

PACEdge™

USER MANUAL



Contents

Section 1: Introduction.....1

1.1	Revision History.....	1
1.2	New Features in PACEEdge 2.2.0	1
1.2.1	Applications Updated to a Newer Version.....	1
1.2.2	Functional Improvements in Node-RED.....	2
1.2.3	Optimizations in Grafana	2
1.2.4	New Features in Telegraf	3
1.2.5	OPC UA Port Forwarding Feature	3
1.2.6	Remote Access Technologies.....	3
1.2.7	Adding new User Role: Service	4
1.2.8	New and Improved Password Management Tool.....	4
1.2.9	Progea's Name was Changed to Movicon	4
1.2.10	New PACEEdge Update Utility	4

Section 2: Physical Connections and Configuration

Settings.....5

2.1	Network, Keyboard, Display	5
2.2	Power	6

Section 3: PACEEdge Getting Started7

3.1	PACEEdge Usage Models	7
3.2	PACEEdge in a Direct-Use Configuration.....	7
3.2.1	Getting Started.....	7
3.3	PACEEdge in Headless Configuration	8
3.3.1	Getting Started.....	8
3.4	Initial Login.....	9
3.5	PACEEdge Users, Rights, and Passwords	11
3.5.1	Changing Passwords via Automated Password Management Utility.....	14
3.5.2	Changing Cockpit/Linux User Passwords.....	17
3.6	PACEEdge System Level Settings.....	18
3.6.1	System Configuration Changes via Cockpit.....	18
3.6.2	Physical – Logical Ethernet Port Mapping	20

3.7	PACEEdge License File	21
3.7.1	Backing Up License Files.....	22
3.7.2	Licensed Connex and WebHMI Features.....	23

Section 4: PACEEdge Architecture Details24

4.1	PACEEdge Applications.....	24
4.1.1	Application Development Components.....	26
4.1.2	Database Components	28
4.1.3	Administration Components	28
4.2	PACEEdge Communications and Data Flow	29
4.2.1	PACEEdge Internal Communication Parameters.....	31
4.2.2	PACEEdge External Communication Parameters	32
4.2.3	Accessing Connex OPC UA Server	35
4.2.4	Accessing Connex OPC UA Server from Movicon.NExT Browser....	35
4.2.5	Accessing OPC UA Server via Port Forwarding	36
4.2.6	PACEEdge Suggested Data Flows.....	37
4.2.7	Southbound Communication Capabilities.....	38
4.2.8	North Bound Communication Capabilities.....	39
4.2.9	PACEEdge Internal Communications and Data Flow	39
4.3	PACEEdge Remote Access.....	40
4.3.1	Remote Access using MeshCentral.....	41
4.3.2	Steps to setup MeshCentral: Create MeshCentral Account	42
4.3.3	Remote Access using ZeroTier	53
4.3.4	Remote Access using WireGuard	57
4.3.5	Remote Access using OpenVPN	59

Section 5: PACEEdge Usage Examples.....60

5.1	Node-RED Getting Started	60
5.1.1	Example Description	60
5.1.2	Prerequisites	60
5.1.3	Example Walkthrough	60
5.2	Using the InfluxDB Database	65
5.2.1	Using Emerson Customized InfluxDB Nodes.....	65
5.2.2	Using Node-RED Standard InfluxDB Nodes	70
5.3	Using MySQL Database	73

5.3.1	Using Emerson Customized MySQL Nodes	74
5.3.2	Using Node-RED Standard MySQL Nodes	77
5.4	Using MQTT Communication	80
5.4.1	Using Emerson Customized MQTT Nodes.....	80
5.4.2	Using Node-RED Standard MQTT Nodes	82
5.5	Using Grafana.....	85
5.5.1	Example Description	86
5.5.2	Prerequisites	86
5.5.3	Example Walk Through.....	86
5.6	Using Connex and WebHMI.....	96
5.6.1	Prerequisites	96
5.6.2	Example Walk Through.....	96
5.7	Configuring Telegraf to Log Data into InfluxDB.....	112
5.7.1	Example Description	112
5.7.2	Prerequisites	112
5.7.3	Example Walk Through.....	112

Section 6: PACEEdge Performance Indication117

6.1	OPC UA Performance Indication.....	117
6.2	Node-RED Dashboard Performance Indication.....	119
6.3	Node-RED Modbus TCP Performance Indication	121
6.4	Node-RED Connex Historian Performance Indication	121

Section 7: Saving and Restoring User Data122

7.1	Exporting and Importing Flows in Node-RED	122
7.1.1	Export Flow.....	122
7.1.2	Import Flow	123
7.2	Exporting and Importing Views in Grafana.....	123
7.2.1	Export Flow.....	123
7.2.2	Import Flow	123
7.3	Importing Projects in Connex/WebHMI	124
7.4	Backing Up and Restoring InfluxDB and MySQL Databases	124
7.5	Saving License Files.....	125

Section 8: PACEEdge Software Backup/Recovery126

8.1	PACEEdge Software Backup on RXi2-BP, RXi2-LP IPCs	127
8.2	PACEEdge Software Restore on RXi2-BP, RXi2-LP IPCs.....	129

8.3	PACEEdge Software Backup on CPL410, CPE400 Controllers.....	131
8.4	PACEEdge Software Restore on CPL410, CPE400 Controllers	133
8.4.1	CPL410, CPE400 PACEEdge Recovery to Factory Default	133
8.4.2	CPL410, CPE400 Restore of PACEEdge Backup Image.....	133

Section 9: Upgrade from Earlier PACEEdge Versions.....136

9.1	Upgrading RXi2-BP, RXi2-LP from PACEEdge 2.1.1 to 2.2.0	136
9.1.1	PACEEdge Upgrade using Navigator.....	137
9.1.2	PACEEdge Upgrade using USB Stick.....	144
9.2	Upgrading Older RXi2-BP, RXi2-LP from PACEEdge 2.0 to 2.1.1	145
9.2.1	Upgrade PACEEdge v2.0 to PACEEdge 2.1 (old version).....	146
9.2.2	Apply Patch to PACEEdge v2.1 to get v2.1.1 (old version)	149
9.3	Upgrading CPL410, CPE400 from PACEEdge 2.1.1 to 2.2.0.....	150
9.3.1	PACEEdge Upgrade using Navigator.....	150
9.3.2	PACEEdge Upgrade using USB Stick.....	150
9.4	Upgrading CPL410, and CPE400 with Older PACEEdge to PACEEdge 2.1.1 .151	
9.4.1	Upgrade PACEEdge v1 to PACEEdge 2.1	151

Section 10: Software Utilities and Troubleshooting154

10.1	Mounting USB Stick	154
10.2	Un-Mounting USB Stick.....	157
10.3	Difficulties Accessing PACEEdge Components, Erratic Behavior	158
10.4	PACEEdge Files.....	159
10.4.1	Installation Test	159
10.5	Docker Commands	161
10.5.1	docker-compose up -d.....	162
10.5.2	docker-compose down	162
10.6	./config-compose.sh	162

General Contact Information163

Technical Support163

Warnings and Caution Notes as Used in this Publication

WARNING

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

CAUTION

Caution notices are used where equipment might be damaged if care is not taken.

