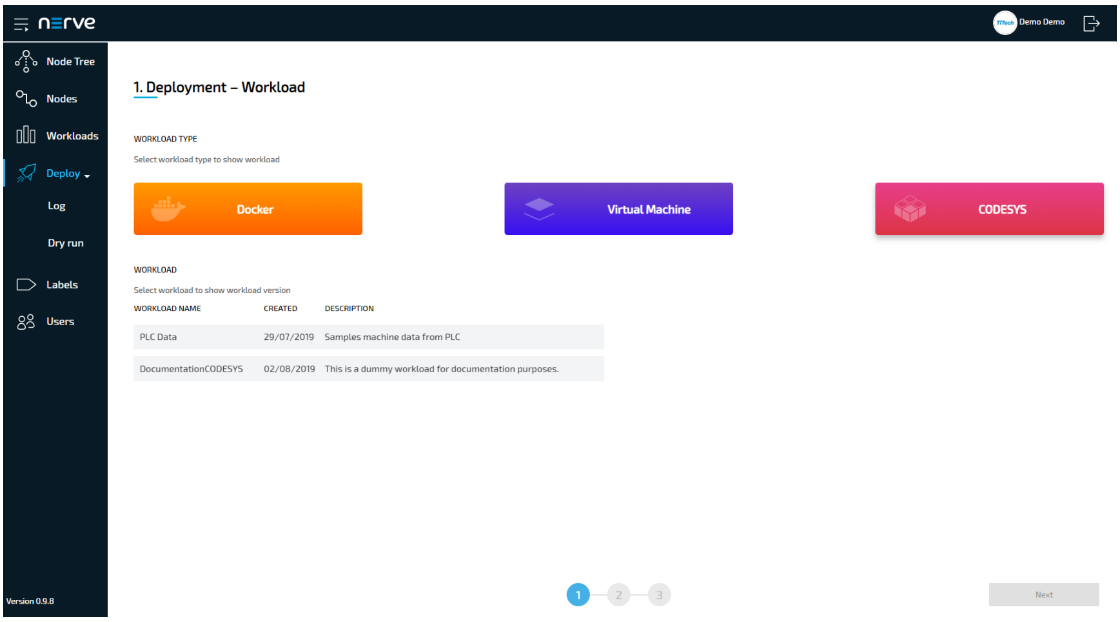
Deploying a CODESYS workload

One CODESYS workload is available with first login that can be used with the kit immediately. The workload is named **Nerve Blue Kit** and it has two versions: **Nerve Blue Starter Kit App1** and **Nerve Blue Starter Kit App2**. For more information on the deployment process for all workload types refer to the user guide.

1. Select **Deploy** in the left-hand menu.



2. Select the CODESYS workload icon on the right. A list of CODESYS workloads will appear below.



1. Deployment – Workload

WORKLOAD TYPE

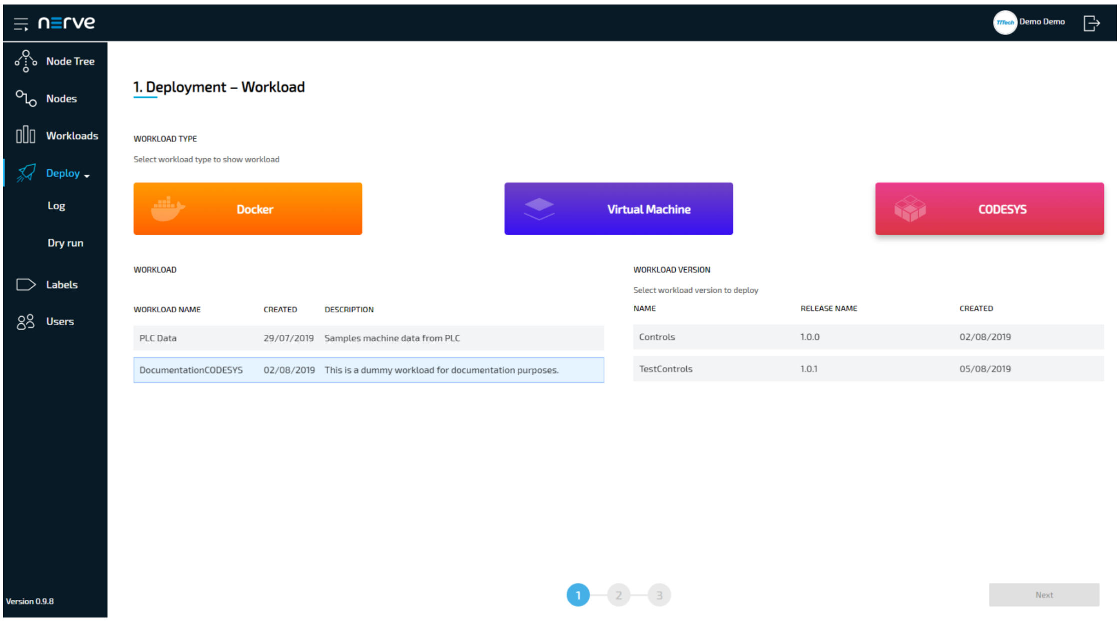
Select workload type to show workload

WORKLOAD

Select workload to show workload version

WORKLOAD NAME	CREATED	DESCRIPTION
PLC Data	29/07/2019	Samples machine data from PLC
DocumentationCODESYS	02/08/2019	This is a dummy workload for documentation purposes.

3. Select a workload from the list. A list of versions of this workload will appear to the right.



1. Deployment – Workload

WORKLOAD TYPE

Select workload type to show workload

WORKLOAD

WORKLOAD NAME	CREATED	DESCRIPTION
PLC Data	29/07/2019	Samples machine data from PLC
DocumentationCODESYS	02/08/2019	This is a dummy workload for documentation purposes.

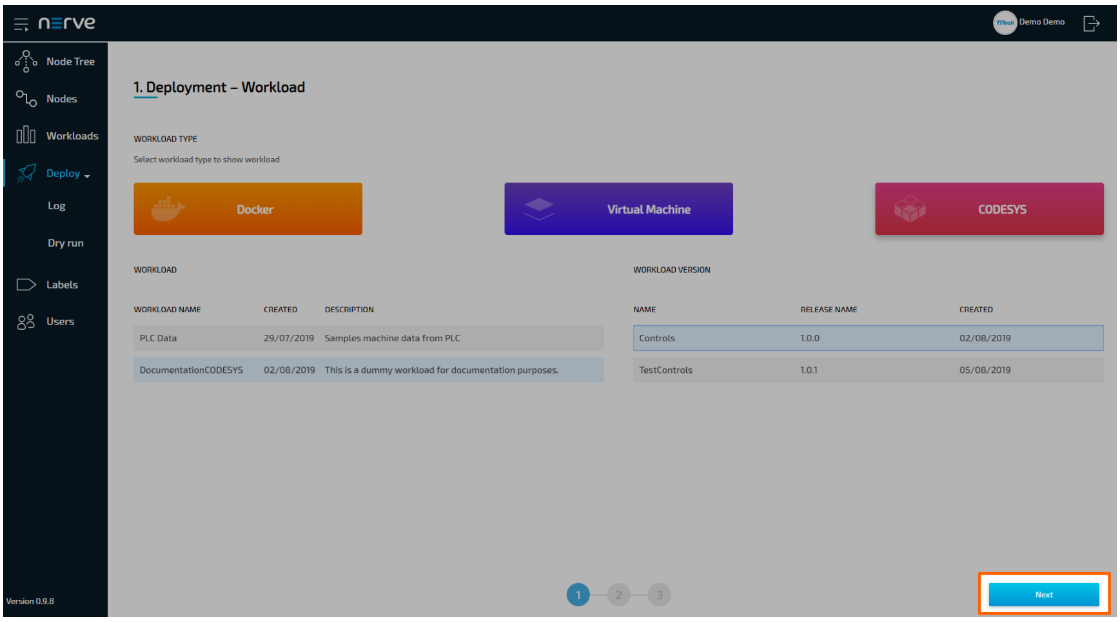
WORKLOAD VERSION

Select workload version to deploy

NAME	RELEASE NAME	CREATED
Controls	1.0.0	02/08/2019
TestControls	1.0.1	05/08/2019

4. Select the version of the workload.

5. Click **Next** in the bottom-right corner.



1. Deployment – Workload

WORKLOAD TYPE

Select workload type to show workload

WORKLOAD

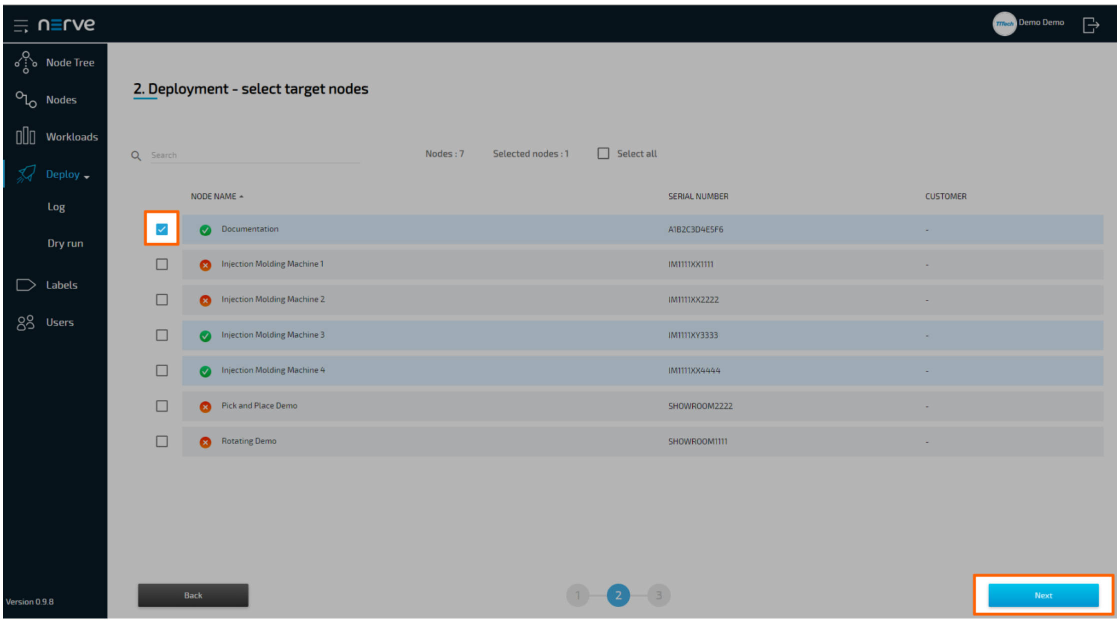
WORKLOAD NAME	CREATED	DESCRIPTION
PLC Data	29/07/2019	Samples machine data from PLC
DocumentationCODESYS	02/08/2019	This is a dummy workload for documentation purposes.

WORKLOAD VERSION

NAME	RELEASE NAME	CREATED
Controls	1.0.0	02/08/2019
TestControls	1.0.1	05/08/2019

6. In the next window, select one or more nodes from the list for deployment by ticking the checkboxes on the left.

7. Select **Next** in the lower-right corner.

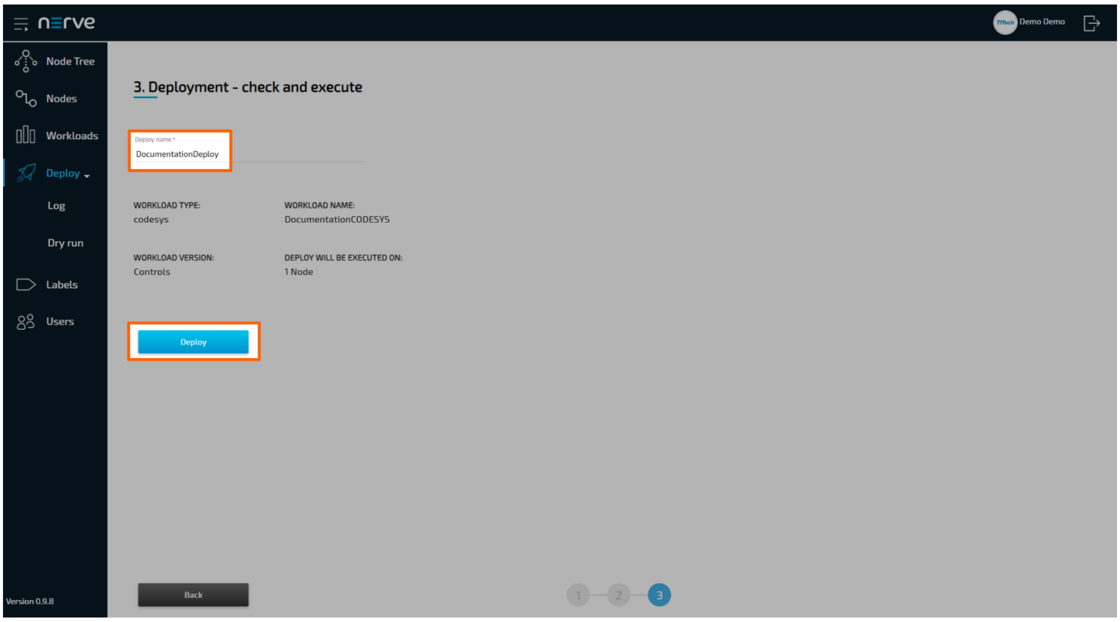


2. Deployment - select target nodes

Search Nodes : 7 Selected nodes : 1 □ Select all

	NODE NAME	SERIAL NUMBER	CUSTOMER
☑	✓ Documentation	A1B2C3D4E5F6	-
□	☒ Injection Molding Machine 1	IM1111XX1111	-
□	☒ Injection Molding Machine 2	IM1111XX2222	-
□	✓ Injection Molding Machine 3	IM1111XY3333	-
□	✓ Injection Molding Machine 4	IM1111XX4444	-
□	☒ Pick and Place Demo	SHOWROOM2222	-
□	☒ Rotating Demo	SHOWROOM1111	-

8. Select **Deploy** to execute the deployment.
Optional: Enter a **Deploy name** *above the* **Summary** *of the workload to make this deployment easy to identify. A timestamp is filled in automatically.*



3. Deployment - check and execute

Deploy name *
DocumentationDeploy

WORKLOAD TYPE:
codesys

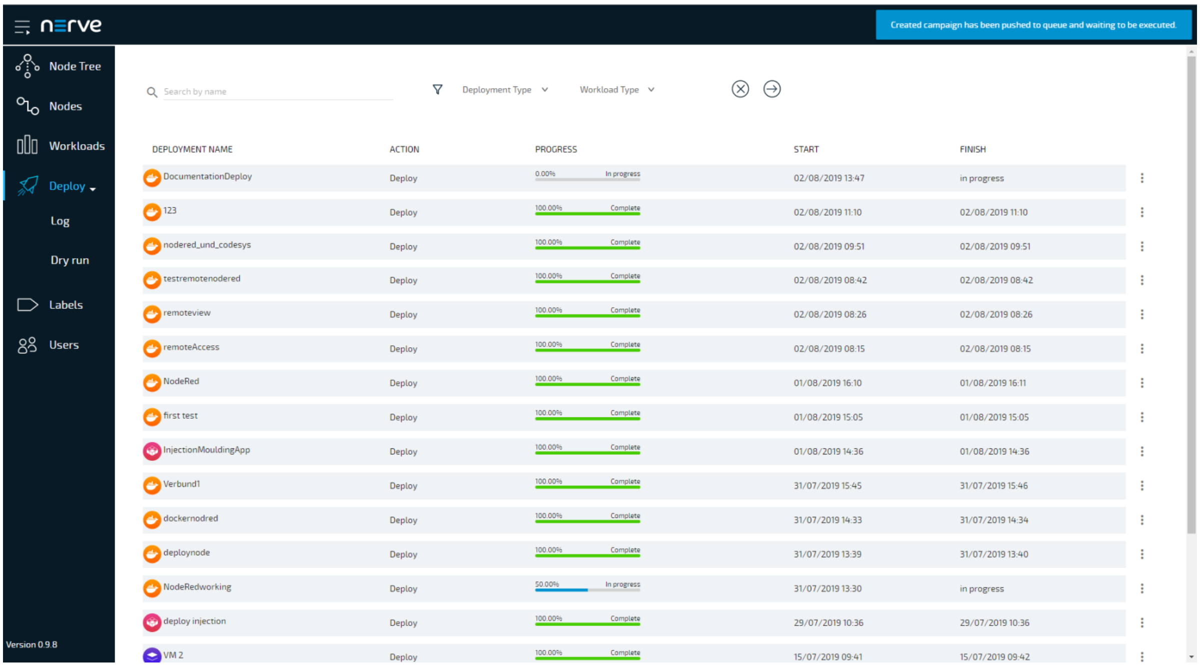
WORKLOAD NAME:
DocumentationCODESYS

WORKLOAD VERSION:
Controls

DEPLOY WILL BE EXECUTED ON:
1 Node

Deploy

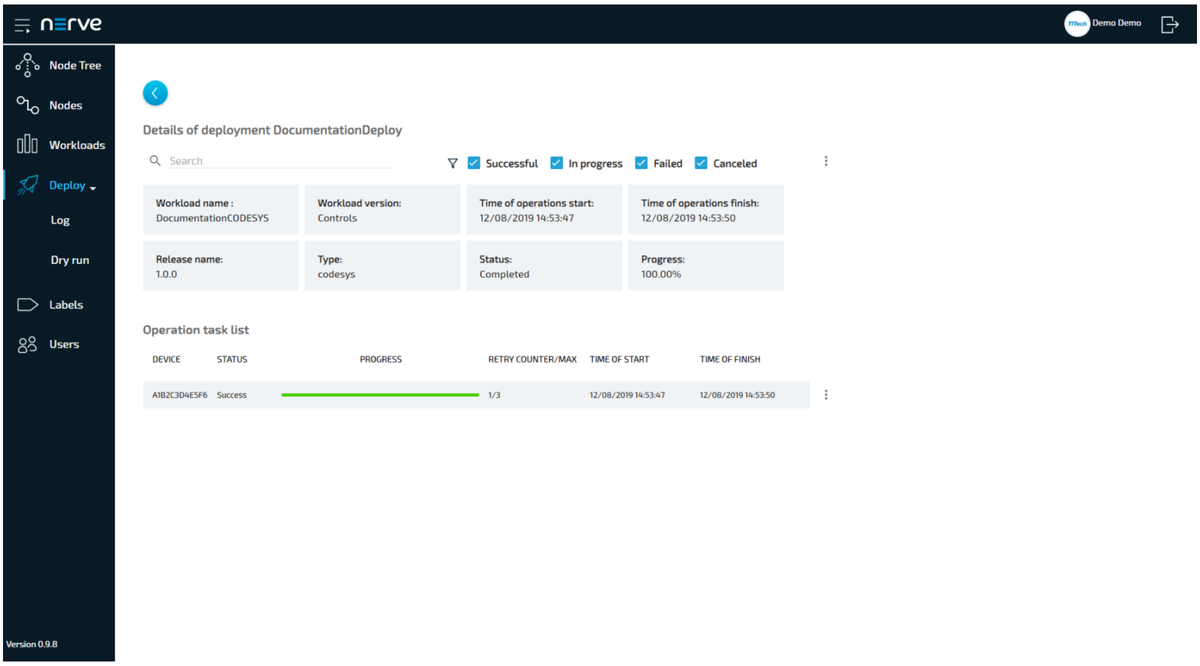
The Management System will deploy the log next. The current deployment is at the top of the list. The **Deploy name** chosen before is the name that identifies the deployment in the log.