Note: Notes merely call attention to information that is especially significant to understanding and operating the equipment.

These instructions do not purport to cover all details or variations in equipment nor to provide for every possible contingency to be met during installation, operation, and maintenance. The information is supplied for informational purposes only, and Emerson makes no warranty on the accuracy of the information included herein. Changes, modifications, and/or improvements to equipment and specifications are made periodically, and these changes may or may not be reflected herein. It is understood that Emerson may make changes, modifications, or improvements to the equipment referenced herein or to the document itself at any time. This document is intended for trained personnel familiar with the Emerson products referenced herein.

Emerson may have patents or pending patent applications covering the subject matter in this document. The furnishing of this document does not provide any license whatsoever to any of these patents.

Emerson provides the following document and the information included therein as-is and without warranty of any kind, expressed or implied, including but not limited to any implied, statutory warranty of merchantability or fitness for a particular purpose.

Section 1: Introduction

PACEEdge processing maximizes the value of your data by improving operational reliability, safety, and energy consumption. PACEEdge provides all aspects of edge processing and simplifies your IIoT application development, deployment, and administration. All components necessary in the IIoT application lifecycle are combined in one package to provide a unified interface to decrease your development time and increase your deployable footprint.

1.1 Revision History

Rev	Date	Description
F	Mar 2023	Added cautions warning against the use of dollar signs (\$) in passwords under Section 3.4 Initial Login.
E	Nov 2022	Updates related to the PACEEdge v2.2 release
D	Dec 2021	Added details on Accessing the Connex OPC UA Server (Section 0)
C	Sep 2021	PACEEdge 2.1 Initial release
B	Nov 2020	Added Section 7.2 Node-RED Dashboard Performance Indication
A	Sep 2020	PACEEdge 2.0 Initial release

1.2 New Features in PACEEdge 2.2.0

1.2.1 Applications Updated to a Newer Version

In PACEEdge v2.2.0, all applications have been updated to a newer version. GFK-3198, PACEEdge IPI lists the exact versions of each application, but the most important updates are as follows:

- Added support for Movicon 4.2
- Node-RED has been updated to version 2.2.2. This includes functional updates, look and feel improvements, and the ability to create the Subflows.
- Grafana® has been updated to version 9.0.2.

Telegraf® has been updated to version 1.23.2. The most important feature that this version adds is the ability to parse MQTT Topic and Payload to extract Measurement name, uPTagsRre, and Fields to be stored in InfluxDB®.

1.2.2 Functional Improvements in Node-RED

- Updated examples that are accessible via Import->Examples->node-red-contrib-emerson, which now include PacedgeInfluxDB, PacedgeMQTT, and PacedgeMySQL examples. These examples will also install PACEEdge customized Nodes in the palette PACEEdge category, which is preconfigured to access PACEEdge MQTT brokers, InfluxDB “data” database, and MySQL “data” database. Using these customized nodes user only needs to provide the user name and password, but all other access parameters are already pre-configured. Detailed description of how to use these examples flows and nodes is in **Section 5: PACEEdge Usage Examples.**
Note: To have Emerson customized nodes added to the Node-RED node palette; import the corresponding example flow, as described in sections: 0, 5.3, 5.4
- New Nodes in the palette:
 - Emerson pre-configured nodes to access InfluxDB, MySQL, and MQTT
 - Media node for Dashboard, allowing to add jpg or video files to Node-RED Dashboards easily
 - The SVG node for the Dashboard shows vector graphics drawings (SVG) inside the dashboard, including animation.

1.2.3 Optimizations in Grafana

Grafana improvements were focused on streamlining the getting started process so that users can see real data as quickly as possible without configuring databases and dashboards.

For details, please see section 5.5, *Using Grafana*.

- Grafana now has pre-configured PACEEdge default databases:
 - EMR_InfluxDB: InfluxDB database named **data**. Used by Node-RED examples and PACEEdge pre-configured InfluxDB nodes
 - EMR_InfluxDB_Telegraf: InfluxDB database named **telegraf_metrics**, used by Telegraf to store PACEEdge hardware and OS performance and health data, as well as by Telegraf MQTT parser, which allows to automatically parse and store data published on specified topics. More about this feature in the Telegraf section
 - EMR_MariaDB_SQL: MySQL database named **data** and used by Node-RED examples and PACEEdge pre-configured MySQL nodes.
- Grafana has a pre-installed community dashboard to visualize hardware and OS performance data stored in the InfluxDB database. This dashboard starts by default when Grafana is opened the very first time.

1.2.4 New Features in Telegraf

Telegraf is an agent that can gather data from different sources and store it in the InfluxDB database. PACEdge 2.2.0 uses Telegraf for two functions:

- Gathering PACEdge system statistics and health data, such as CPU, memory, network, disc usage, interrupt and process statistics and storing them in InfluxDB database named **telegraf_metrics**, so that pre-installed dashboard in Grafana can be used to visualize and analyze it. Thresholds and alarms can be configured, and notifications to the user can be sent.
- Connecting to PACEdge internal MQTT broker and subscribing to topics starting with **emr_v1/**. Messages with these topics will be parsed and extracted. Measurement names, Tags, and Fields are stored in the InfluxDB **telegraf_metrics** database. An example later in this document shows how data tags in Connex can now be automatically stored in InfluxDB by simply publishing them via MQTT.

1.2.5 OPC UA Port Forwarding Feature

With older PACEdge versions, when using Movicon.NExT Editor to browse and configure OPC UA variables, the user had to ensure that the PC with Movicon Editor has direct access to the network where the PLC OPC UA server is connected. In the case of CPL410, a second Ethernet cable had to be plugged into the LAN1 port, which made a remote use case very difficult. The engineer had to get close to PLC and plug in the cable. The OPC UA port forwarding feature now allows users to forward OPC UA ports within the PACEdge device so that the user's PC gets access to the remote OPC UA Server without needing to leave your desk.

For detailed usage example, please see: 5.6 Using Connex and WebHMI

1.2.6 Remote Access Technologies

PACEdge v2.2.0 comes prepared for remote access via one of the following technologies:

- MechCentral
- ZeroTier
- WireGuard
- OpenVPN

Remote access tools enable accessing PACEdge devices remotely from the internet and help to safely cross the firewall without needing to open specific ports.

Due to cyber security, these remote access mechanisms are disabled by default, but instructions on how to enable them are later in this document 4.3 PACEdge Remote Access.

1.2.7 Adding new User Role: Service

New user role Service has been added to existing roles: admin, developer, and operators.

1.2.8 New and Improved Password Management Tool

The password management tool has been redesigned to streamline usage and data entries. It covers all PACEEdge applications, including Telegraf, Portainer, and Movicon Deploy. It minimizes the number of times that the admin password needs to be entered.

1.2.9 Progea's Name was Changed to Movicon

Progea's name was changed to Movicon, which includes docker container names and browser paths. So now, when deploying a project from within Movicon.NExT Editor, one has to use **ip_address/movicondeploy** instead of **ip_address/progeadeploy**.

1.2.10 New PACEEdge Update Utility

Starting with PACEEdge version 2.2.0, it will be possible to upgrade to the older PACEEdge version without manually restoring user data after the update. Existing Node-RED flows, Grafana dashboards, and data in InfluxDB and MySQL databases will be preserved and accessible after the upgrade. Furthermore, the upgrade utility will allow the user to also make a backup of this data.

As we advance, PACEEdge updates will be split into the host operating system and the PACEEdge Application update. This way user can choose to keep his PACEEdge Applications as is, without touching them, but still update the host Linux OS to get the latest security upgrades and patches.

Section 2: Physical Connections and Configuration Settings

2.1 Network, Keyboard, Display

PACEEdge is designed to operate on a headless computing device using web interfaces; however, if desired, on RXi2-BP and RXi2-LP Industrial PCs, a monitor and keyboard can be attached.

The user must know the IP address assigned to the Ethernet port to use web interfaces. The IP address assignment can be easily checked using a keyboard and local display using Linux commands. Otherwise following options are available:

PACEEdge Factory Installation			
Static IP on 1 st Ethernet port: 192.168.3.100			DHCP assigned IP
RXi2-LP: LAN port next to the USB port	RXi2-BP: ETH0 port	CPL410/CPE400: ETH port	RXi2-LP, RXi2-BP: on any Ethernet port CPL410/CPE400: ETH port

1. Follow the Ethernet Port and IP address guidance above.
2. Attach the Ethernet cable to an appropriate Ethernet port.

Optionally (RXi2-LP, RXi2-BP), attach a USB keyboard to any USB ports.

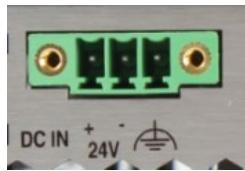
Optionally (RXi2-LP, RXi2-BP), attach a display to the DP++ port.

Note: DisplayPort (DP++) connection is well suited for conversion to HDMI and VGA.
(Conversion adapters are not included).

2.2 Power

1. Connect a 24V +/-25% power to the power connector and power up the unit (Figure 1).
2. The corresponding plug must be a Phoenix Contact "1748367" or equivalent. RXi2-LP, RXi2-BP, CPL410, and CPE400 can be powered with a DC power supply with 24V (+-25%) and at least delivering 2 A.

Figure 1: 24 V DC-IN



Signal Name	Pin (left to right)
Power + (24V DC)	1
Power - (24V DC)	2
FGND	3

Section 3: PACEdge Getting Started

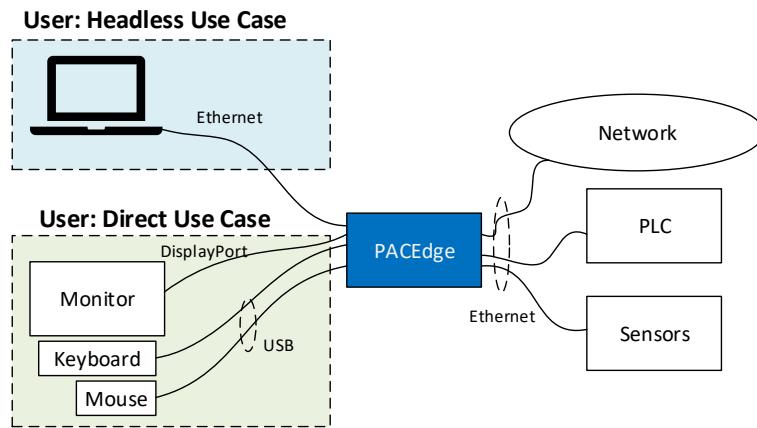
Note: PACEdge software comes pre-installed on Emerson Industrial PCs and Controllers.

3.1 PACEdge Usage Models

PACEdge software has two usage models:

1. **Direct Use Model:** Running on the Industrial PC., which has a directly attached Monitor, keyboard, and mouse.
Note: this mode is not supported on CPL410 and CPE400 Controllers
2. **Headless Use Model:** Running on the Industrial PC, which is operating headless mode, the user accesses it remotely via Ethernet using a web interface.

Figure 2: Use Case Models



Note: for configuration, administrative tasks, or file transfer user can also use an SSH Client to access PACEdge. The same Linux (Cockpit) login credentials apply.

3.2 PACEdge in a Direct-Use Configuration

3.2.1 Getting Started

1. Connect the monitor to the RXi2-BP or RXi2-LP device using a DisplayPort cable.
Note: If the monitor of choice has an HDMI or VGA input, use a standard off-the-shelf DP-HDMI or DP-VGA adapter.
2. Connect a keyboard and mouse to any of the USB ports.
3. Power up the PACEdge unit and wait until it boots.

4. The boot process will pause and ask for login details. Login as **admin** with password **edgestack**.
Note: The user will be asked to change the default password to a unique password at the first login.
5. Most interactions with PACEdge are done via a browser-based interface. Click on Activities->Show Applications and start the Firefox browser to get started.
6. Within the Firefox browser, go to <https://localhost>
7. Proceed to the section entitled 3.4, *Initial Login*.

3.3 PACEdge in Headless Configuration

In a Remote Headless configuration, the user interfaces with PACEdge via Ethernet using a separate device's web browser.