Created campaign has been pushed to queue and waiting to be executed.

DEPLOYMENT NAME	ACTION	PROGRESS	START	FINISH
DocumentationDeploy	Deploy	0.00% In progress	02/08/2019 13:47	in progress
123	Deploy	100.00% Complete	02/08/2019 11:10	02/08/2019 11:10
nodered_und_codesys	Deploy	100.00% Complete	02/08/2019 09:51	02/08/2019 09:51
testremotenodered	Deploy	100.00% Complete	02/08/2019 08:42	02/08/2019 08:42
remoteview	Deploy	100.00% Complete	02/08/2019 08:26	02/08/2019 08:26
remoteAccess	Deploy	100.00% Complete	02/08/2019 08:15	02/08/2019 08:15
NodeRed	Deploy	100.00% Complete	01/08/2019 16:10	01/08/2019 16:11
first test	Deploy	100.00% Complete	01/08/2019 15:05	01/08/2019 15:05
InjectionMouldingApp	Deploy	100.00% Complete	01/08/2019 14:36	01/08/2019 14:36
Verbund1	Deploy	100.00% Complete	31/07/2019 15:45	31/07/2019 15:46
dockernodred	Deploy	100.00% Complete	31/07/2019 14:33	31/07/2019 14:34
deploynode	Deploy	100.00% Complete	31/07/2019 13:39	31/07/2019 13:40
NodeRedworking	Deploy	50.00% In progress	31/07/2019 13:30	in progress
deploy injection	Deploy	100.00% Complete	29/07/2019 10:36	29/07/2019 10:36
VM 2	Deploy	100.00% Complete	15/07/2019 09:41	15/07/2019 09:42

Check the progress of the current deployment and click the workload to see a more detailed view.



Details of deployment DocumentationDeploy

Search ☑ Successful ☑ In progress ☑ Failed ☑ Canceled

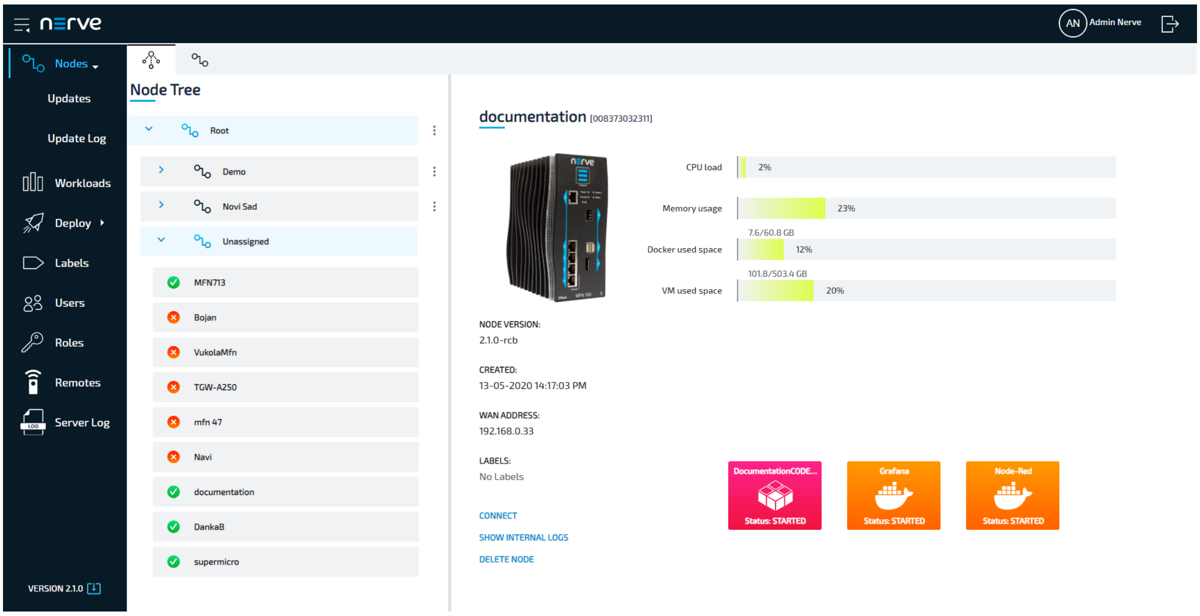
Workload name : DocumentationCODESYS	Workload version: Controls	Time of operations start: 12/08/2019 14:53:47	Time of operations finish: 12/08/2019 14:53:50
Release name: 1.0.0	Type: codesys	Status: Completed	Progress: 100.00%

Operation task list

DEVICE	STATUS	PROGRESS	RETRY COUNTER/MAX	TIME OF START	TIME OF FINISH
A1B2C3D4E5F6	Success		1/3	12/08/2019 14:53:47	12/08/2019 14:53:50

Confirm the deployment by viewing the workload in the node details view in the node tree. Select **Nodes** in the navigation on the left and select the node tree tab

on the right. Select the node that has workloads deployed.



documentation [008373032311]

CPU load	2%
Memory usage	23%
Docker used space	7.6/60.8 GB 12%
VM used space	101.8/503.4 GB 20%

NODE VERSION:
2.1.0-rcb

CREATED:
13-05-2020 14:17:03 PM

WAN ADDRESS:
192.168.0.33

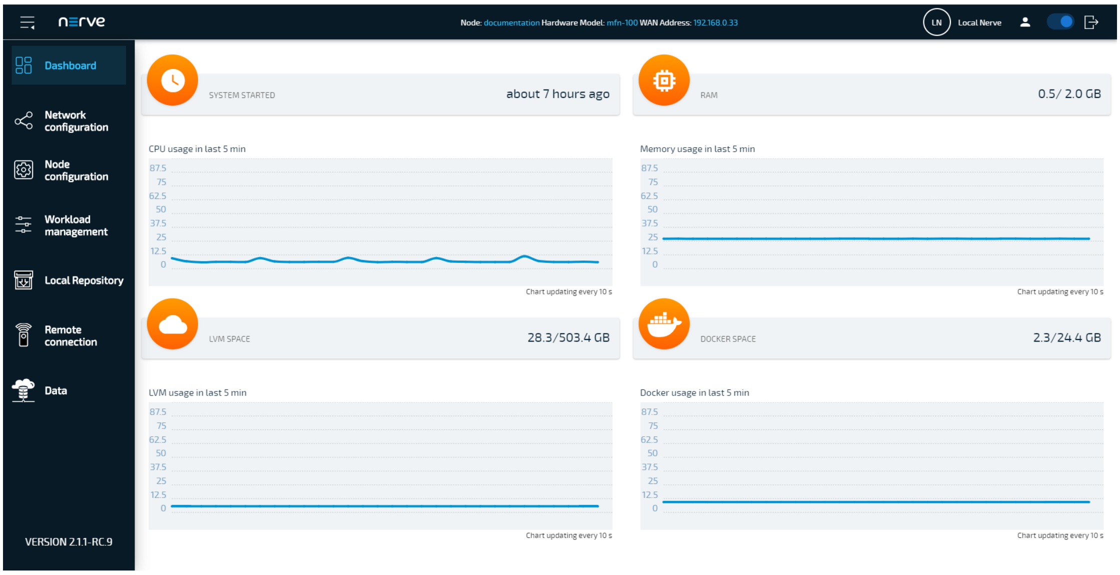
LABELS:
No Labels

Clicking a workload tile leads to the workload control screen. This is where workloads can be controlled. However, CODESYS workloads can only be controlled through the Local UI.
All workloads are started as soon as they are deployed.

Connecting to the Local UI

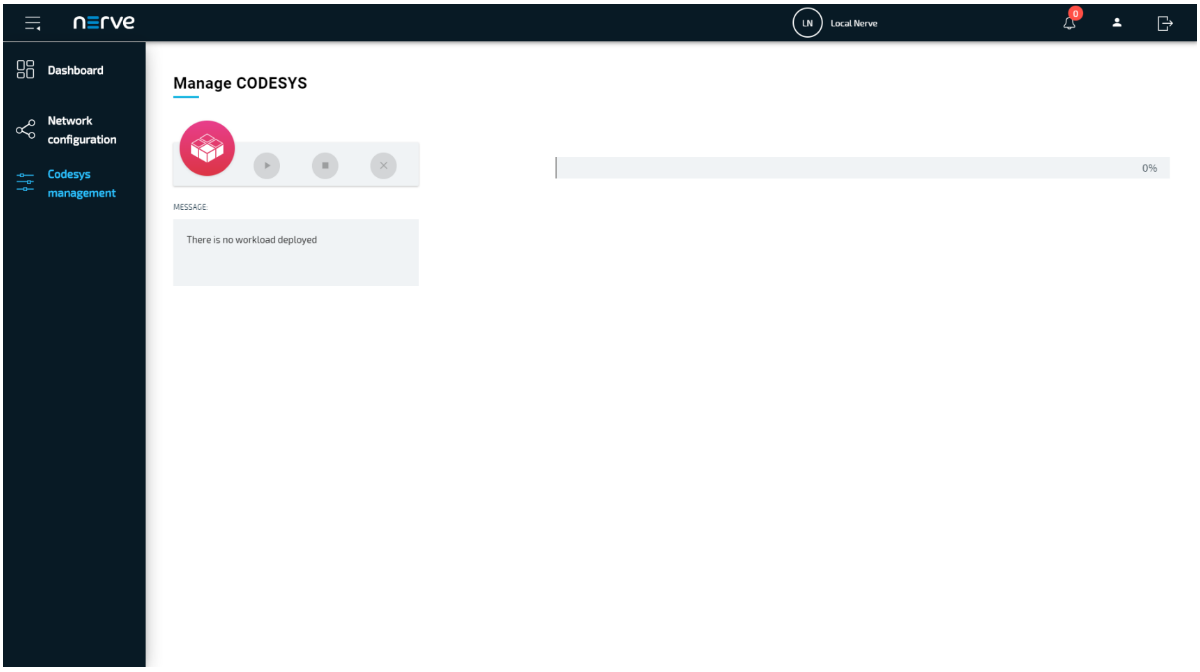
In order to access the Local UI, connect a workstation to the console port **P1** of the MFN 100 and configure the network adapter of the workstation. The IP address of the network adapter has to be in the range from `172.20.2.5` to `172.20.2.254` with a `255.255.255.0` subnet mask. The credentials for the Local UI found in the customer profile are also required.

1. Follow this link to connect to the Local UI: http://172.20.2.1:3333/
2. Log in with the credentials from the customer profile to reach the main page of the Local UI.



Control of CODESYS applications

CODESYS workloads can only be controlled in the Local UI, as operation of a CODESYS workload may have an impact on machine operation and therefore should not be controlled remotely. Select **Workload management** in the menu on the left-hand side to reach the interface for controlling a CODESYS application running on the Nerve Device:



MESSAGE:

There is no workload deployed

Function Name	Description
Start	This starts the CODESYS application.

Function Name	Description
Stop	This stops the CODESYS application and it is reset to its initial values.
Remove	This removes the CODESYS application from the Nerve Device. To deploy the CODESYS application again, do so through the Management System.
Message	CODESYS workloads have the following set of messages: • "Preparing files for installation" • "Starting CODESYS application" • "CODESYS application started" • "Stopping CODESYS application" • "CODESYS application stoppped" • "Removing CODESYS application file" • "An unexpected error has occurred. <errormessage>" Here, <errormessage> is a message that is sent by the CODESYS Development System.

NOTE

It takes a moment before CODESYS applications are actually started, stopped or removed.

Uploading new CODESYS applications

In order to work with new CODESYS applications on the MFN 100, new CODESYS workloads need to be provisioned in the Management System. Here, provisioning is the creation of a workload and its storage in the workload repository of the Management System so that it can be deployed to nodes. This requires configuration of the workload and the upload of the required files to the Management System. After that, the workload can be deployed to nodes.

Before the workload can be provisioned, however, a CODESYS application has to be loaded into the CODESYS runtime first. Refer to the introduction to working with CODESYS and the MFN 100 first before continuing.

NOTE

Note that **app1** is already loaded into the CODESYS runtime by default.

Once a CODESYS application has been loaded into the MFN 100, the following steps have to be taken before provisioning a CODESYS workload:

- Creating the ZIP file of the CODESYS application
- Transferring the ZIP file to a local workstation

Also the workstation needs to be connected to the console port **P1** of the MFN 100 and the network adapter of the workstation needs to be configured. The IP address of the network adapter has to be in the range from `172.20.2.5` to `172.20.2.254`.

Creating the ZIP file on the Nerve Device

First, the CODESYS project needs to be zipped on the Nerve Device before it can be copied from the CODESYS runtime. This is done through the Local UI.

1. Connect to the Local UI as described above.
2. Select **Workload management** in the navigation on the left.
3. Click **Download CODESYS app archive**.
4. Select **YES** in the pop-up. Note that the CODESYS application will be stopped.

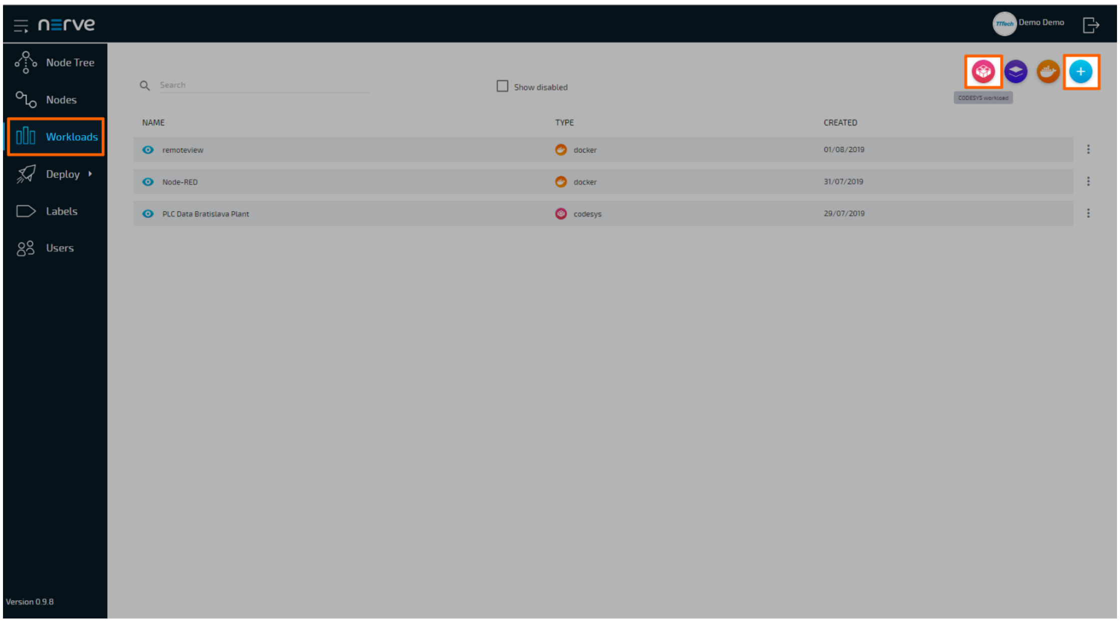
The ZIP file is automatically downloaded to the workstation and a CODESYS workload can now be provisioned in the Management System.

Provisioning a CODESYS workload

The following instructions cover the basic requirements for provisioning a CODESYS workload. Optional settings will be left out. Extended options are addressed in the user guide.

1. Log in to the Management System.
2. Select **Workloads** in the left-hand menu.
3. Select the plus symbol in the upper-right corner.

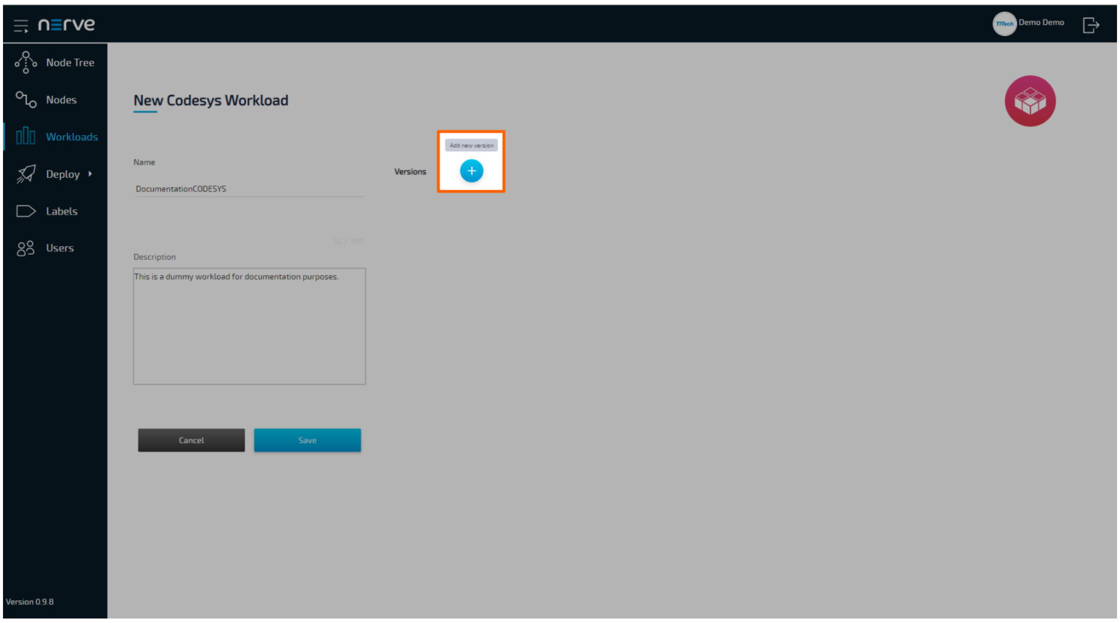
4. Select the CODESYS symbol (**CODESYS workload**) on the left of the three symbols that expanded.