3.3.1 Getting Started

1. Connect the Ethernet cable to the Ethernet port, depending on the hardware, labeled as follows:
 - a. RXi2-BP: ETH0
 - b. **RXi2-LP:** LAN (one next to COM port)
 - c. CPL410/CPE400: ETH
2. Setup the User PC Ethernet port IP address to be statically assigned as follows:
 - a. **IPv4 Static IP: 192.168.3.10**
(or similar in the same subnet)
 - b. Netmask: 255.255.255.0
3. Power up the PACEdge unit and wait until it boots.
Note: on CPL410/CPE400 wait until the GPOK LED is lit.
4. Open the browser of your choice and type in **192.168.3.100** in the address bar.

Note: All Ethernet ports are configured to get IP addresses assigned by the DHCP server. This dynamically assigned IP address can also be used to access PACEdge. On CPL410 and CPE400 first two IP addresses can be read from a built-in display by going into **Edge Settings->Network Config**.

5. Proceed to Section 3.4, Initial Login.

3.4 Initial Login

A warning message will display when the user connects to PACEEdge for the first time (using the instructions outlined in PACEEdge in a Direct-Use Configuration or PACEEdge in Headless Configuration) due to the self-signed certificate on the unit. The message will state that the device's identity cannot be confirmed. Please click on **Advanced** and **Accept** to proceed.

Please read and accept the Emerson End User License Agreement (EULA).

Once EULA has been accepted, you will be redirected to the Password Management screen. PACEEdge software consists of multiple tools and applications, each with user management and password settings. To simplify user and password management tasks, PACEEdge comes with pre-configured users: *admin*, *developer*, service, and *operators*, each having its password. Passwords need to be changed in two steps (Figure 3):

CAUTION

Do not use the dollar sign "\$" in a password. The dollar sign "\$" special character is incorrectly labeled as a supported character. Using "\$" can result in PACEEdge applications being inaccessible and unable to reach Grafana or Telegraf.

1. Use the Password Management page to change passwords for all PACEEdge applications.
2. Use the link to Cockpit to change passwords for Cockpit/Linux.

Once logging into Cockpit for the first time, it will automatically force the user to change the *admin* password. Once changed, please log out and log back into Cockpit as the *admin* user, go to the Accounts page and change passwords for the remaining users: *developer*, *service*, and *operators*.

Note: In Cockpit, do not set any passwords for the user *root*. For security purposes, this user is disabled, and setting a password for it would enable it, which is not recommended. Please consult the PACEEdge Secure Deployment Guide (GFK-3197) for more details.

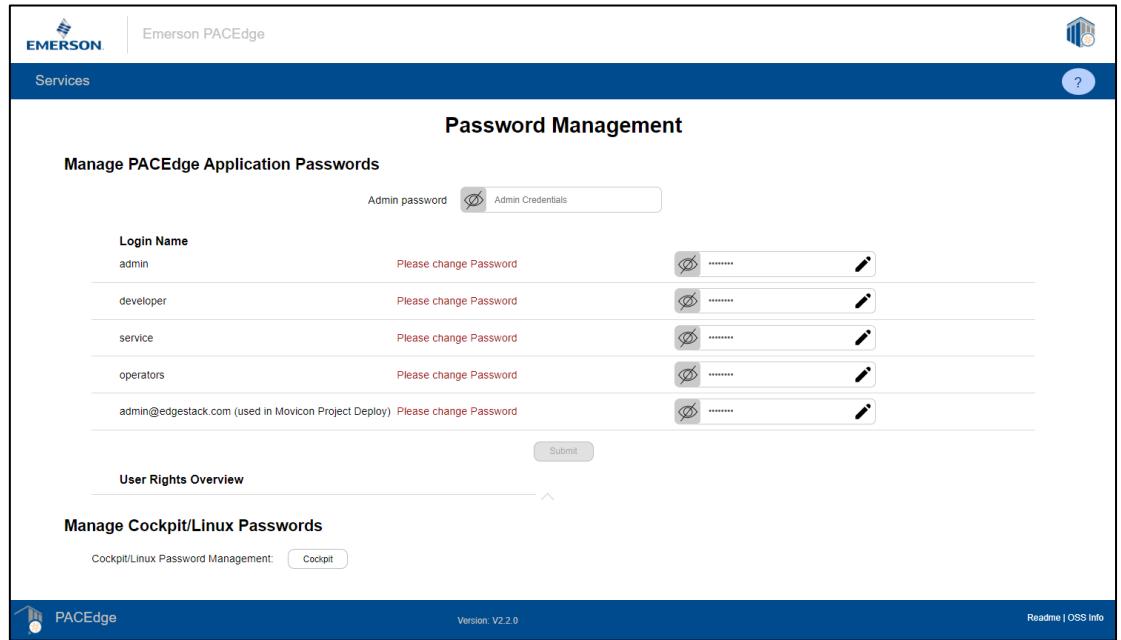
Note: When you click the Apply Changes button and then follow the Restart button, web services used for this connection will be restarted. As a result, depending on the browser used, you might get a message such as "This site can't be reached" or similar. This is normal behavior; manually enter https://ip_address/pw/ in the browser address field to return to the Password Management page. You might need to refresh your browser to see the updated password state. The restart will take up to 60 seconds.

Note: For cybersecurity reasons and proper account setup, the user must change the default passwords at the first login.

Important

Please consult the PACEEdge Secure Deployment Guide (GFK-3197) for recommended password changes and other cybersecurity-relevant settings. Detailed password change procedures can be found in 3.5, *PACEEdge Users, Rights, and Passwords*.

Figure 3: Password Management Screen (Upper portion to change the password for all PACEEdge Applications, a lower portion for Cockpit/Linux password change)

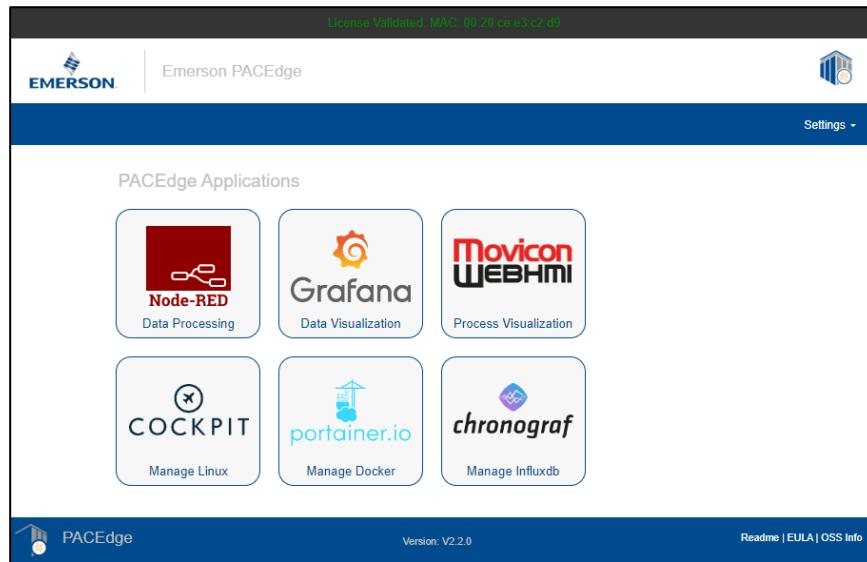


Note: A comment in red *Please change Password* will disappear once the password has been changed; however, it may require the user to refresh the browser.

For more detailed information about user roles, passwords, and the password management utility, please refer to Section 3.5, *PACEEdge Users, Rights, and Passwords*.

Once the passwords have been changed, click on the Emerson logo in the upper-left corner. The logo is a shortcut to the PACEEdge Landing Page, which has different shortcuts to Node-RED, Grafana, Cockpit, Portainer, and other applications. The user can return to the Password Management screen from the PACEEdge landing page by clicking **Password Management** in the Settings drop-down list on the right side.

Figure 4: PACEEdge Landing Page



Note: A link to Movicon WebHMI is only active when an actual WebHMI project has been deployed on the PACEEdge using Movicon.NExT Editor. When starting a new PACEEdge unit, no WebHMI project is deployed; therefore, this shortcut is greyed out.

To explore PACEEdge, click on Node-RED and log in with the following:

- **user:** admin
- **password:** (the password that was set on the Password Management page)

Start exploring example flows or create your own flows. Consult the Usage Example section in this document for details.

Important: Please consult the PACEEdge Secure Deployment Guide (GFK-3197) for recommended password changes and other cybersecurity-relevant settings.

3.5 PACEEdge Users, Rights, and Passwords

In PACEEdge user roles and associated passwords are split into two groups:

- Linux/Cockpit users and passwords
- PACEEdge Application users and passwords.
(Password Management Utility, Node-RED, Grafana, Portainer, InfluxDB, MySQL, Chronograf)

Passwords for these two groups are independent of each other and have to be individually changed for each group.

The passwords of PACEEdge shall be changed at the first login. Emerson strongly recommends using long (10 characters or more), complex passwords wherever passwords are used for authentication. Recommendations on password complexity and management can be found in NIST Special Publication 800-63-3, Digital Identity Guidelines (<https://pages.nist.gov/800-63-3/sp800-63b.html>).

In PACEEdge, each of the following applications has its users and user credential management interfaces:

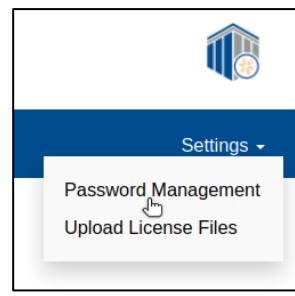
Table 1: Default Passwords

Functionality	Authenticated Subjects (user)	Default Passwords
SSH remote login	admin	Default password: "edgestack"
Cockpit/Linux	admin	Default password: "edgestack"
Chronograf	admin	Default password: "edgestack"
Grafana	admin	Default password: "edgestack"
Node-RED	admin	Default password: "edgestack"
Portainer	admin	Default password: "edgestack"
InfluxDB	admin	Default password: "edgestack"
MySQL	admin	Default password: "edgestack"
Connext/WebHMI	admin@edgestack.com	Default password: "Edgestack123!"

To secure PACEEdge, it comes with pre-configured user roles, each role having different access rights and passwords. An automated Password Management utility helps administrators to configure user role passwords across a whole set of applications. This utility can be accessed from the PACEEdge landing page.

Only the **admin** role can access this utility and change passwords for all the user roles.

Figure 5: Password Management



By default, all the login credentials in PACEEdge are set to the following:

User Name: **admin**

Password: **edgestack**

Once on the Password Management page (Figure 6), the options to change passwords for each user role automatically across all PACEEdge applications are on the top. On the bottom, the user can find the shortcut to change the passwords for the Cockpit/Linux.

Figure 6: Password Management Page

The screenshot shows the 'Password Management' section of the PACEEdge interface. At the top, there's a header with the Emerson logo and the text 'Emerson PACEEdge'. Below the header, a blue bar contains the word 'Services'. In the top right corner, there's a question mark icon. The main content area is titled 'Password Management' and has a sub-section titled 'Manage PACEEdge Application Passwords'. It lists five user roles: 'admin', 'developer', 'service', 'operators', and 'admin@edgestack.com (used in Movicon Project Deploy)'. Each role has a 'Please change Password' message next to it. To the right of each message is a password field with a visibility icon, a 'Change' icon, and a 'Delete' icon. A 'Submit' button is located at the bottom left of this section. Below this, there's a collapsed section titled 'User Rights Overview'. At the bottom of the page, there's a footer with the PACEEdge logo, the text 'Version: V2.2.0', and links for 'Readme | OSS Info'.

Note the line in the utility called **User Right Overview**. If this line is expanded, a table is shown with all PACEEdge applications on top and all users on the side, indicating each user's access rights for each application. In this table, R indicates Read, W indicates Write, and symbol (-) indicates no access rights. For Grafana, instead of R/W, the standard organization roles are used. To get more details about the Grafana roles, please visit https://grafana.com/docs/grafana/latest/permissions/organization_roles/. In Portainer, there is only one user role called Administrator. Notice a separate user admin@edgestack.com. This user is only for deploying the Movicon project onto the PACEEdge device and can not be used to access any other PACEEdge applications.

Figure 7 User Roles and their access rights

The screenshot shows the 'User Rights Overview' table. The columns represent applications: MySQL, InfluxDB, Node-RED, License Mgmt, Portainer, Grafana, and Movicon Project Deploy. The rows represent user roles: admin, developer, service, operators, and admin@edgestack.com. The table uses a light gray background with alternating row colors. The 'Grafana' column uses color-coded icons to represent organization roles: green for Admin, yellow for Viewer, and gray for others. The 'Movicon Project Deploy' column contains a single '-' character for all rows.