NAME	TYPE	CREATED
remoteview	docker	01/08/2019
Node-RED	docker	31/07/2019
PLC Data Bratislava Plant	codesys	29/07/2019

5. Enter a name for the workload in the new window.

6. Select the plus symbol next to **Versions** to add a new version of the workload.



New Codesys Workload

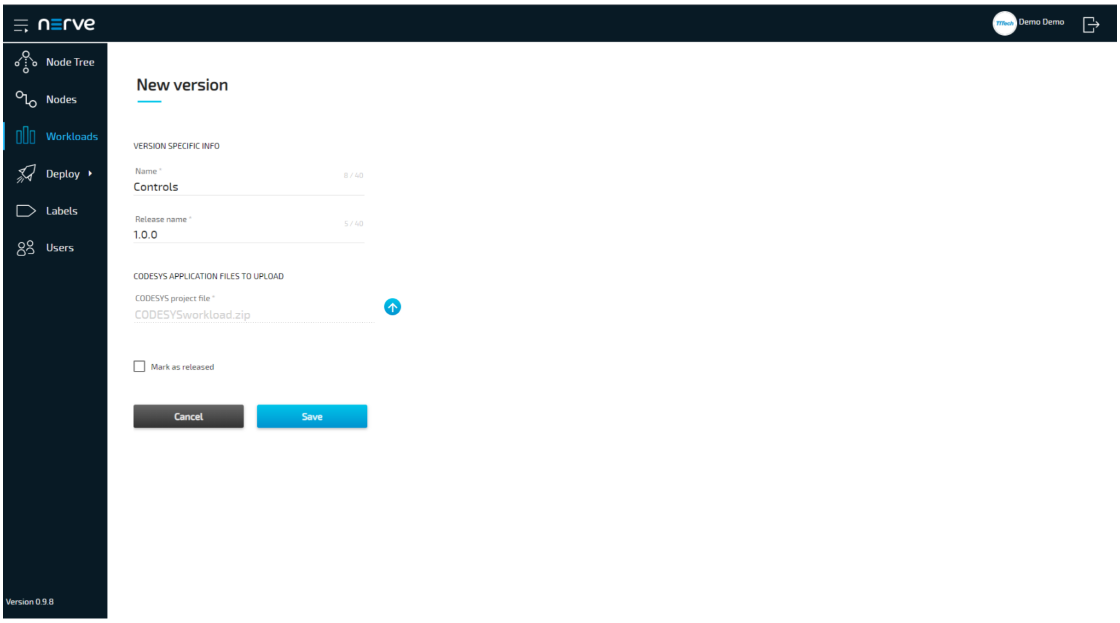
Name
DocumentationCODESYS

Description
This is a dummy workload for documentation purposes.

Versions

7. Enter the following information in the new window:

Item	Description
Name	Enter a **Name** for the version of this workload.
Release name	Enter a **Release name** for the version of this workload.
CODESYS project file	Click the **upward arrow** symbol to open the file browser and add the CODESYS application ZIP file. This is the ZIP file that has been created before.



New version

VERSION SPECIFIC INFO

Name * 8 / 40
Controls

Release name * 5 / 40
1.0.0

CODESYS APPLICATION FILES TO UPLOAD

CODESYS project file *
CODESYSworkload.zip

□ Mark as released

8. Click **Save**.

The workload has now been provisioned and is ready to be deployed in the **Deploy** menu.

Connecting new sensors and actuators

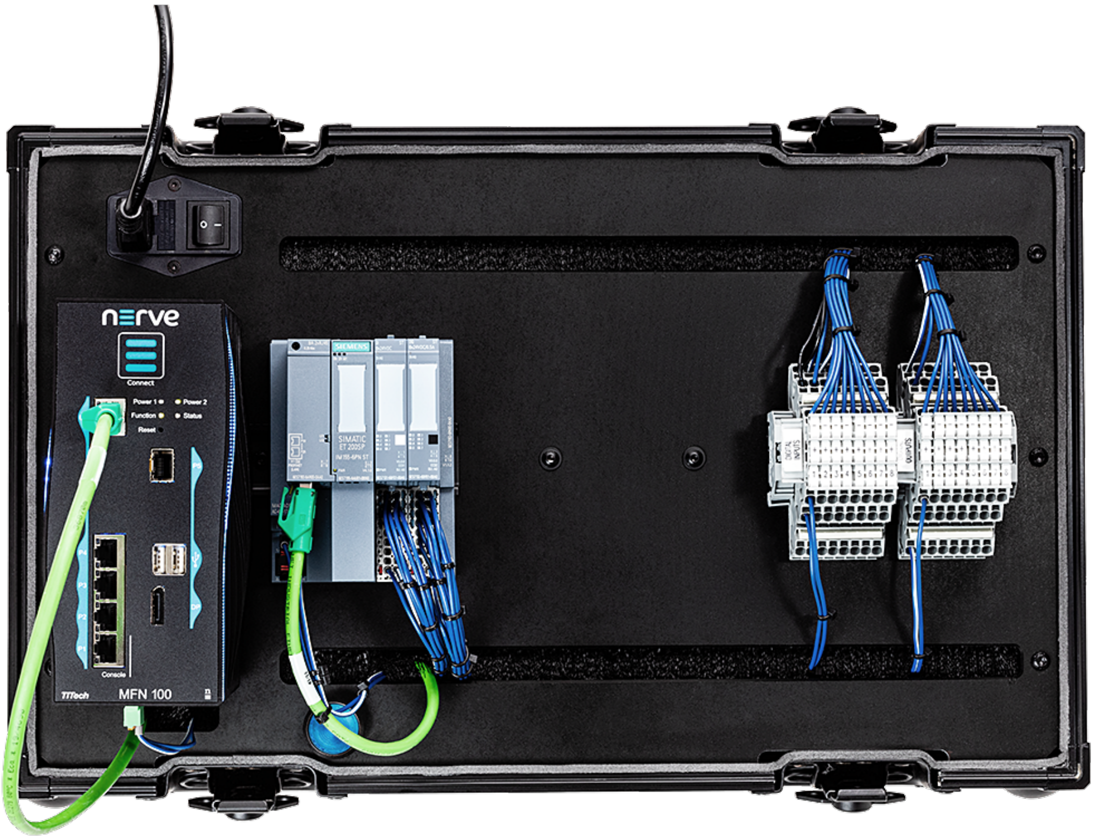
Add sensors and actuators to the kit to execute control applications and visualize the corresponding data. With the delivered set-up it is possible to add up to 7 additional inputs and outputs. I/O blocks can also be added to increase the number or type of inputs and outputs.

Wiring a new sensor or actuator

NOTE

- Before wiring any new components, review the Nerve Blue Kit Circuit Diagram. Contact a sales representative for more information.
- Disconnect the power supply from the power outlet before wiring new I/O devices to prevent injury to persons or damage to equipment.
- Only staff with knowledge about electrical circuits should perform the tasks described in this section.

The inputs and outputs of the SIMATIC ET200 SP I/O module are wired to the terminal blocks on the right hand side of the kit. The left terminal block is used to connect digital inputs. The right terminal block is used to connect digital outputs.

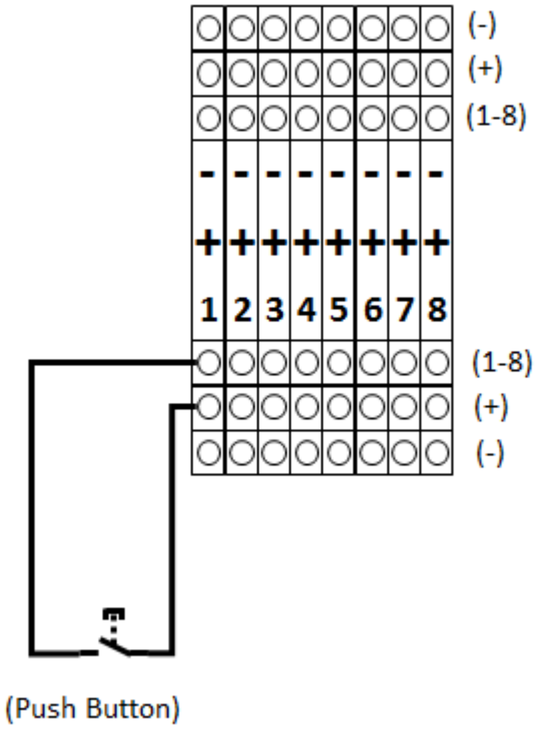


Connecting a digital input

This section shows how to connect an additional digital input to the kit. A push button is used for demonstration purposes.

1. Connect the power supply of the button to the middle row of the I/O module (+24V).

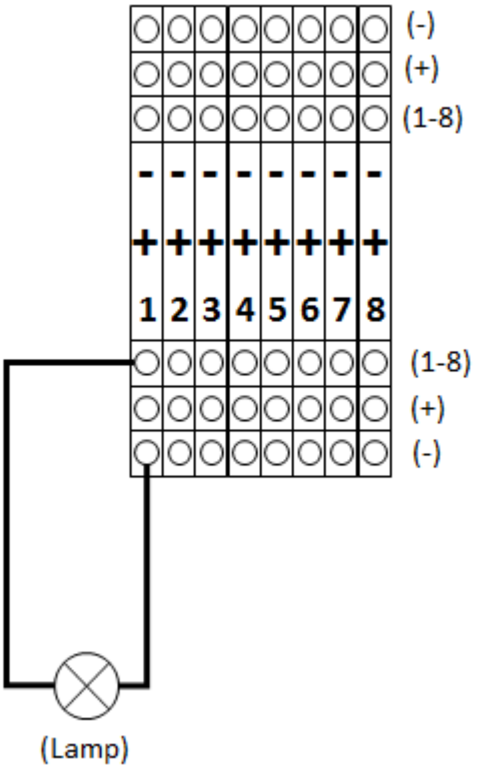
2. Connect the input to the top row.



Connecting a digital output

This section shows how to connect an additional digital output to the kit. A lamp is used for demonstration purposes.

1. Connect the lamp to the top row of the I/O module.
2. Connect the common wire to the bottom row to close the electrical circuit.



After wiring the sensors or actuators to the inputs or outputs respectively switch the kit back on.

The next chapter describes how to assign variables to the inputs and outputs in order to read data from newly connected sensors or control actuator functionality.

First steps with CODESYS and the MFN 100

This chapter will give an introduction on how to start working with the integrated soft PLC in the MFN 100. First, some configuration and installation of files and libraries are required.

NOTE

- Download the CODESYS Development System V3 from store.codesys.com for this chapter.
  We recommend version 3.5 SP14 (32 bit) or newer.
- Connect the workstation to the console port **P1** of the MFN 100.

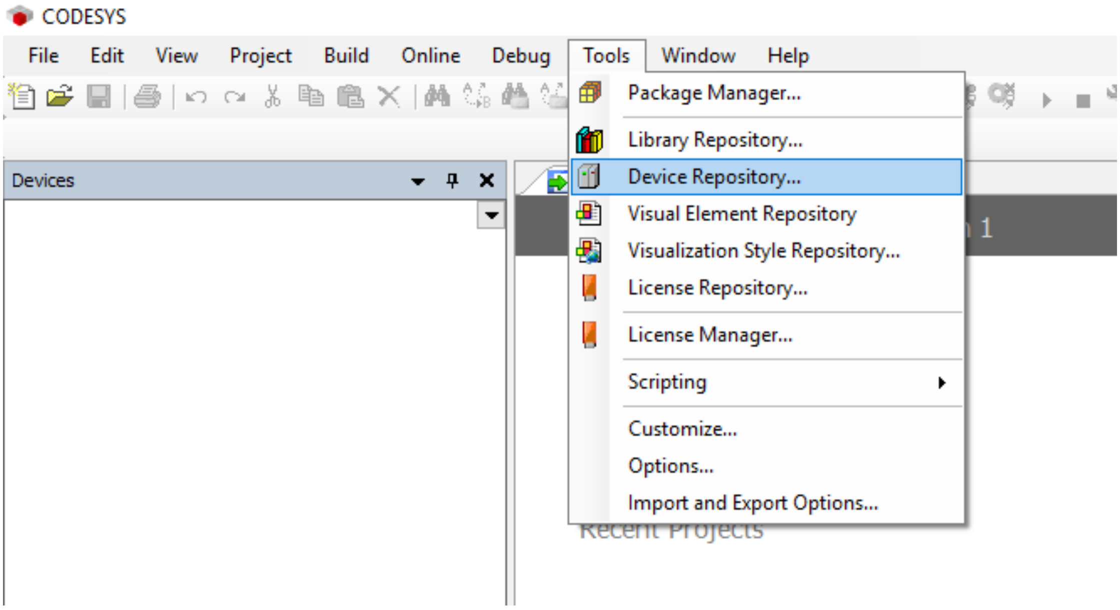
Installing the device descriptions

After downloading and installing the CODESYS Development System on the workstation, install the device descriptions of the MFN 100 and the SIMATIC ET200 SP I/O module in the CODESYS Development System. The device descriptions have the following filenames:

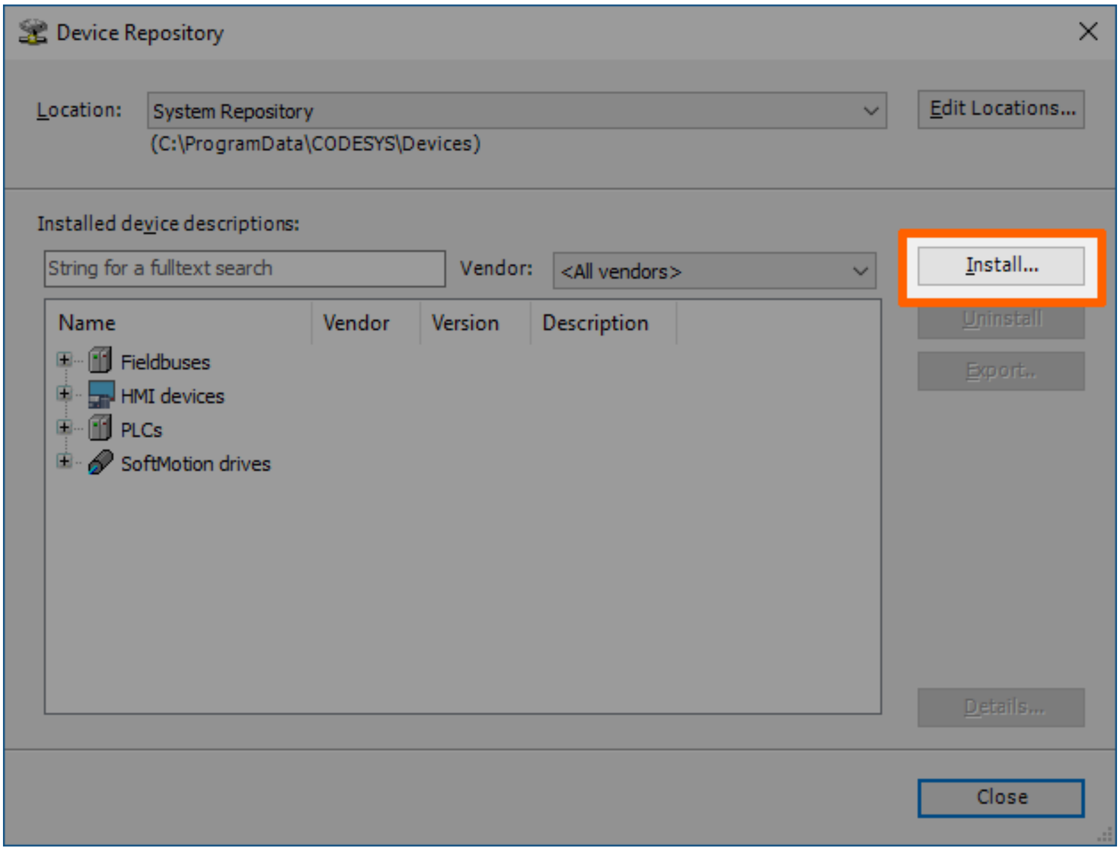
MFN 100	`Nerve_MFN_100_V3.5.XX.X.devdesc.xml`
SIMATIC ET200 SP	`GSDML-V2.34-Siemens-ET200SP-20180926.xml`

XX.X stands for the current version of the CODESYS Development System. The device descriptions of the MFN 100 and the SIMATIC ET200 SP I/O module are available at the Nerve Software Center.

1. Start the CODESYS Development System.
2. Go to **Tools > Device Repository**.



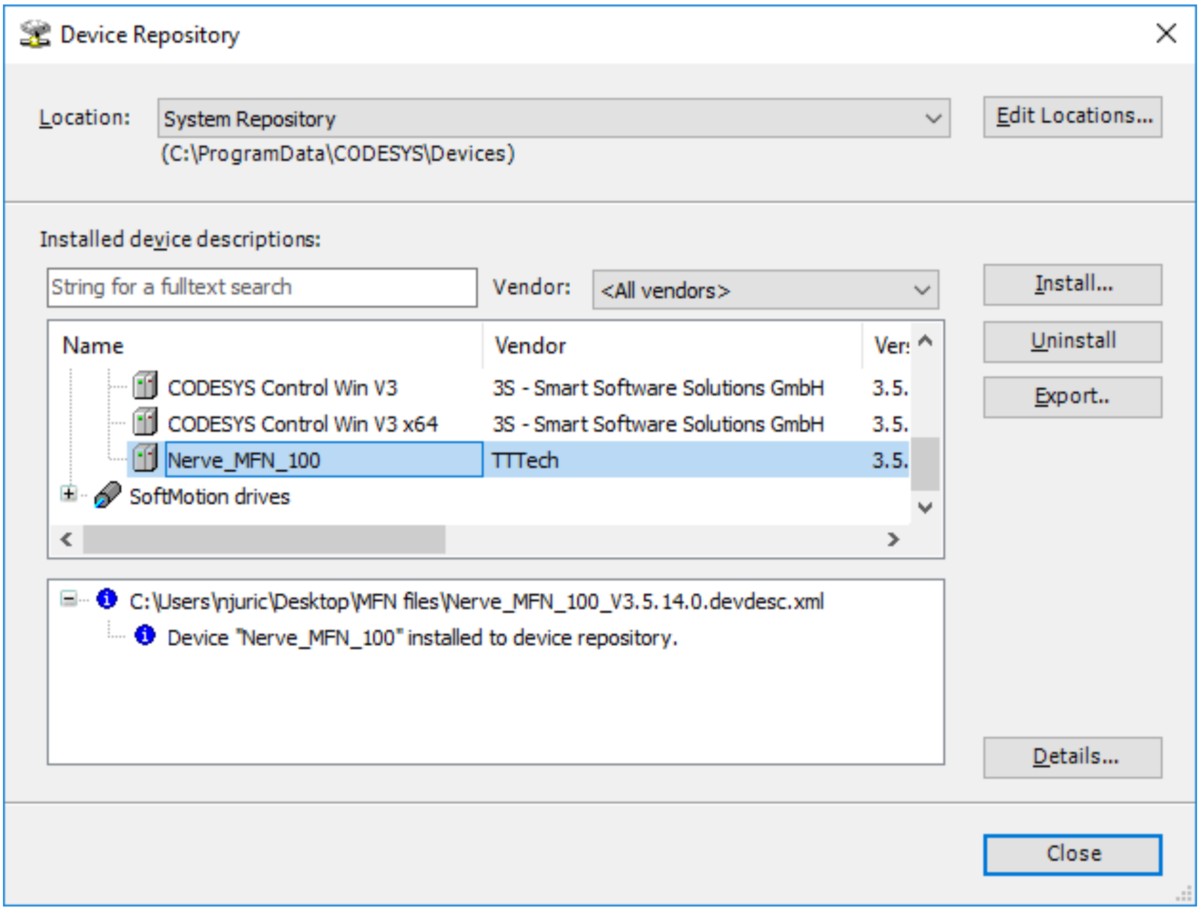
3. Click **Install**.