User Rights Overview							
Login Name	MySQL	InfluxDB	Node-RED	License Mgmt	Portainer	Grafana	Movicon Project Deploy
admin	R W	R W	R W	R W	Administrator	Admin	-
developer	R W	R W	R W	-	Administrator	Admin	-
service	R W	R W	R W	-	Administrator	Admin	-
operators	-	-	R	-	-	Viewer	-
admin@edgestack.com	-	-	-	-	-	-	R W

Initially, the page will have notes urging the user to change passwords. After the password has been changed once, these notes will disappear; however, a browser refresh might be required to see the change reflected.

Important

It is highly recommended to change passwords by using provided automated utility instead of manually going into a specific application (such as Node-Red) and changing the password there. *If passwords between different applications get out of sync password management utility will fail.*

3.5.1 Changing Passwords via Automated Password Management Utility

The Password Management utility will automatically set passwords for all PACEdge applications and each user role. Furthermore, the utility will provide a link to change Cockpit/Linux passwords at the bottom of the page. Please remember to change both PACEdge Application passwords and Cockpit/Linux passwords.

To use the Password Management utility user needs to be logged in as an **admin**.

To change PACEdge Application passwords, please click on the pencil icon to open the entries, fill in all the fields and then click on *Submit* button at the bottom of this table.

Notice that the *Submit* button will be greyed out and inactive if some password conditions are not met. These conditions are explicitly listed, initially in red, and dynamically change to green once satisfied, see (Figure 8).

CAUTION

Do not use the dollar sign "\$" in a password. The dollar sign "\$" special character is incorrectly labeled as a supported character. Using "\$" can result in PACEdge applications being inaccessible and unable to reach Grafana or Telegraf.

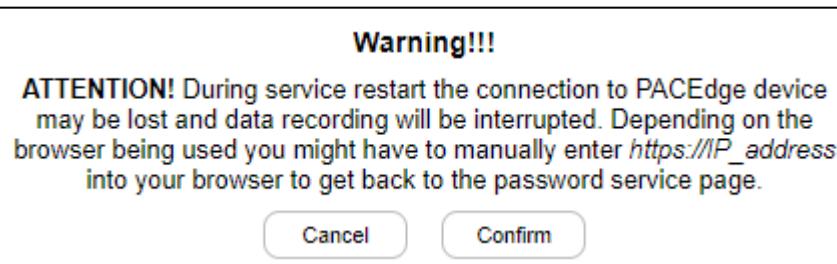
Figure 8: Hints Dialog

Login Name	Admin password	Rules
admin	All passwords are different ✓ At least 8 characters ✓ New Password = Confirm Password ✗
developer	
service	
operators	
admin@edgestack.com (Movicon.NExT Editor)	

Note: The user could choose to change only a subset of all passwords, which might be practical later on, but the first time this utility is run, the user MUST set all passwords as this also sets up roles in databases and other utilities.

Once the *Submit* button is clicked, a dialogue box will warn that PACEedge Applications will be restarted during the password change, including the webserver that serves this page. Depending on the browser used, the user may receive a message indicating that the connection has been lost. In this case, the user should re-enter https://ip_address to return to the PACEedge landing page. The user can return to the Password Management application by going to Settings in the upper-right corner of the page.

Figure 9: Admin Password Change in Progress



Once you click the Confirm button, some additional checks will be performed, and passwords will be changed if there is no error. In case of an error, an Error page will be shown (Figure 9). Following are most common reasons for this error:

- Current admin password is incorrect. The Password Management utility uses the admin password from the PACEEdge Applications, not for Linux/Cockpit
- The admin role password in one of the PACEEdge Applications (Node-RED, Grafana, Portainer, InfluxDB, MySQL) is different than in others. The most likely way for this to happen is when upgrading PACEEdge from Version 2.1 to 2.2, where Portainer password was never changed and is still default **edgestack**. If this is the case, log into Portainer directly using **ip_address/portainer** (ex: **192.168.3.100/portainer**) as admin with the old password (**edgestack** if left at default) and change it to match admin password for the rest of the PACEEdge applications.

Figure 10: Error When Entering Passwords



Note: The Password Management utility will change passwords for each user role in each PACEEdge Application. However, in Node-RED flows, nodes accessing InfluxDB and MySQL applications still need to manually set/change the password (s).

Similarly, in Grafana, the Password Management utility will automatically change passwords for the three PACEEdge default databases as shown in Figure 11. However, if the user has set up additional custom databases, those passwords need to be updated manually.

Figure 11: PACEEdge default databases pre-configured in Grafana



3.5.2 Changing Cockpit/Linux User Passwords

Please note that in addition to using the Password Management utility to change all passwords for the PACEdge Applications, the Cockpit/Linux user passwords must be managed separately.

Note: The automated Password Management utility, described in the previous chapter, does not change Cockpit/Linux user passwords. The default **admin** password is **edgestack**.

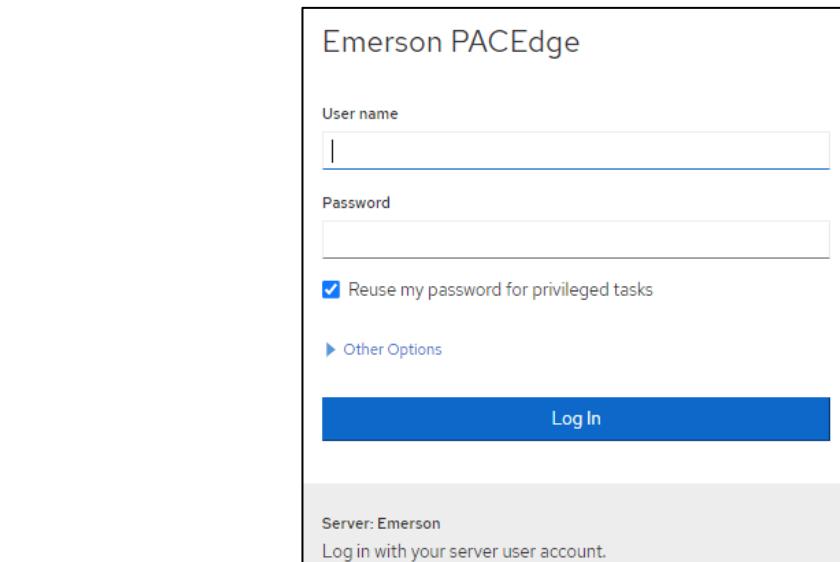
The easiest way to manage the Cockpit/Linux user passwords is to click on the Cockpit button, which is a link, on the bottom portion of the Password Management page, Figure 12: Cockpit/Linux Password Management Link

Figure 12: Cockpit/Linux Password Management Link



Once asked, log in with the **admin** credentials (Figure 13). The user will be prompted to change the default admin password if this is the first login.