4. Go to the directory of the previously downloaded device description.

5. Select the device description of the MFN 100.
6. Click **Open**.
7. Repeat steps 3 to 6 to install the device description SIMATIC ET200 SP I/O module.

When the installation was successful, the MFN 100 and the SIMATIC ET200 SP I/O module will appear in the list of device descriptions in the middle of the window. Close the window afterwards.



Location: System Repository
(C:\ProgramData\CODESYS\Devices)

Installed device descriptions:

String for a fulltext search

Vendor: <All vendors>

Name	Vendor	Ver:
CODESYS Control Win V3	3S - Smart Software Solutions GmbH	3.5.
CODESYS Control Win V3 x64	3S - Smart Software Solutions GmbH	3.5.
Nerve_MFN_100	TTTech	3.5.
SoftMotion drives		

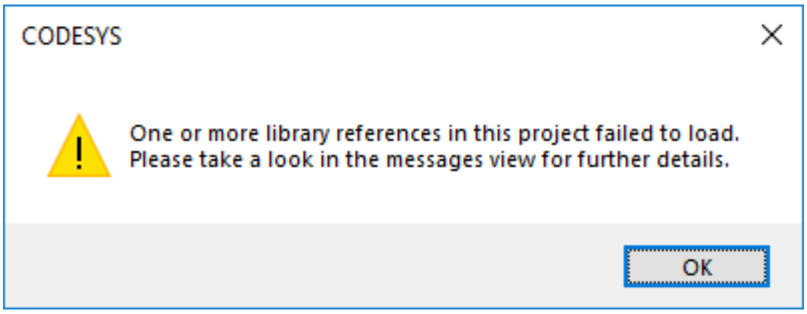
C:\Users\njuric\Desktop\MFN files\Nerve_MFN_100_V3.5.14.0.devdesc.xml

Device "Nerve_MFN_100" installed to device repository.

After installing the device description the CODESYS Development System can be worked with. However, libraries and device descriptions of generic devices might be missing so that the CODESYS Development System can work properly.

Downloading missing libraries

The error message for missing libraries might appear when opening or creating a CODESYS project. The CODESYS Development System identifies the missing libraries automatically but the following process might have to be repeated a few times.

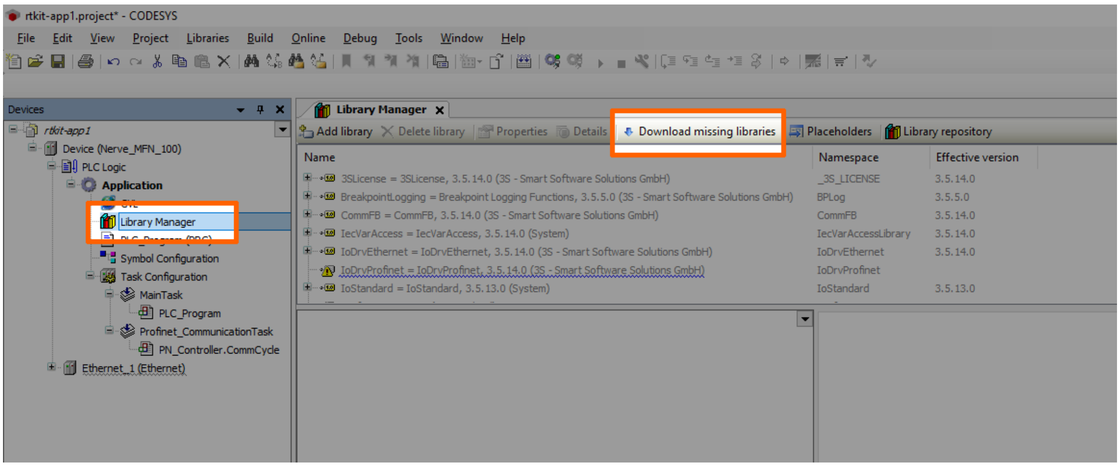


One or more library references in this project failed to load.
Please take a look in the messages view for further details.

1. Open or create a CODESYS project.
2. If the error message about missing libraries appears, click **OK**.

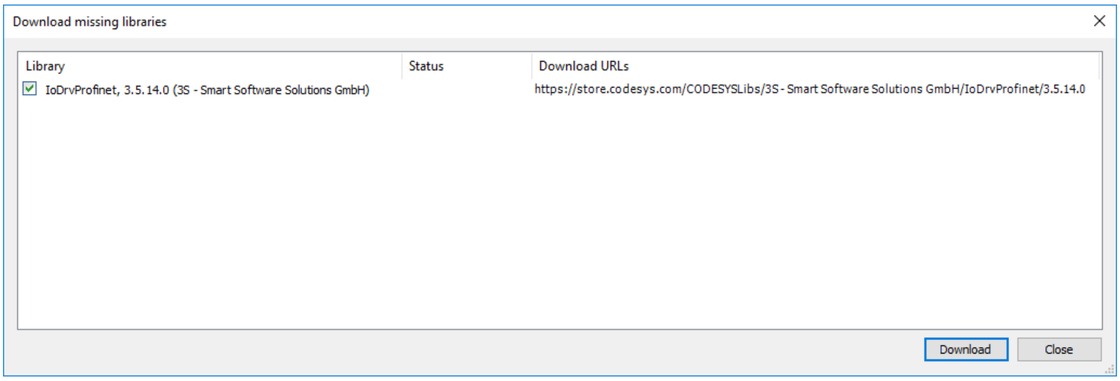
3. Double-click **Library Manager** in the tree view on the left.

4. Click **Download missing libraries**.



Name	Namespace	Effective version
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.14.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
CommFB = CommFB, 3.5.14.0 (3S - Smart Software Solutions GmbH)	CommFB	3.5.14.0
IecVarAccess = IecVarAccess, 3.5.14.0 (System)	IecVarAccessLibrary	3.5.14.0
IoDrvEthernet = IoDrvEthernet, 3.5.14.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	3.5.14.0
IoDrvProfinet = IoDrvProfinet, 3.5.14.0 (3S - Smart Software Solutions GmbH)	IoDrvProfinet	
IoStandard = IoStandard, 3.5.13.0 (System)	IoStandard	3.5.13.0

5. Click **Download** in the new window.



Library	Status	Download URLs
☑ IoDrvProfinet, 3.5.14.0 (3S - Smart Software Solutions GmbH)		https://store.codesys.com/CODESYSLibs/3S - Smart Software Solutions GmbH/IoDrvProfinet/3.5.14.0

6. Click **Close** when the download is finished.

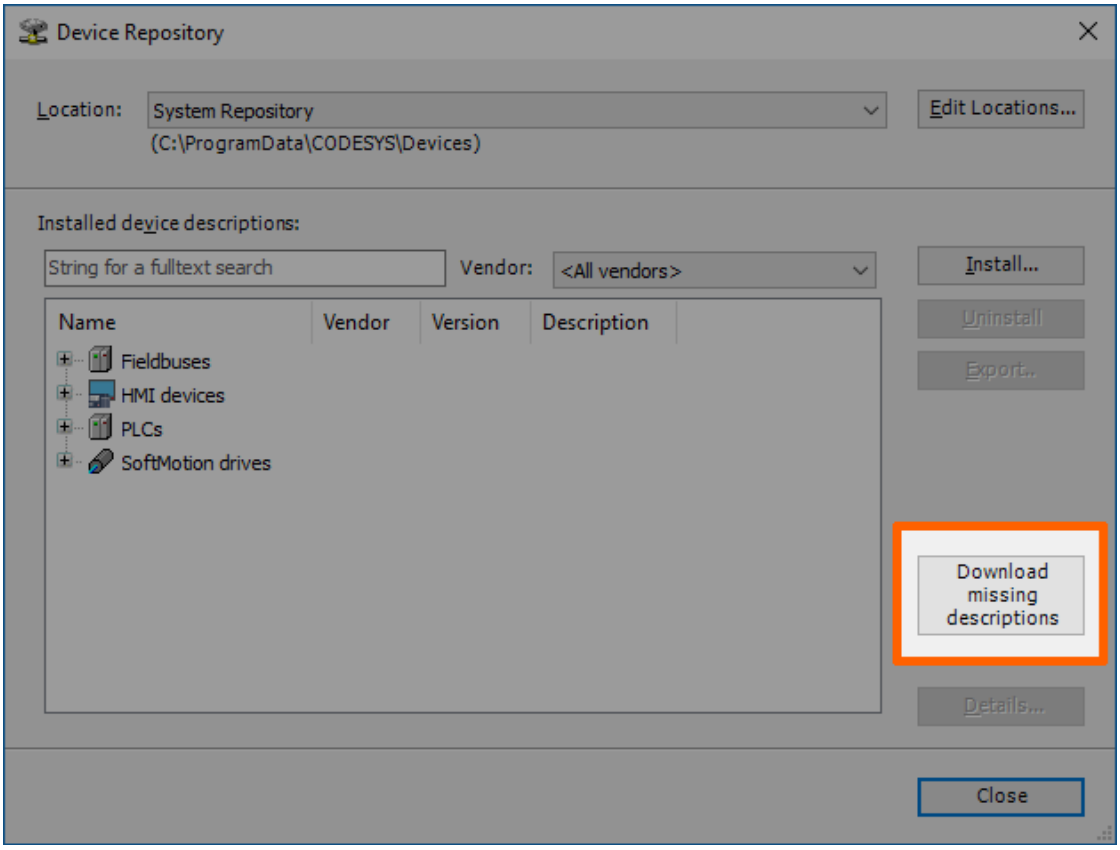
7. Repeat steps 3 to 5 until no more libraries appear in the download window.

Downloading missing device descriptions

Apart from the device description for the MFN 100, device descriptions of generic devices may be missing for the CODESYS Development System to function as intended. The CODESYS Development System will identify the missing device descriptions automatically but this time it will not generate an error message unless a CODESYS application is loaded into the MFN 100.

1. Click **Tools > Device Repository**.

2. Click **Download missing descriptions**.



NOTE

If no device descriptions of generic devices are missing, the button for downloading missing descriptions will not appear.

3. Click **Download** in the new window.

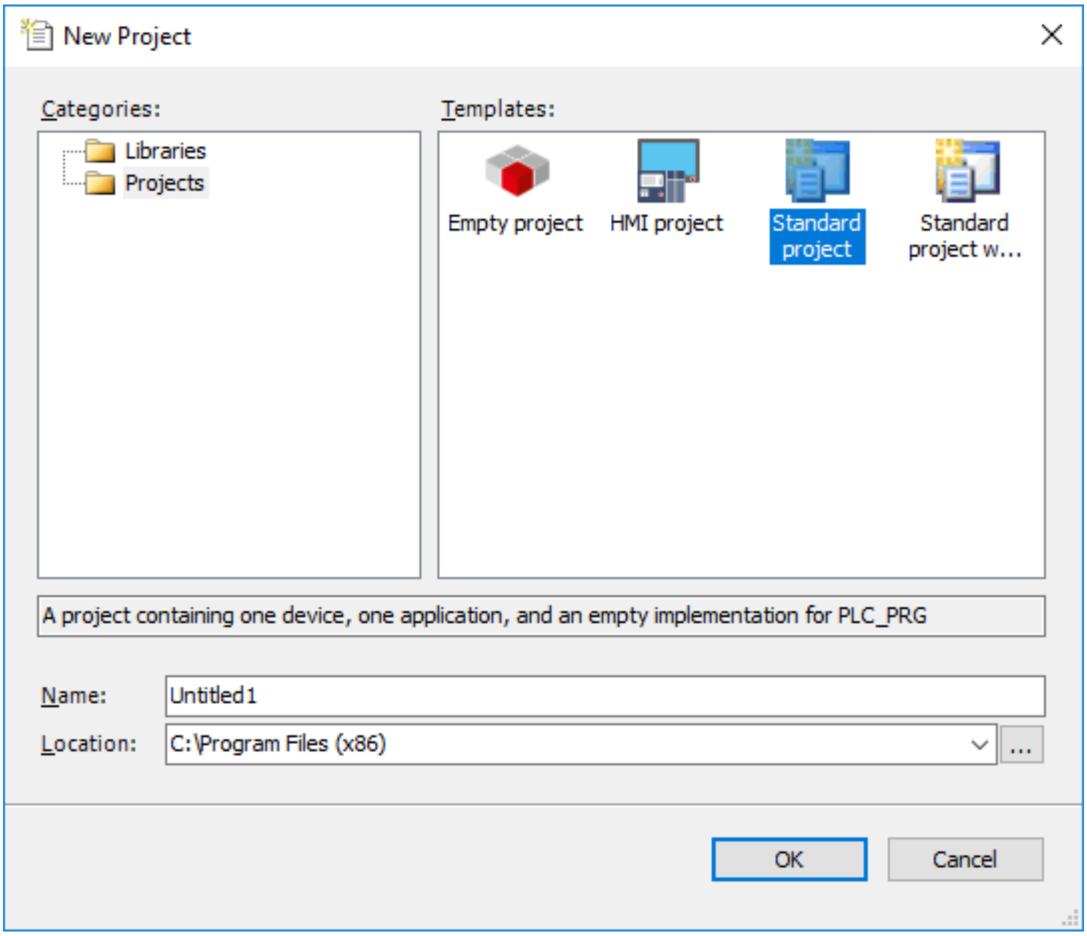
4. Click **Close** when the download is finished.

Creating a new CODESYS project

This example shows how to create a new project in the CODESYS Development System. The easiest way to get started is to create a **Standard project**.

1. Start CODESYS
2. Go to **File > New Project**.
3. Click **Standard project** on the right side among the templates.
4. Enter a name for the project.
5. Choose a **Location** where the project will be saved.

6. Click **OK** to save the project.



New Project

Categories:

- Libraries
- Projects

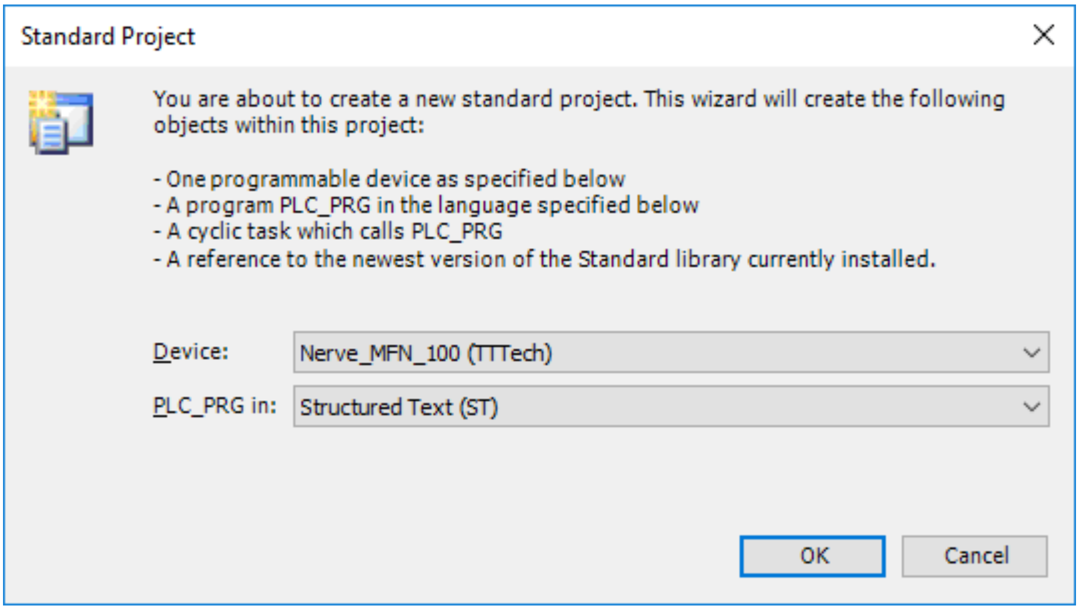
Templates:

A project containing one device, one application, and an empty implementation for PLC_PRG

Name: Untitled1

Location: C:\Program Files (x86)

7. Select **Nerve_MFN_100 (TTTech)** as the device.



Standard Project

You are about to create a new standard project. This wizard will create the following objects within this project:

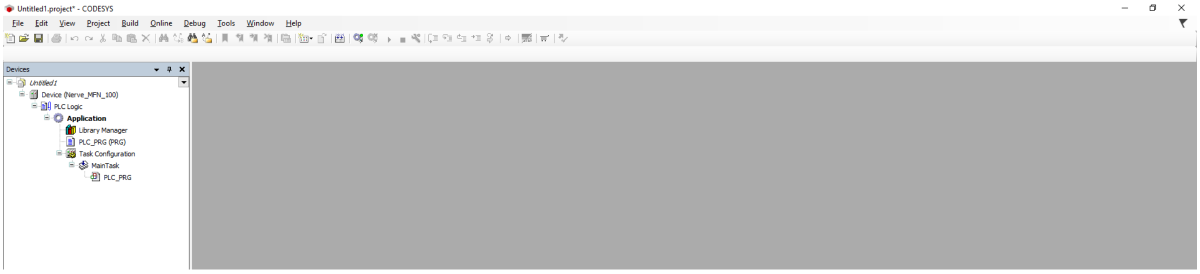
- One programmable device as specified below
- A program PLC_PRG in the language specified below
- A cyclic task which calls PLC_PRG
- A reference to the newest version of the Standard library currently installed.

Device: Nerve_MFN_100 (TTTech)

PLC_PRG in: Structured Text (ST)

8. Click **OK**.

The result is an empty project that is open in the main view of CODESYS.

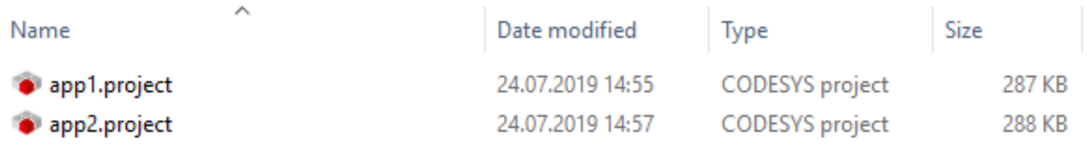


Working with the default applications

To work with existing applications first, modify the default applications `app1.project` and `app2.project`. They can be downloaded from the Nerve Software Center under **Example applications**.