Figure 13: Login to Cockpit/Linux



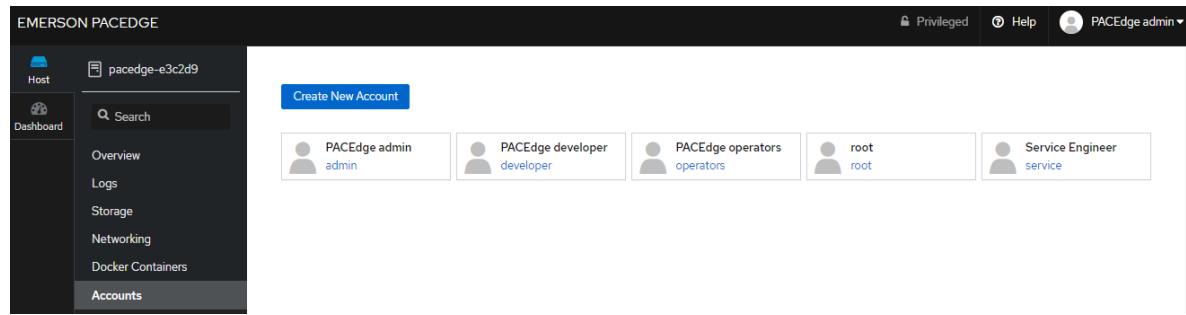
After changing the **admin** password, log out and log in to Cockpit again so that changes can take effect and the admin privileges are granted. The user can do this by accessing a drop-down menu in the screen's upper-right corner.

Once logged in as admin, set the passwords for the developer, service, and operators roles, Figure 14

⚠ CAUTION

Do not set any password for the **root** user, as this would enable the **root** account and create a cybersecurity vulnerability.

Figure 14: User Profiles



3.6

PACEEdge System Level Settings

PACEEdge is Linux-based software but designed to be user-friendly with a graphical user interface (Cockpit) for all basic configuration tasks. To access Cockpit, open the browser, and type in the IP address of PACEEdge. This will bring you to the PACEEdge landing page, where you can click on the Cockpit icon.

3.6.1

System Configuration Changes via Cockpit

Cockpit supports most system configuration tasks, such as:

- Changing Linux/Cockpit password refer to: Section 3.5.2 Changing Cockpit/Linux User Passwords
- Adding /Removing users accounts
- Changing Hostname refer to: Section Changing Host Name under Section 3.6.1
- Modifying IP addresses, VLANs, enabling Firewall
- Managing Storage
- Managing Docker Containers
- Browsing and moving files between PACEEdge and remote computer
- Monitoring CPU, Memory, Network, Storage usage, and performance
- Analyzing system logs

- Monitoring system services
- Setting up the OPC UA Port Forwarding feature
- Shortcuts to other PACEEdge applications
- Linux Terminal (for users that desire further customization)

Changing Host Name

PACEEdge supports mDNS/DNS-SD protocols, allowing devices to be discovered on the network using Hostname. By default, PACEEdge systems are delivered with Hostname set to **pacedge-xxxxxxxx**, where **xxxxxxxx** is either an 8-digit serial number of the hardware device or the last 6-digit MAC address. If HW does not allow access to this information, **xxxxxxxx** is set to word **Default**.

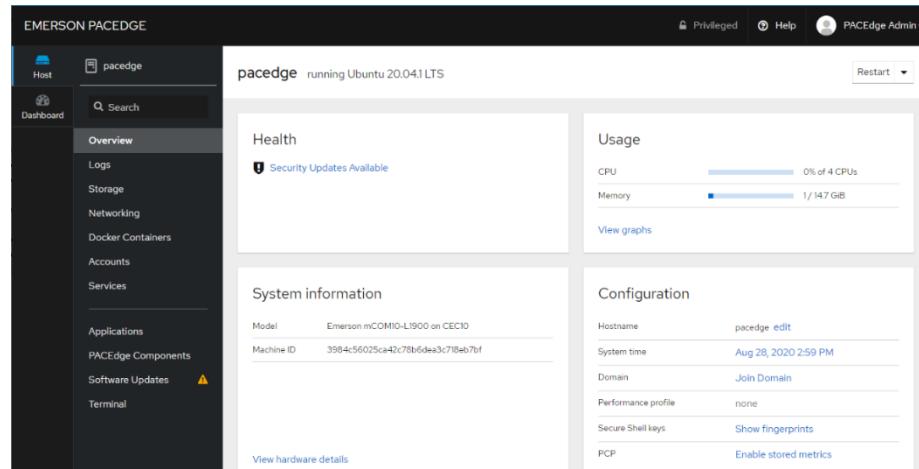
Note: In CPL410 and CPE400 serial number of the HW is not known to the Linux; hence last six digits on the Ethernet controller MAC address are used. To change the Hostname:

1. From the PACEEdge landing page, open Cockpit and log in as an admin user.
2. In the Overview tab, in the Configuration section, click on the **edit** link that follows the Hostname.

Gracefully Restarting, Shutting Down the System

To gracefully shut down the system, log in to Cockpit and select the desired action in the Overview tab in the upper-right corner. Note that when selecting an action, there is also an option to select time delay.

Figure 15: Cockpit Screen



For further Cockpit details and documentation, please consult online resources.

3.6.2 Physical – Logical Ethernet Port Mapping

The Linux Operating system enumerates Ethernet ports in a way that is not obvious which physical port corresponds to which logic port, as seen in Cockpit. The tables below will provide physical to logical Ethernet port mapping for different HW devices

Table 2 RXi2-BP Physical to Logical Ethernet Port Mapping

Physical Ethernet port (Label on device)	Logical Ethernet Port (seen in Cockpit)
ETH 0	enp1s0f0
ETH 1	enp10s0
ETH 2	enp2s0
ETH 3	enp7s0

Table 3 RXi2-LP Physical to Logical Ethernet Port Mapping

Physical Ethernet port (Label on device)	Logical Ethernet Port (seen in Cockpit)
ETH (upper, next to Serial)	enp2s0
ETH (lower, next to USB)	enp1s0

Table 4 CPL410/CPE400 Physical to Logical Ethernet Port Mapping

Physical Ethernet port (Label on device)	Logical Ethernet Port (seen in Cockpit)
ETH	enp1s0

3.7 PACEEdge License File

PACEEdge is protected with the license file, which is locked to the physical device that PACEEdge is running. PACEEdge devices will come with software and a valid license pre-installed from the factory. However, if performing a restore to the factory default, as described in Section 8.2 *PACEEdge Software Restore on RXi2-BP, RXi2-LP IPCs*, on RXi2-BP and RXi2-LP a license will have to be manually installed. Note that on CPL410 and CPE400 Factory Default image is integrated within the unit, hence if restored, the installation process will automatically restore the valid license, and the user does not have to take any action.