1. Start CODESYS.
2. Go to **File > Open Project**.
3. Select the download location of the default applications.

4. Select the application to work with.



Name	Date modified	Type	Size
app1.project	24.07.2019 14:55	CODESYS project	287 KB
app2.project	24.07.2019 14:57	CODESYS project	288 KB

5. Click **Open**.

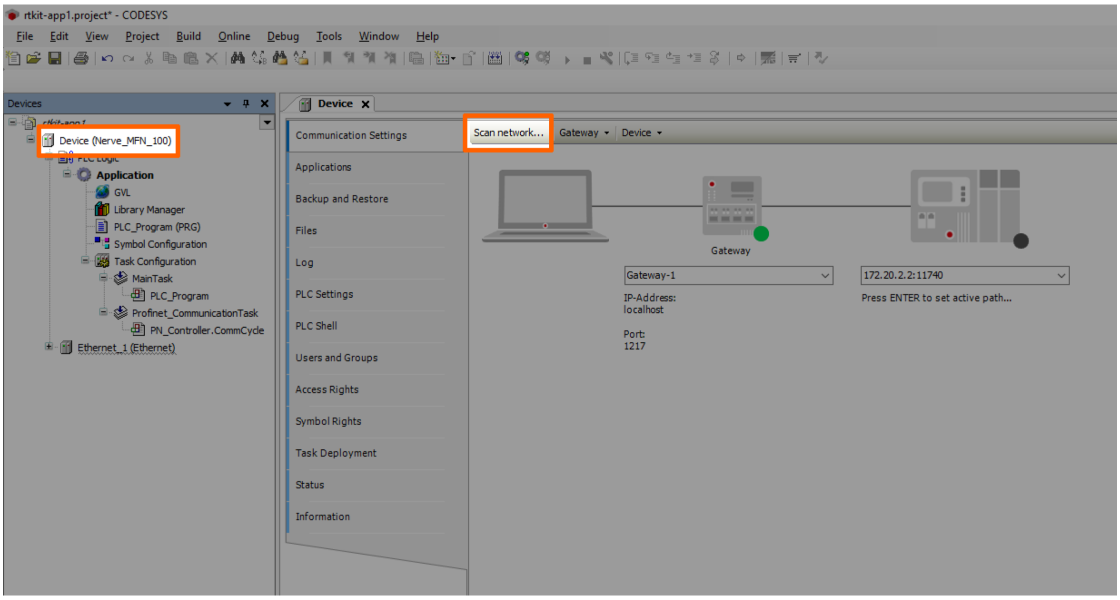
When opening the default applications for the first time, some libraries and device descriptions will be missing. Follow the instructions above to see how to download the missing files.

Connecting to the MFN 100

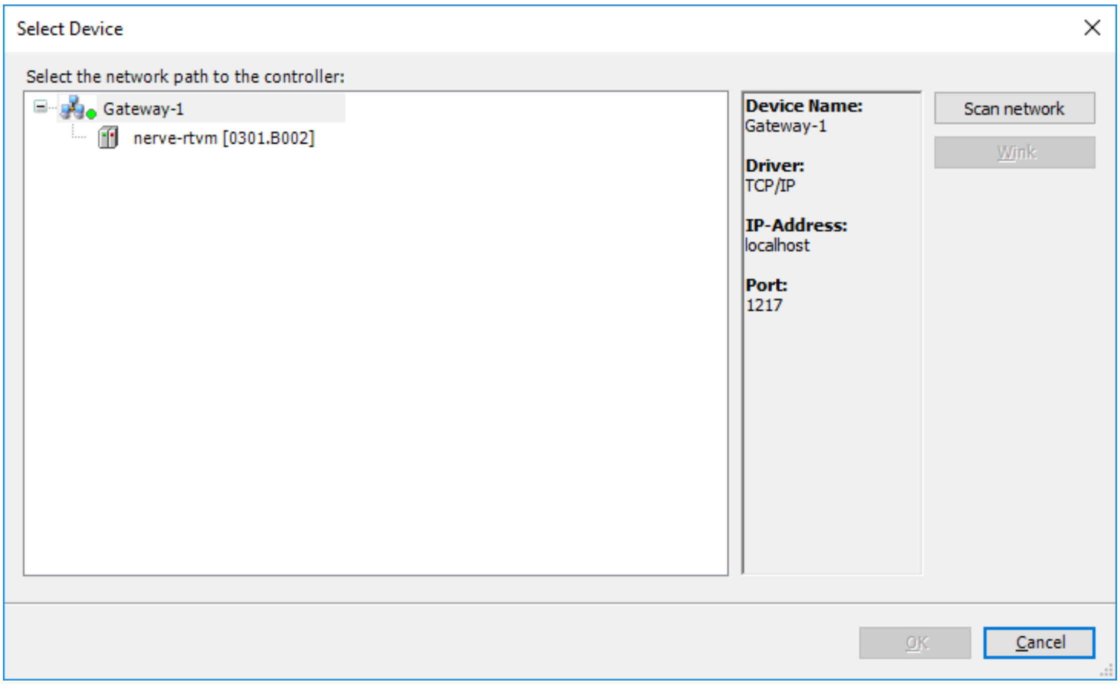
Before downloading CODESYS applications to the MFN 100, make sure that the device description of the MFN 100 is installed in the CODESYS Development System.

1. Open or create a CODESYS project.
2. Double-click **Device (Nerve_MFN_100)** in the tree view on the left.

3. Go to **Communication Settings > Scan network...**.



4. Select the MFN 100 (here **nerve-rtvm [XXXX.XXXX]**) in this window.



Select Device

Select the network path to the controller:

- Gateway-1
  - nerve-rtvm [0301.B002]

Device Name:
Gateway-1

Driver:
TCP/IP

IP-Address:
localhost

Port:
1217

NOTE

When more than one network is active on the workstation, it sometimes happens that the MFN 100 cannot be found. Continue reading if the MFN 100 does not appear in this window.

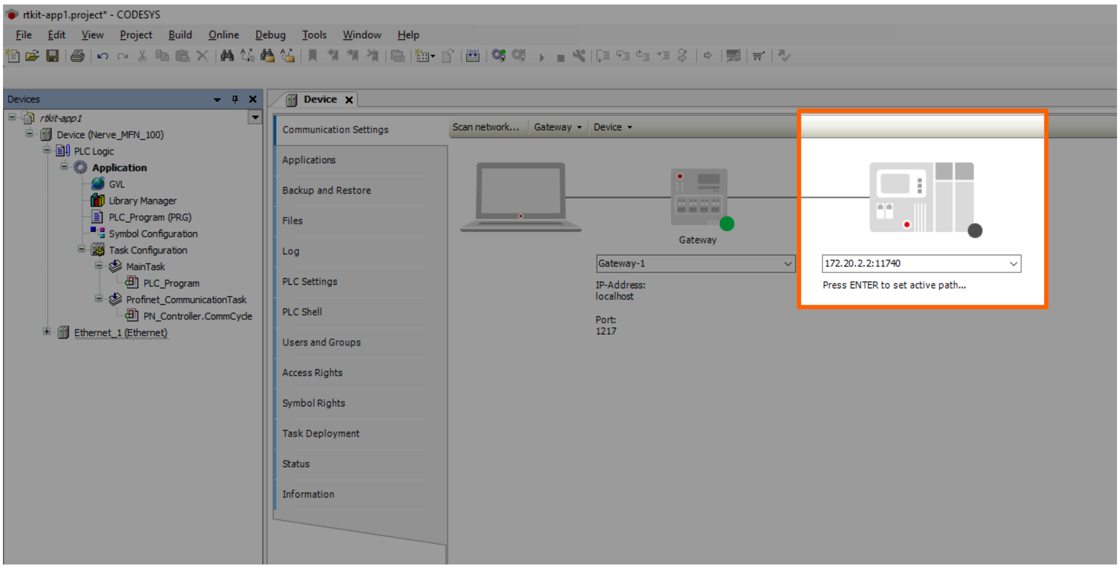
5. Click **OK**.

Typically the MFN 100 will be found automatically. If the MFN 100 cannot be found, enter the IP address and port of the CODESYS runtime manually.

1. Double-click **Device (Nerve_MFN_100)** in the tree view on the left.

2. Go to **Communication Settings** in the middle of the window.

3. Enter `172.20.2.2:11740` in the text box under the device on the right.



IP-Address:
localhost

Port:
1217

172.20.2.2:11740

Press ENTER to set active path...

4. Press Enter.

The CODESYS Development System is now connected to the MFN 100 and applications can be downloaded into the CODESYS runtime.

Downloading an application to the MFN 100

CODESYS applications can be loaded directly into the MFN 100. However, before downloading an application into the MFN 100 it needs to be free of errors.

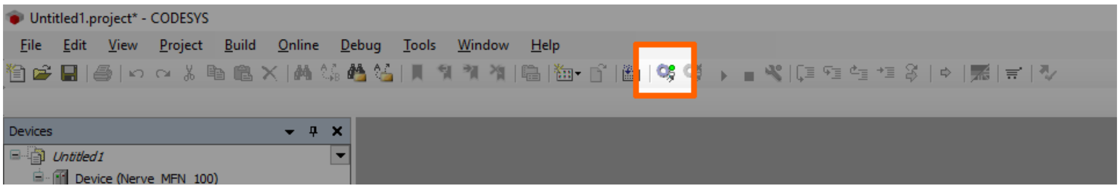
The process of downloading an application is slightly different when downloading an entirely new application into the MFN 100 or updating an application that has already been downloaded into the MFN 100. Continue with Downloading an Updated Application to the MFN 100 further down below if an application is being updated.

Downloading a new Application to the MFN 100

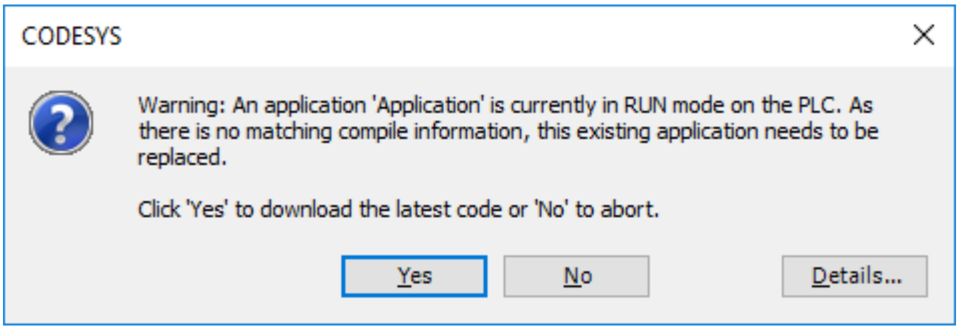
After creating a project and finishing with programming, the CODESYS application can be downloaded into the MFN 100 directly.

1. Open the CODESYS project to load into the MFN 100.

2. Click the **Login** symbol in the CODESYS menu bar.


Untitled1.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
Untitled1
Device (Nerve_MFN_100)

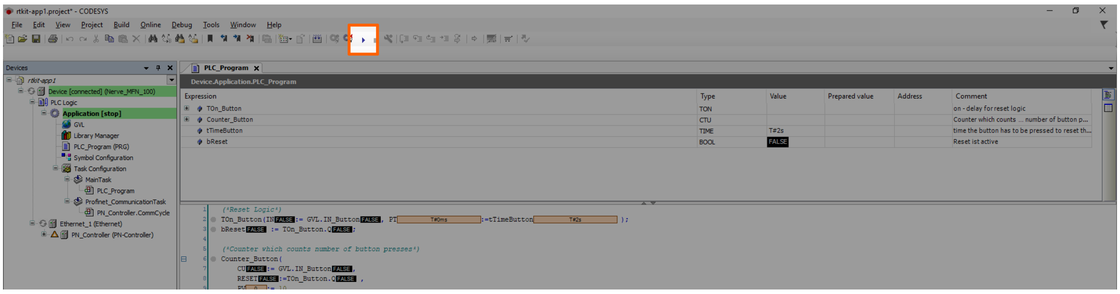
3. Click **Yes** in the pop-up window.



Warning: An application 'Application' is currently in RUN mode on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

4. The application is stopped now. Click the **Play** symbol in the CODESYS menu bar.


rtkit-app1.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
rtkit-app1
Device [connected] (Nerve_MFN_100)
PLC Logic
Application [stop]
GVL
Library Manager
PLC_Program (PRG)
Symbol Configuration
Task Configuration
MainTask
PLC_Program
Profinet_CommunicationTask
PN_Controller.CommCycle
Ethernet_1 (Ethernet)
PN_Controller (PN-Controller)
PLC_Program
Device.Application.PLC_Program

Expression	Type	Value	Prepared value	Address	Comment
TOn_Button	TON				on - delay for reset logic
Counter_Button	CTU				Counter which counts ... number of button p...
tTimeButton	TIME	T#2s			time the button has to be pressed to reset th...
bReset	BOOL	FALSE			Reset ist active

```
(*Reset Logic*)
TOn_Button(INFALSE:= GVL.IN_ButtonFALSE, PT T#0ms :=tTimeButton T#2s );
bResetFALSE := TOn_Button.QFALSE;

(*Counter which counts number of button presses*)
Counter_Button(
    CUFALSE:= GVL.IN_ButtonFALSE,
    RESETFALSE:=TOn_Button.QFALSE ,
```

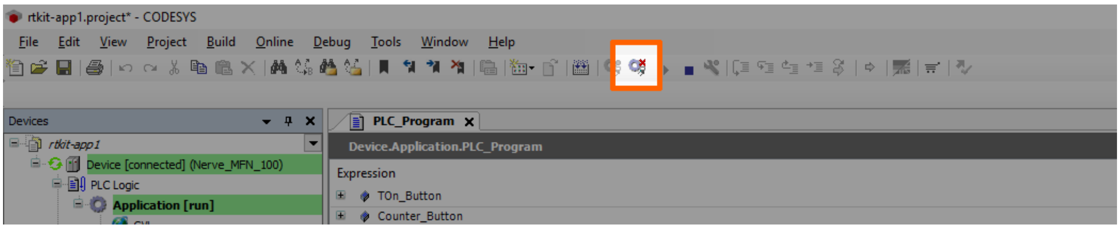
The application is now loaded to the MFN 100.

Downloading an updated application to the MFN 100

Updating an application after loading it into the MFN 100 requires another download into the MFN 100. The download process is slightly different from downloading a new application into the MFN 100.

1. Stop the CODESYS application that has been loaded into the MFN 100 through the Local UI.

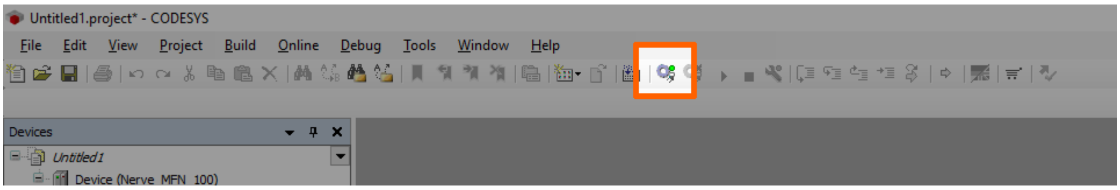
2. Click the **Logout** button in the CODESYS toolbar.


rtkit-app1.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
rtkit-app1
Device [connected] (Nerve_MFN_100)
PLC Logic
Application [run]
PLC_Program
Device.Application.PLC_Program
Expression
TOn_Button
Counter_Button

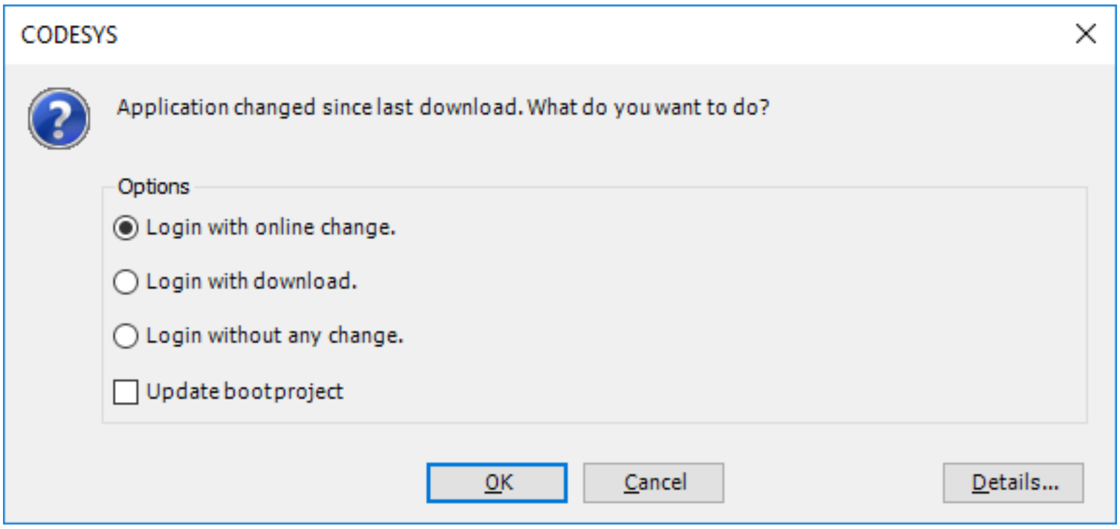
3. Expand **Device (Nerve_MFN_100) > PLC Logic > Application**.

4. Double-click **PLC_Program (PRG)**.
5. Perform the changes.

6. Click the **Login symbol** in the CODESYS menu bar.


Untitled1.project* - CODESYS
File Edit Project Build Online Debug Tools Window Help
Devices
Untitled1
Device (Nerve_MFN_100)

7. In the pop-up window, select one of the options.


CODESYS

Application changed since last download. What do you want to do?

Options

● Login with online change.

○ Login with download.

○ Login without any change.

□ Update bootproject

Item	Description
Login with online change.	The updated application will be loaded into the MFN 100. Variable values will not be reset. If the application was running before, it will be running after the download.
Login with download.	The updated application will be loaded into the MFN 100. Variable values will be reset. The application is stopped.
Login without any change.	The updated application will not be loaded into the MFN 100 but the code will keep the changes.

8. Click **OK**.

The application is now loaded to the MFN 100.

NOTE

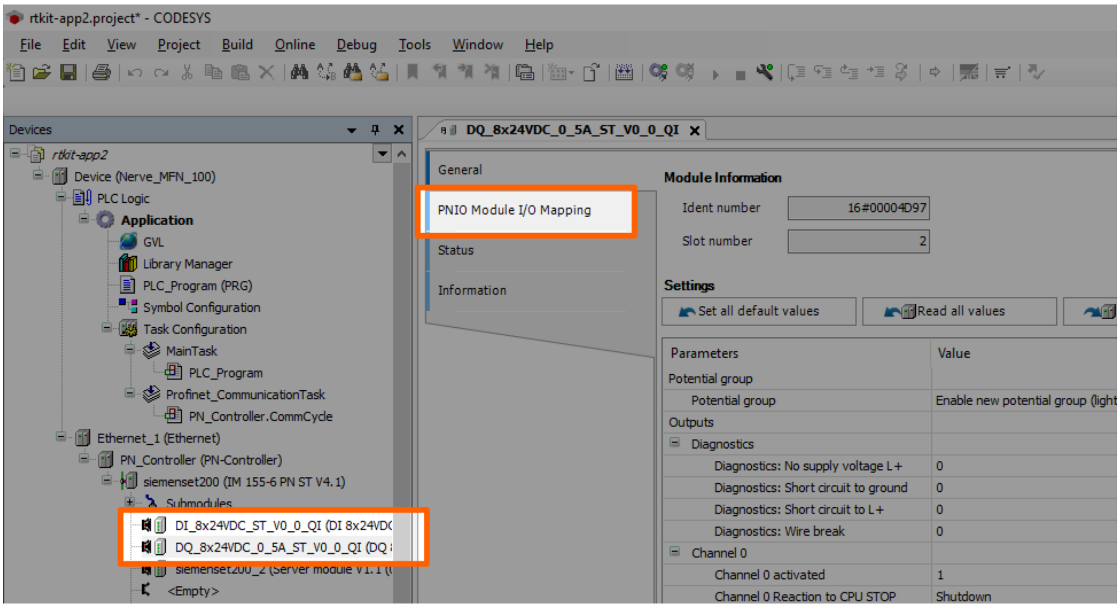
For more help with programming PLC applications in the CODESYS Development System go to help.codesys.com.

Allocating variables to inputs or outputs

After connecting new sensors and actuators, variables need to be assigned to the I/O channel in CODESYS.

1. Open a CODESYS project.
2. Expand **Device (Nerve_MFN_100) > PLC Logic > Ethernet_1 > PN_Controller > siemenset200 (IM 155-6 PN ST V4.1)** in the tree structure on the left.
3. Double-click **DI_8x24VDC_ST_V0_0_QI (...)** for digital inputs.
   Double-click **DQ_8x24VDC_0_5A_ST_V0_0_QI (...)** for digital outputs.

4. Select **PNIO Module I/O Mapping**.


rtkit-app2.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
rtkit-app2
Device (Nerve_MFN_100)
PLC Logic
Application
GVL
Library Manager
PLC_Program (PRG)
Symbol Configuration
Task Configuration
MainTask
PLC_Program
Profinet_CommunicationTask
PN_Controller.CommCycle
Ethernet_1 (Ethernet)
PN_Controller (PN-Controller)
siemenset200 (IM 155-6 PN ST V4.1)
Submodules
DI_8x24VDC_ST_V0_0_QI (DI 8x24VDC
DQ_8x24VDC_0_5A_ST_V0_0_QI (DQ
siemenset200_2 (Server module V1.1 (
<Empty>
DQ_8x24VDC_0_5A_ST_V0_0_QI
General
PNIO Module I/O Mapping
Status
Information
Set all default values
Read all values

Module Information

Ident number: 16#00004D97

Slot number: 2

Settings

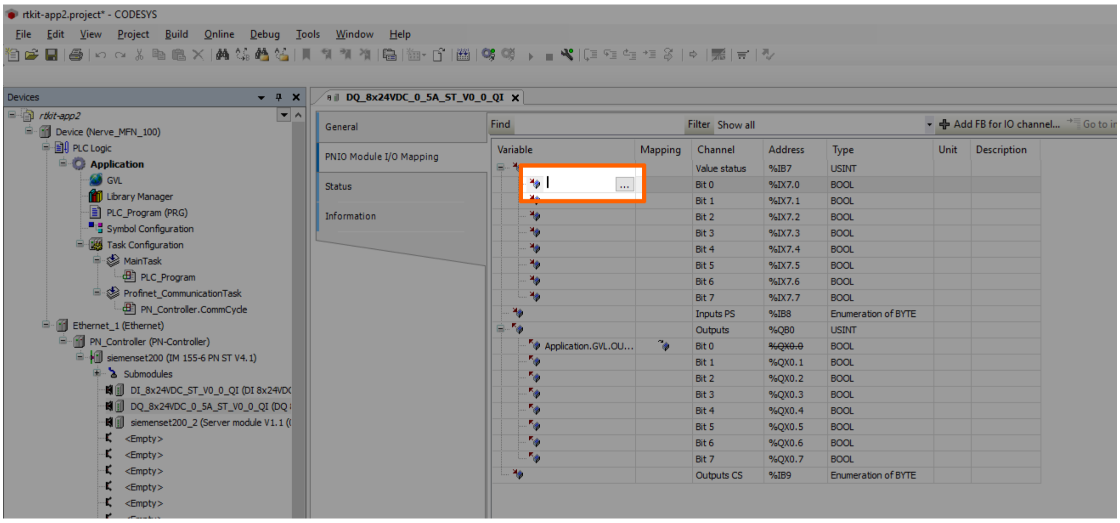
Parameters	Value
Potential group	
Potential group	Enable new potential group (light
Outputs	
Diagnostics	
Diagnostics: No supply voltage L+	0
Diagnostics: Short circuit to ground	0
Diagnostics: Short circuit to L+	0
Diagnostics: Wire break	0
Channel 0	
Channel 0 activated	1
Channel 0 Reaction to CPU STOP	Shutdown

5. Fully expand the tree view.

6. Double-click the variable slot to assign.

> **NOTE**
>
> The inputs in this view do not match the physical inputs of the I/O module on the kit. The inputs here go from 0 to 7. The physical inputs go from 1 to 8. Therefore input 0 in this view represents the physical input 1 on the I/O module. This also applies to outputs.

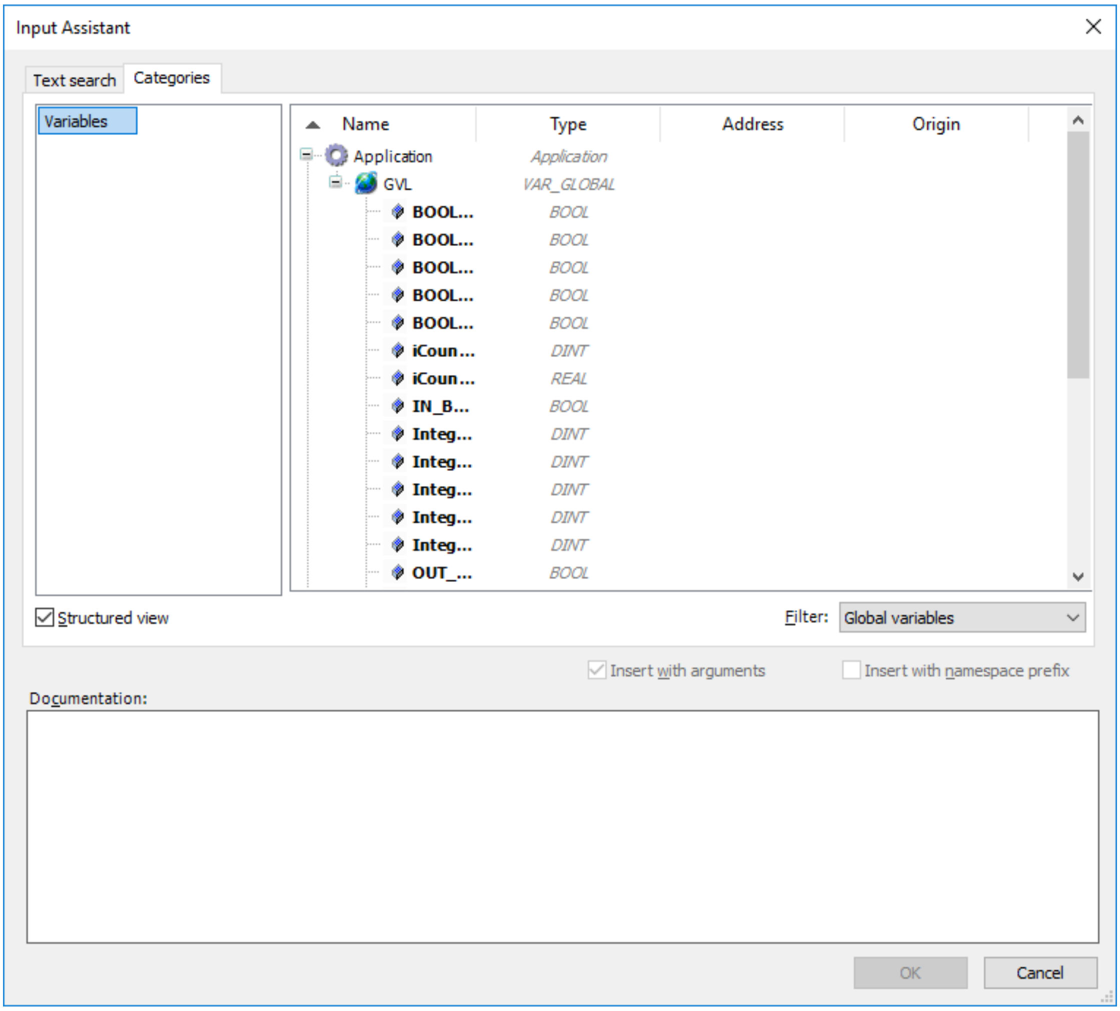
7. Click the three dots next to the variable slot.



Filter: Show all

Variable	Mapping	Channel	Address	Type	Unit	Description
		Value status	%IB7	USINT		
| ...		Bit 0	%IX7.0	BOOL		
		Bit 1	%IX7.1	BOOL		
		Bit 2	%IX7.2	BOOL		
		Bit 3	%IX7.3	BOOL		
		Bit 4	%IX7.4	BOOL		
		Bit 5	%IX7.5	BOOL		
		Bit 6	%IX7.6	BOOL		
		Bit 7	%IX7.7	BOOL		
		Inputs PS	%IB8	Enumeration of BYTE		
		Outputs	%QB0	USINT		
Application.GVL.OU...		Bit 0	~~%QX0.0~~	BOOL		
		Bit 1	%QX0.1	BOOL		
		Bit 2	%QX0.2	BOOL		
		Bit 3	%QX0.3	BOOL		
		Bit 4	%QX0.4	BOOL		
		Bit 5	%QX0.5	BOOL		
		Bit 6	%QX0.6	BOOL		
		Bit 7	%QX0.7	BOOL		
		Outputs CS	%IB9	Enumeration of BYTE		

8. Expand **Application > GVL** in the new window.



Name	Type	Address	Origin
Application	Application		
GVL	VAR_GLOBAL		
BOOL...	BOOL		
BOOL...	BOOL		
BOOL...	BOOL		
BOOL...	BOOL		
BOOL...	BOOL		
iCoun...	DINT		
iCoun...	REAL		
IN_B...	BOOL		
Integ...	DINT		
Integ...	DINT		
Integ...	DINT		
Integ...	DINT		
Integ...	DINT		
OUT_...	BOOL		

☑ Structured view

Filter: Global variables

☑ Insert with arguments

□ Insert with namespace prefix

9. Select the variable to assign.

NOTE

Make sure to select a variable of the same type as the input, i.e., a **BOOL** variable for a **BOOL** input or output.

10. Click **OK**.

Use the assigned variables to read data from connected sensors or to control actuator functionality.

NOTE

For more help with programming PLC applications in the CODESYS Development System go to help.codesys.com.

Data transfer from CODESYS to the Management System

> **NOTE**
>
> The configuration and visualization of new variables with the Nerve Datapath and Grafana is not available in version 2.0 of Nerve Blue. This chapter covers steps and information before the visualization is configured. The Nerve Datapath and Grafana visualization will be available with version 2.1.

The kit sends data that is generated in CODESYS from the MFN 100 to the Management System using variables. The kit has a pre-configured Gateway configuration for the Data Services that can be further configured to send new values to the Management System. This chapter gives a quick overview on how to configure new variables.

Overview

The CODESYS runtime has an integrated OPC UA server which can be configured via the CODESYS Development System. The data from the OPC UA server is received by an OPC UA client, ingested by the Data Services and written into the database in the Management System.

The Data Services are pre-configured to transmit a set of different kinds of CODESYS variables to the Management System. Use the following global variables to transmit data to the Management System:

Global Variables			
BOOL_1	Integer_1	Real_1	STRING_1
BOOL_2	Integer_2	Real_2	STRING_2
BOOL_3	Integer_3	Real_3	STRING_3
BOOL_4	Integer_4	Real_4	STRING_4
BOOL_5	Integer_5	Real_5	STRING_5

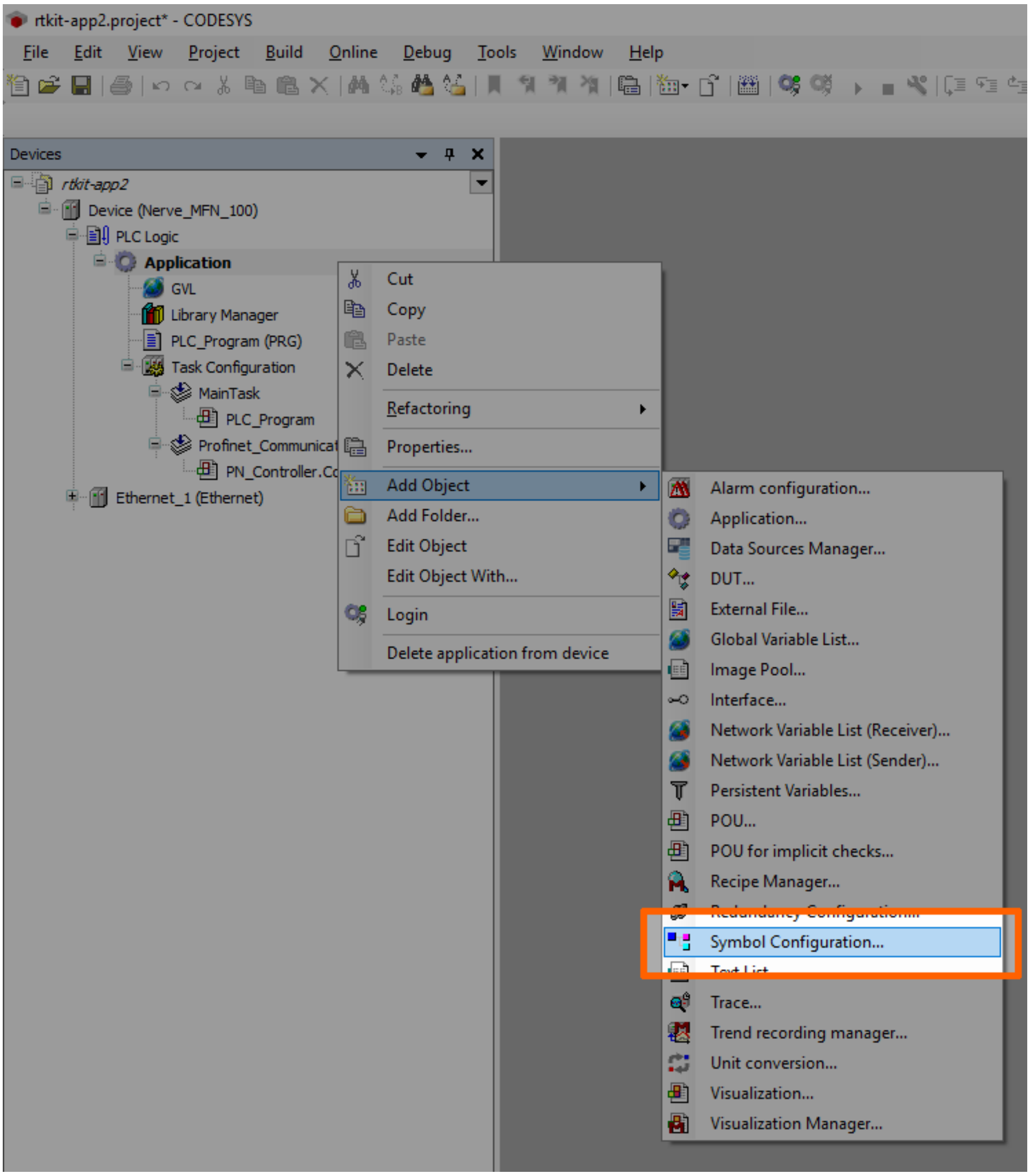
The following instructions are only needed if new variables are added or a completely new CODESYS project is created.

Configuring the CODESYS OPC UA server

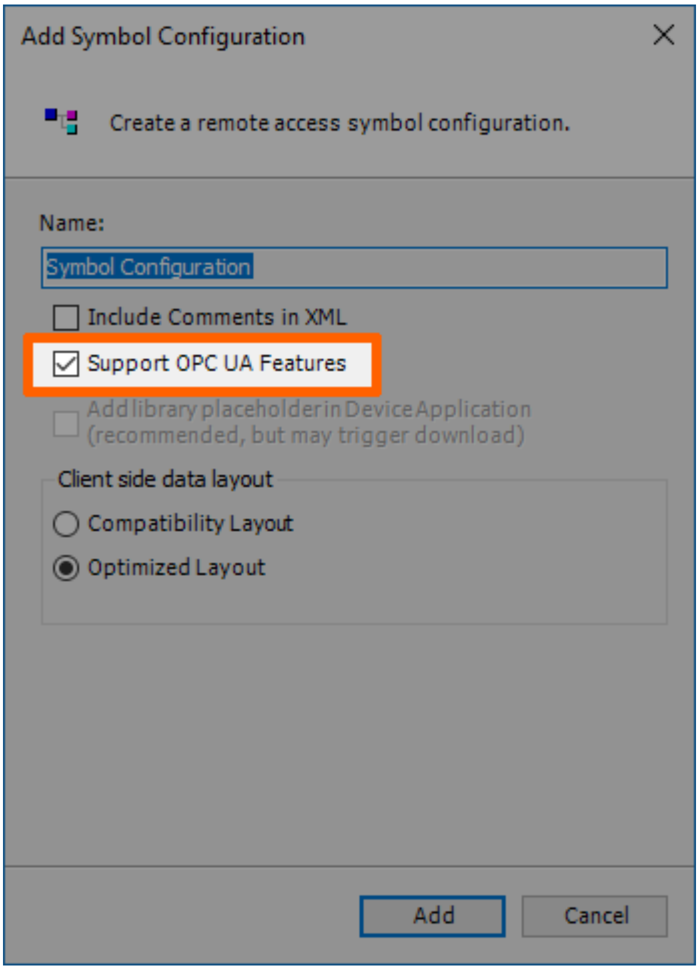
In the CODESYS Development System it is possible to configure which variables are available in the OPC UA server. Configuring variables requires an object called **Symbol Configuration**.

1. Open a CODESYS project to configure variables.
2. Expand **Device (Nerve_MFN_100) > PLC Logic**.
3. Right click **Application**.
4. Select **Add Object**.

5. Click **Symbol Configuration...**



6. Tick **Support OPC UA Features**.



Add Symbol Configuration

Create a remote access symbol configuration.

Name: Symbol Configuration

☐ Include Comments in XML

☑ Support OPC UA Features

☐ Add library placeholder in Device Application (recommended, but may trigger download)

Client side data layout

○ Compatibility Layout

● Optimized Layout

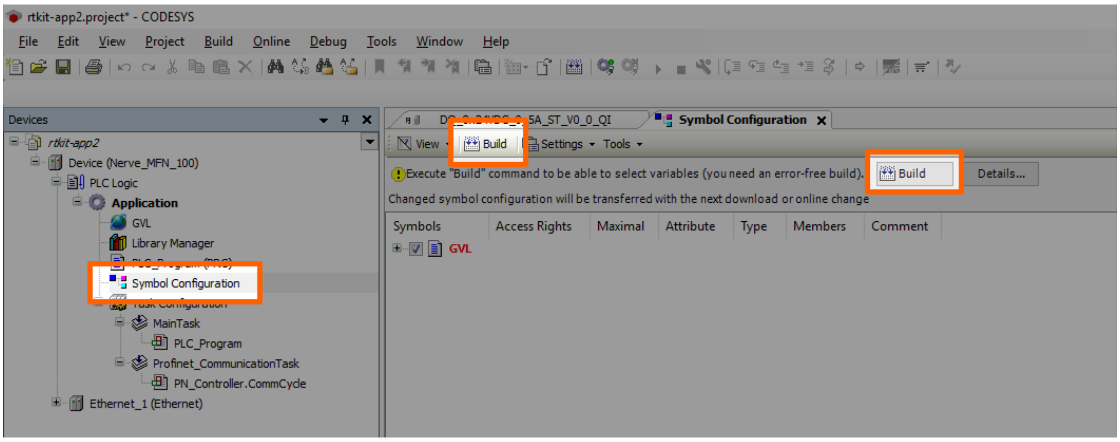
7. Click **Add**.

Adding variables to the OPC UA server

Before adding variables to the OPC UA server, add the object **Symbol Configuration** to the tree structure. Refer to the instructions above on how to add the Symbol Configuration to the tree structure.

1. Open a CODESYS project to configure variables.
2. Expand **Device (Nerve_MFN_100) > PLC Logic > Application**.
3. Double-click **Symbol Configuration**.

4. Click **Build**.

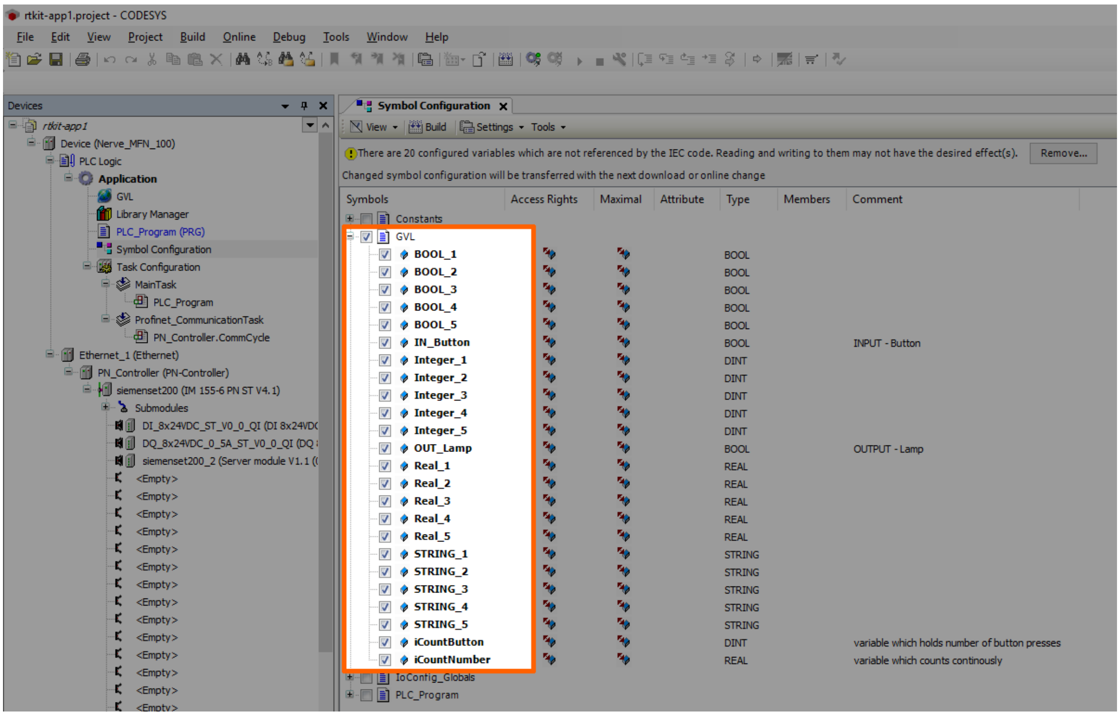


Execute "Build" command to be able to select variables (you need an error-free build).

Changed symbol configuration will be transferred with the next download or online change

5. Expand **GVL**.

6. Tick the variables to add to the OPC UA server.


rtkit-app1.project - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
rtkit-app1
Device (Nerve_MFN_100)
PLC Logic
Application
GVL
Library Manager
PLC_Program (PRG)
Symbol Configuration
Task Configuration
MainTask
PLC_Program
Profinet_CommunicationTask
PN_Controller.CommCycle
Ethernet_1 (Ethernet)
PN_Controller (PN-Controller)
siemenset200 (IM 155-6 PN ST V4.1)
Submodules
DI_8x24VDC_ST_V0_0_QI (DI 8x24VDC
DQ_8x24VDC_0_5A_ST_V0_0_QI (DQ
siemenset200_2 (Server module V1.1 (
<Empty>
Symbol Configuration
View Build Settings Tools
Remove...

There are 20 configured variables which are not referenced by the IEC code. Reading and writing to them may not have the desired effect(s).

Changed symbol configuration will be transferred with the next download or online change

Symbols	Access Rights	Maximal	Attribute	Type	Members	Comment
□ Constants						
☑ GVL						
☑ BOOL_1				BOOL		
☑ BOOL_2				BOOL		
☑ BOOL_3				BOOL		
☑ BOOL_4				BOOL		
☑ BOOL_5				BOOL		
☑ IN_Button				BOOL		INPUT - Button
☑ Integer_1				DINT		
☑ Integer_2				DINT		
☑ Integer_3				DINT		
☑ Integer_4				DINT		
☑ Integer_5				DINT		
☑ OUT_Lamp				BOOL		OUTPUT - Lamp
☑ Real_1				REAL		
☑ Real_2				REAL		
☑ Real_3				REAL		
☑ Real_4				REAL		
☑ Real_5				REAL		
☑ STRING_1				STRING		
☑ STRING_2				STRING		
☑ STRING_3				STRING		
☑ STRING_4				STRING		
☑ STRING_5				STRING		
☑ iCountButton				DINT		variable which holds number of button presses
☑ iCountNumber				REAL		variable which counts continously
□ IoConfig_Globals						
□ PLC_Program						

Adding variables to the Data Services Gateway

The instances of the Nerve Data Services Gateway on the node and in the Management System have a pre-configured Gateway configuration loaded. These configurations need to be adapted when new values are added through sensors, actuators and in the CODESYS Development System.

Local Data Services Gateway on the node

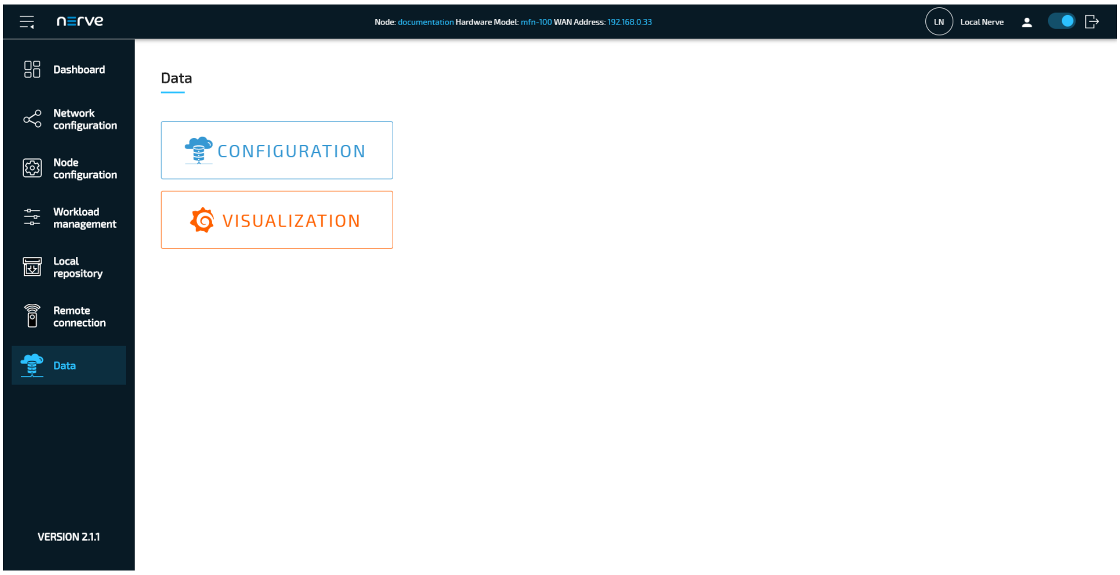
1. Connect the workstation to the console port **P1** of the MFN 100.

2. Configure the network adapter through which the workstation is connected to the MFN 100 the following way:

IP address	`172.20.2.90`
Subnet mask	`255.255.255.0`

Follow this link to reach the Local UI: http://172.20.2.1:3333.
3.
4. Log in with the credentials from the customer profile.
5. Select **Data** in the navigation on the left.

6. Select **CONFIGURATION**.



7. Look for the following part of the currently loaded Gateway configuration:

```
"samplingIntervalAtServer_ms": 500,
 "nodes": [
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.iCountButton",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.iCountNumber",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.BOOL_1",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.BOOL_2",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.BOOL_3",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.BOOL_4",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.BOOL_5",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Integer_1",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Integer_2",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Integer_3",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Integer_4",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Integer_5",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Real_1",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Real_2",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Real_3",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Real_4",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.Real_5",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_1",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_2",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_3",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_4",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_5"
```

8. Edit the last line the following way to add a new variable called `myVar1`:

```
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_2",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_3",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_4",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.STRING_5",
    "ns=4;s=|var|Nerve_MFN_100 .Application.GVL.myVar1"
```

The variable is added as a fully qualified nodeId. Refer to Unified Automation for more information.

> **NOTE**
>
> This NodeId can be found out with an OPC UA Client such as UA Expert. Connect a workstation to **P1** of the MFN 100 and connect to `opc.tcp://172.20.2.2:4840` in the OPC UA Client.

9. Select **Apply** to save the Gateway configuration. The Gateway will restart automatically.

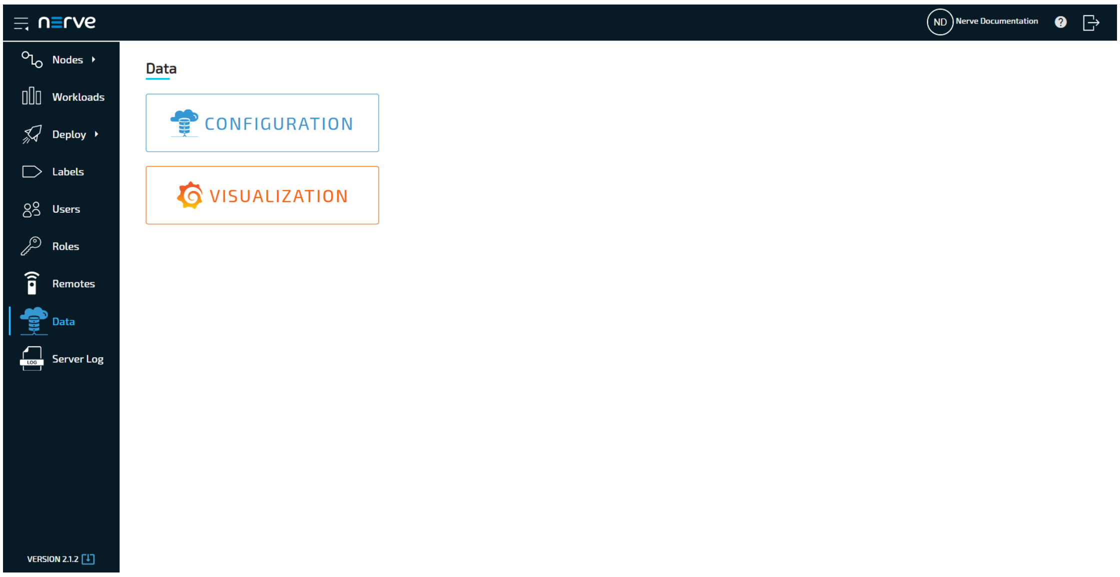
   ![[!Apply Gateway configuration]../img/data_opcua-ms01.png)

The data is automatically stored in the local TimescaleDB and also sent to the Management System. Refer to Nerve Data Services for more information.

Central Data Services Gateway in the Management System

1. Log in to the Management System
2. Select **Data** in the navigation on the left. Make sure that the user has the necessary permissions to access the Data Services.

3. Select **CONFIGURATION** to access the Data Services UI.



4. Look for the following part of the currently loaded Gateway configuration:

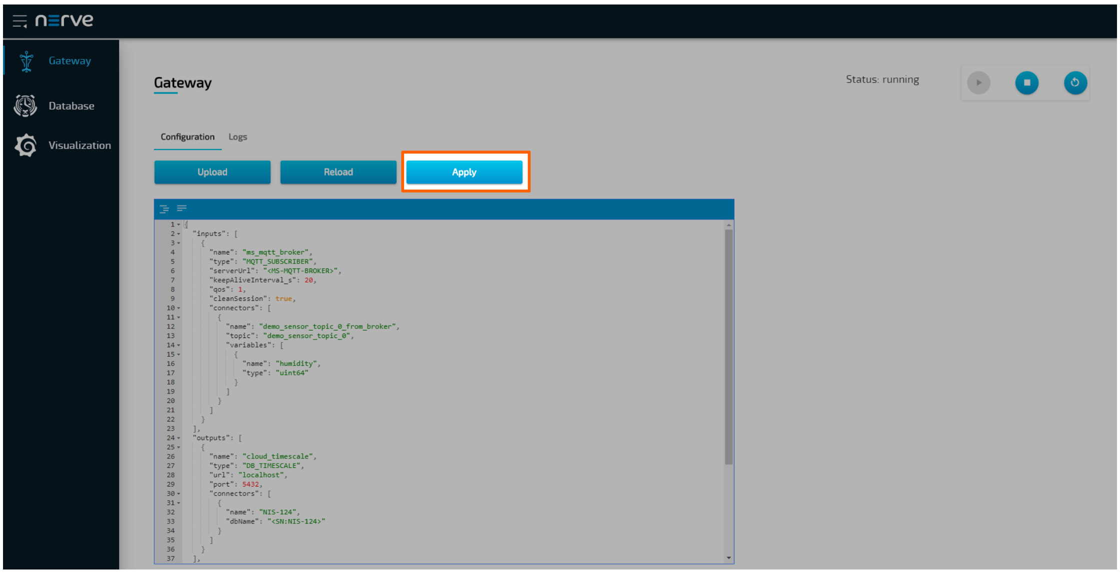
```
      {
   "name":  "STRING_1",
   "type":  "string"
     },
      {
   "name":  "STRING_2",
   "type":  "string"
      },
      {
   "name":  "STRING_3",
   "type":  "string"
      },
    {
"name": "STRING_4",
"type": "string"
    },
    {
"name": "STRING_5",
"type": "string"
    }
```

5. Edit the last line the following way to add a new variable called `myVar1`:

```
    {
"name": "STRING_1",
"type": "string"
  },
    {
"name": "STRING_2",
"type": "string"
    },
    {
"name": "STRING_3",
"type": "string"
    },
    {
"name": "STRING_4",
"type": "string"
    },
    {
"name": "STRING_5",
"type": "string"
    },
    {
"name": "myVar1",
"type": "bool"
    }
```

> **NOTE**
>
> Note that the type of the variable is determined when the new variable is configured in the CODESYS Development System.

6. Select **Apply** to save the Gateway configuration. The Gateway will restart automatically.



```
{
  "inputs": [
    {
      "name": "ms_mqtt_broker",
      "type": "MQTT_SUBSCRIBER",
      "serverUrl": "<MS-MQTT-BROKER>",
      "keepAliveInterval_s": 20,
      "qos": 1,
      "cleanSession": true,
      "connectors": [
        {
          "name": "demo_sensor_topic_0_from_broker",
          "topic": "demo_sensor_topic_0",
          "variables": [
            {
              "name": "humidity",
              "type": "uint64"
            }
          ]
        }
      ]
    }
  ],
  "outputs": [
    {
      "name": "cloud_timescale",
      "type": "DB_TIMESCALE",
      "url": "localhost",
      "port": 5432,
      "connectors": [
        {
          "name": "NIS-124",
          "dbName": "<SN:NIS-124>"
        }
      ]
    }
  ],
```

Visualizing new variables

Newly added variables can also be visualized through the visualization element of the Data Services. The instructions below describe how to add new variables to the existing dashboard by adding a query. New dashboards can also be created for the new variables. Refer to Creating a dashboard for more information.

Adapting the local data visualization on the node

1. Connect the workstation to the console port **P1** of the MFN 100.

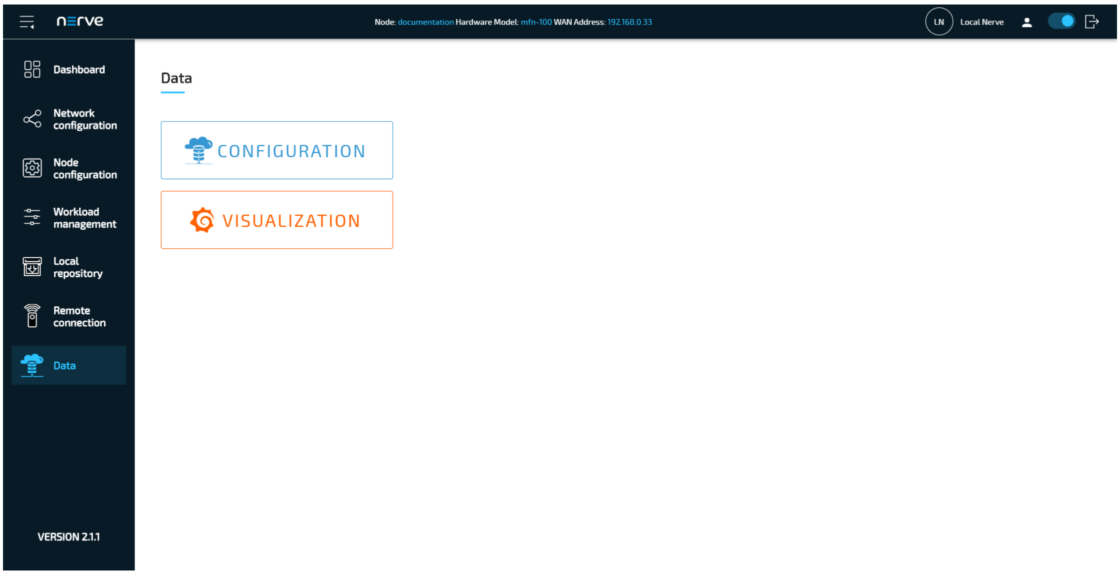
2. Configure the network adapter through which the workstation is connected to the MFN 100 the following way:

IP address	`172.20.2.90`
Subnet mask	`255.255.255.0`

3. Follow this link to reach the Local UI: http://172.20.2.1:3333.

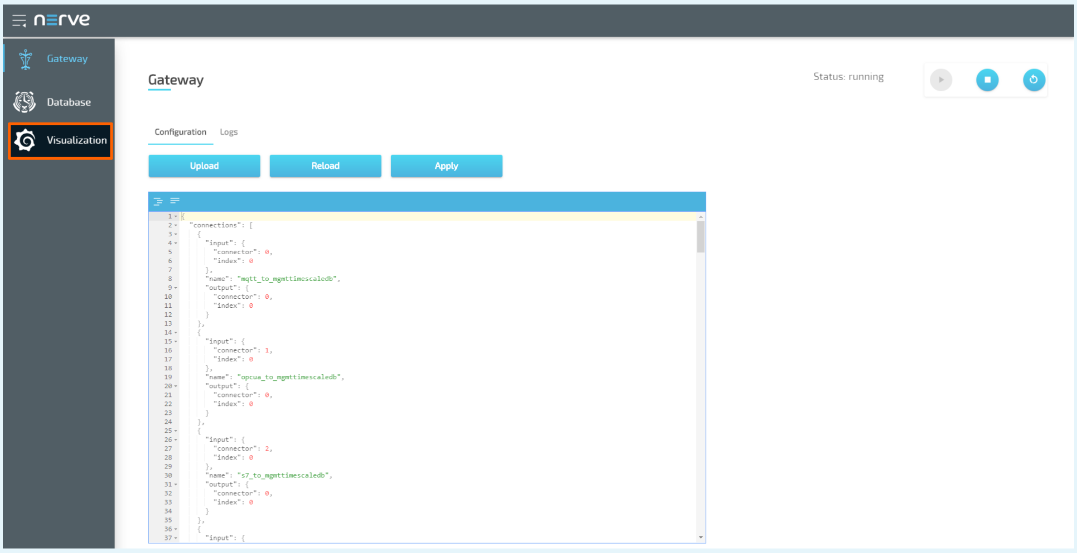
4. Log in with the credentials from the customer profile.
5. Select **Data** in the navigation on the left.

6. Select **VISUALIZATION**.



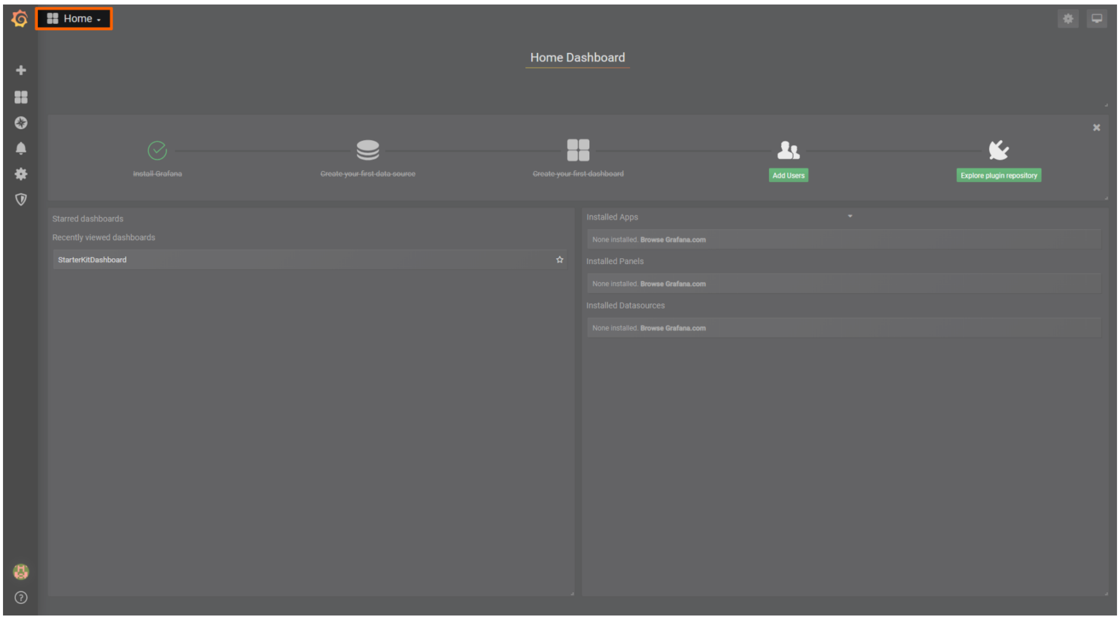
NOTE

The visualization element can also be reached from the Data Services UI. When in the Data Services UI, select **Visualization** in the navigation on the left and select **Open** to reach the Grafana UI.

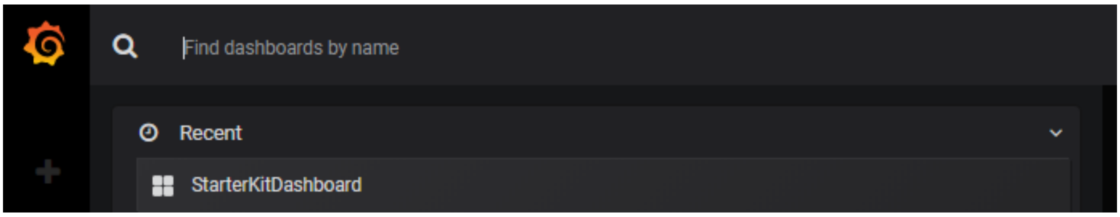


```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

7. Select **Home** in the upper-left corner.


Home
Home Dashboard
~~Install Grafana~~
~~Create your first data source~~
~~Create your first dashboard~~
Add Users
Explore plugin repository
Starred dashboards
Recently viewed dashboards
StarterKitDashboard
Installed Apps
None installed. Browse Grafana.com
Installed Panels
None installed. Browse Grafana.com
Installed Datasources
None installed. Browse Grafana.com

8. Select **Nerve Blue Kit** underneath the search bar.

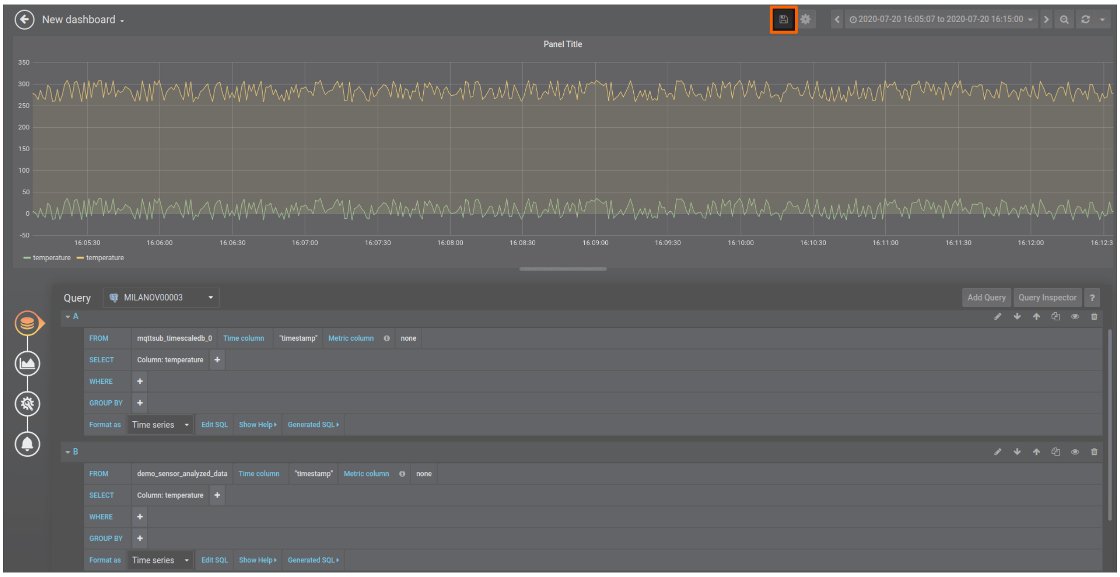


9. Select **Add Query** to the right to add a query for the new variable.

10. Fill in the following query information:

Setting	Value
FROM	`Codesys_to_localdb` **Time column**: "timestamp"
SELECT	Column: myVar1
Format as	Time series

11. Select the save icon in the upper-right corner to save the dashboard.



Query: MILANOV00003

A
FROM mqttsub_timescaledb_0 Time column "timestamp" Metric column none
SELECT Column: temperature +
WHERE +
GROUP BY +
Format as Time series Edit SQL Show Help Generated SQL

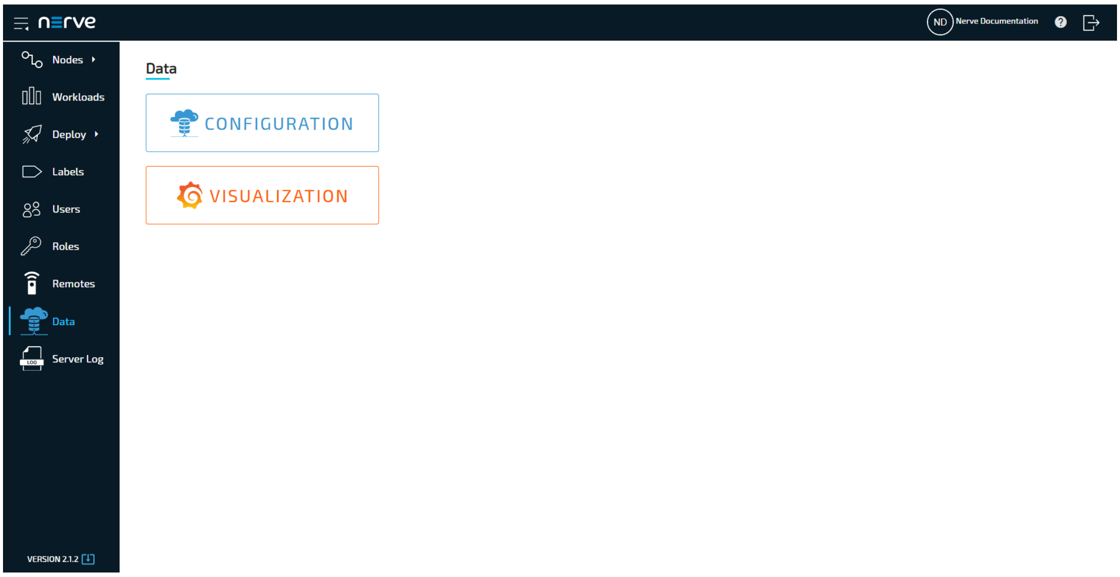
B
FROM demo_sensor_analyzed_data Time column "timestamp" Metric column none
SELECT Column: temperature +
WHERE +
GROUP BY +
Format as Time series Edit SQL Show Help Generated SQL

Adapting the central data visualization in the Management System

1. Log in to the Management System.
2. Select **Data** in the navigation on the left.

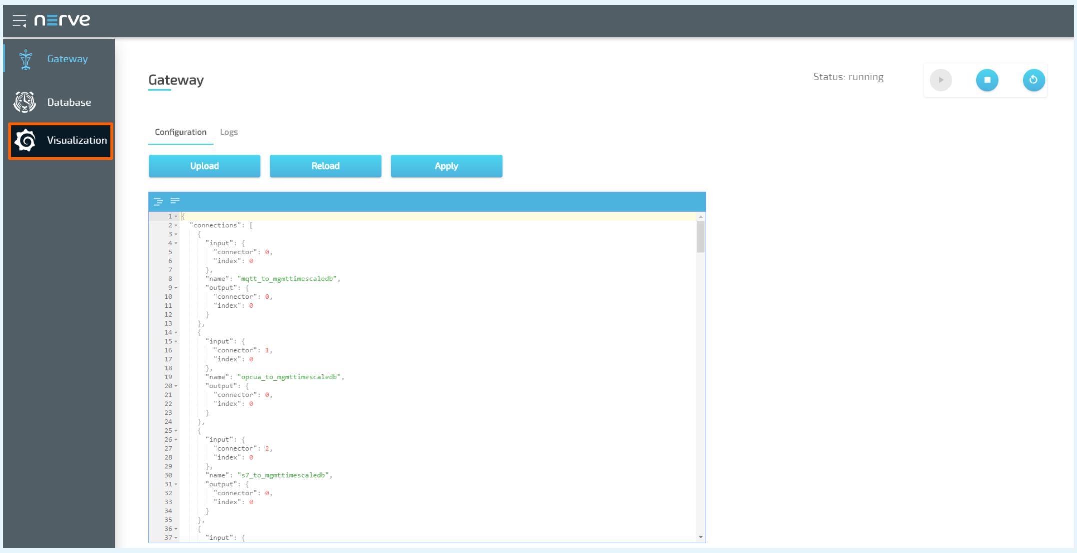
> **NOTE**
>
> If the menu item **Data** is not available, make sure the logged in user has the permission to access the Data Services. Refer to Assigning a role to a user for more information.

3. Select **VISUALIZATION**.



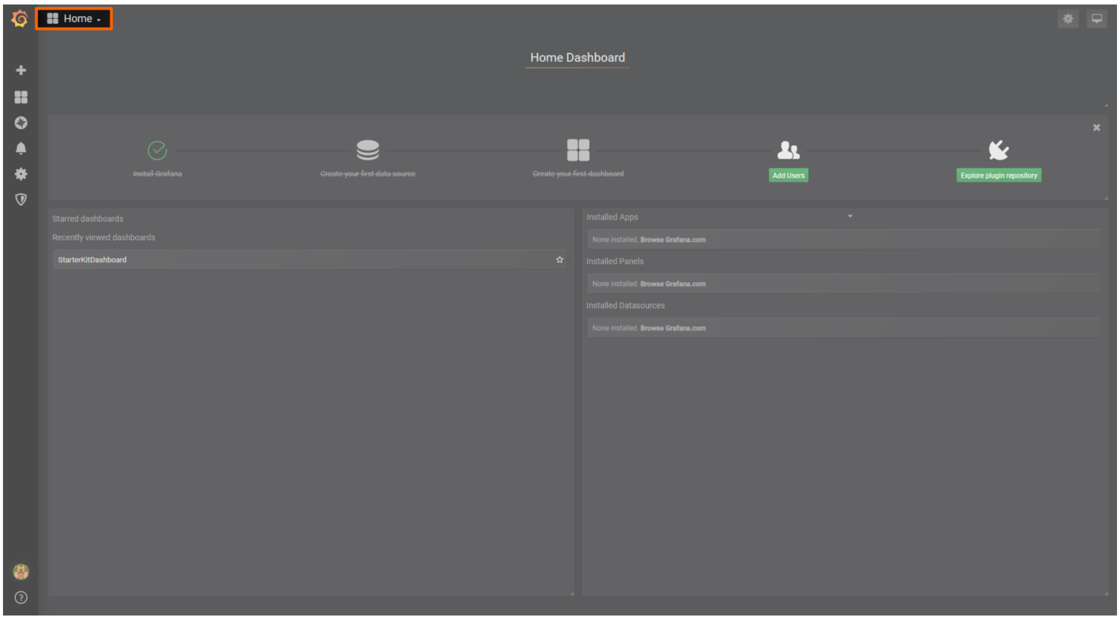
> **NOTE**

The visualization element can also be reached from the Data Services UI. When in the Data Services UI, select **Visualization** in the navigation on the left and select **Open** to reach the Grafana UI.

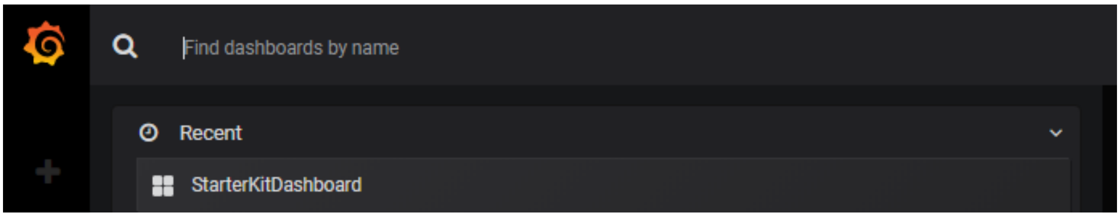


```
{
  "connections": [
    {
      "input": {
        "connector": 0,
        "index": 0
      },
      "name": "mqtt_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 1,
        "index": 0
      },
      "name": "opcua_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
        "connector": 2,
        "index": 0
      },
      "name": "s7_to_mgmttimescaledb",
      "output": {
        "connector": 0,
        "index": 0
      }
    },
    {
      "input": {
```

4. Select **Home** in the upper-left corner.


Home
Home Dashboard
~~Install Grafana~~
~~Create your first data source~~
~~Create your first dashboard~~
Add Users
Explore plugin repository
Starred dashboards
Recently viewed dashboards
StarterKitDashboard
Installed Apps
None installed. Browse Grafana.com
Installed Panels
None installed. Browse Grafana.com
Installed Datasources
None installed. Browse Grafana.com

5. Select **Nerve Blue Kit** underneath the search bar.

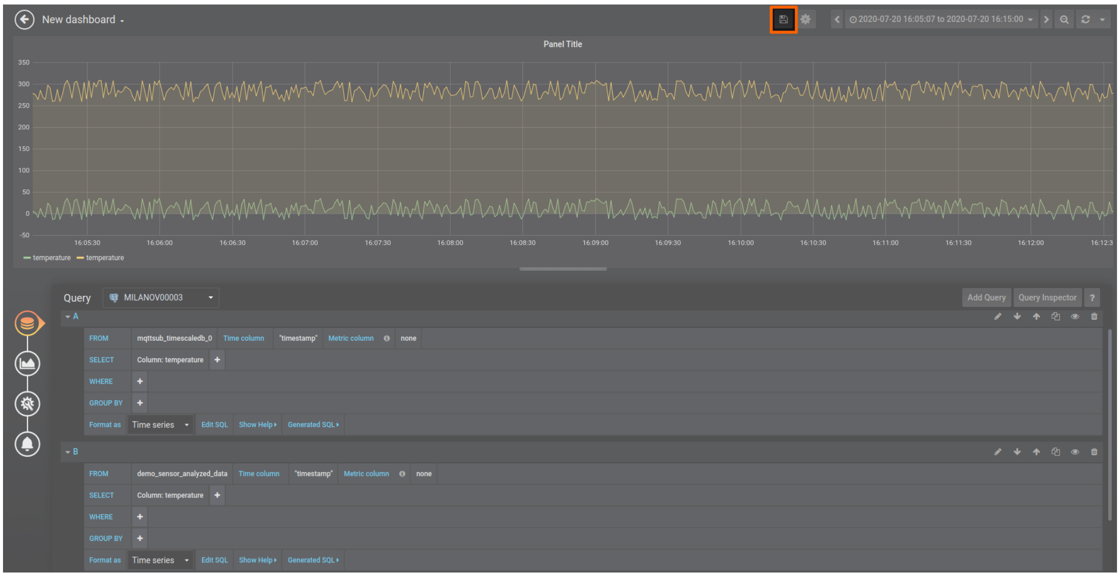


6. Select **Add Query** to the right to add a query for the new variable.

7. Fill in the following query information:

Setting	Value
FROM	`ms_mqtt_broker_to_cloud_timescale_db` **Time column**: "timestamp"
SELECT	Column: myVar1
Format as	Time series

8. Select the save icon in the upper-right corner to save the dashboard.



Query: MILANOV00003

A

FROM	mqttsub_timescaledb_0	Time column	"timestamp"	Metric column	none
SELECT	Column: temperature				
WHERE	+				
GROUP BY	+				
Format as	Time series				

B

FROM	demo_sensor_analyzed_data	Time column	"timestamp"	Metric column	none
SELECT	Column: temperature				
WHERE	+				
GROUP BY	+				
Format as	Time series				

Resources

Previous Versions