To provision the new license, follow the steps below:

1. Once the PACEEdge device is booted, and the user accesses it via the landing page (using a browser), the license status will be indicated at the top of the page (one of the messages as shown below, message in green: OK, message in red: Error):

Figure 16: License Validation (green message: OK, red message: Error)



2. If the message indicates **License Validated**, nothing is to be done; otherwise, copy the MAC Address shown in the message (for instance, 00:20:ce:e3:c2:d9) and get the serial number of the PACEEdge device from the label on the unit. Contact Customer Care to obtain the valid license files. Contact details can be found in the *Technical Support* section located at the end of this document) **Note:** you will need two files: 1) license.json and 2) license.sig
3. Once both license files have been received from Customer Care, go into the PACEEdge landing page, Settings, Upload License Files option, and upload both files:

Figure 17: Upload License Files Menu

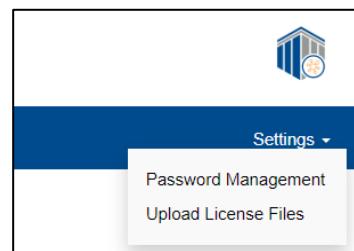
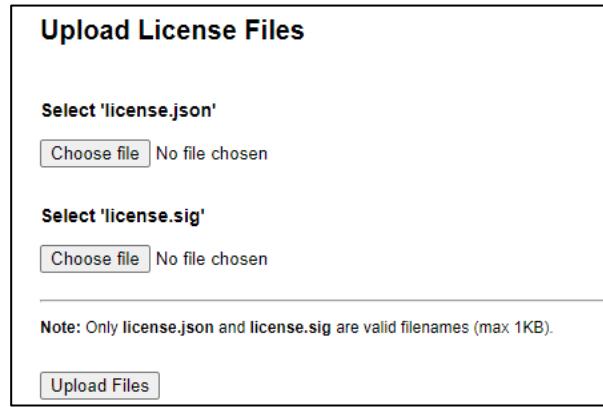


Figure 18: Browse for License Files



Once the files have been uploaded, perform a graceful reboot of the PACEdge device via Cockpit. Information on how to gracefully reboot the system can be found in Section Gracefully Restarting, Shutting Down the System.

4. After reboot, wait a few seconds for all services to fully come up and refresh the browser. If successful, the **License Validated** message in step 1 should be visible.

3.7.1

Backing Up License Files

If desired, license files on the unit can be backed up to a computer that PACEdge is being accessed from and used later when for some reason license is lost. To do so, use the Navigator feature in Cockpit and download and save locally the following two files:

- /home/admin/pacedge/emerson-software/license/license.json
- /home/admin/pacedge/emerson-software/license/license.sig

Note: when using Navigator to download the files, right-click on the file and select the **Download** option. If the error is reported, log out of Cockpit and, log back in as an **admin** user, and repeat the download steps.

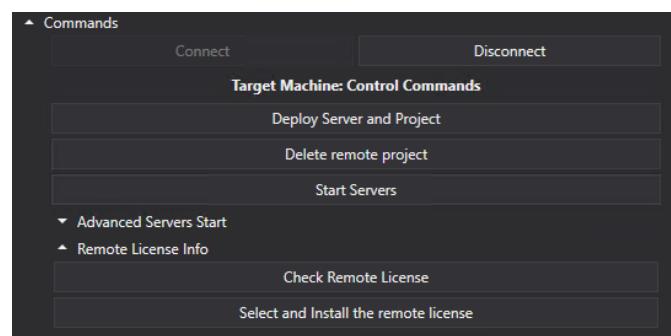
3.7.2 Licensed Connex and WebHMI Features

PACEdge license file enables certain Movicon features, such as Connex or WebHMI. To verify what Movicon features are enabled, please connect to the unit running PACEdge from your workstation where Movicon is.NExT software is installed and does the following:

1. Follow the steps described in: Movicon.NExT to Connex/WebHMI on IPC to establish a connection to PACEdge using the Deploy Project window

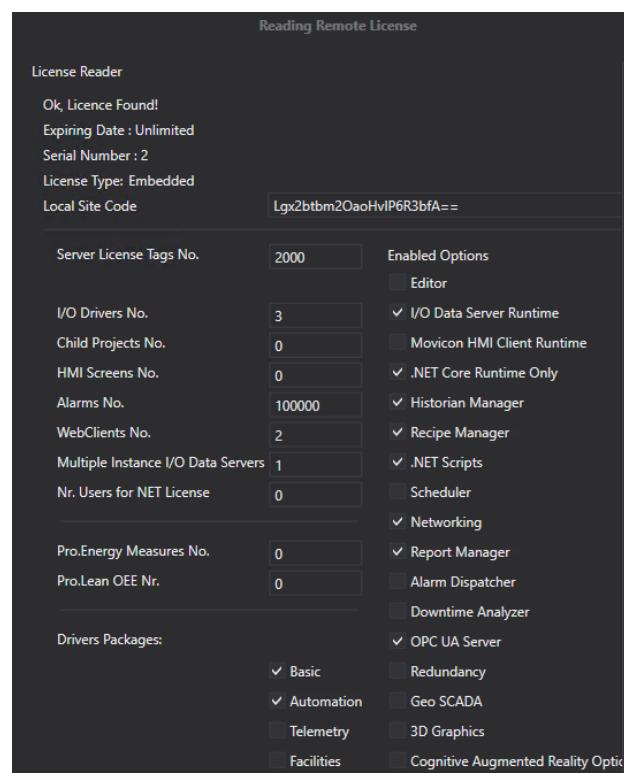
Go to the Remote License Info section and click on Check Remote License:

Figure 19: Connex and WebHMI License Check Dialogue



2. The picture below shows the enabled features for the Connex SKU:

Figure 20: License Enabled Features in Connex SKU

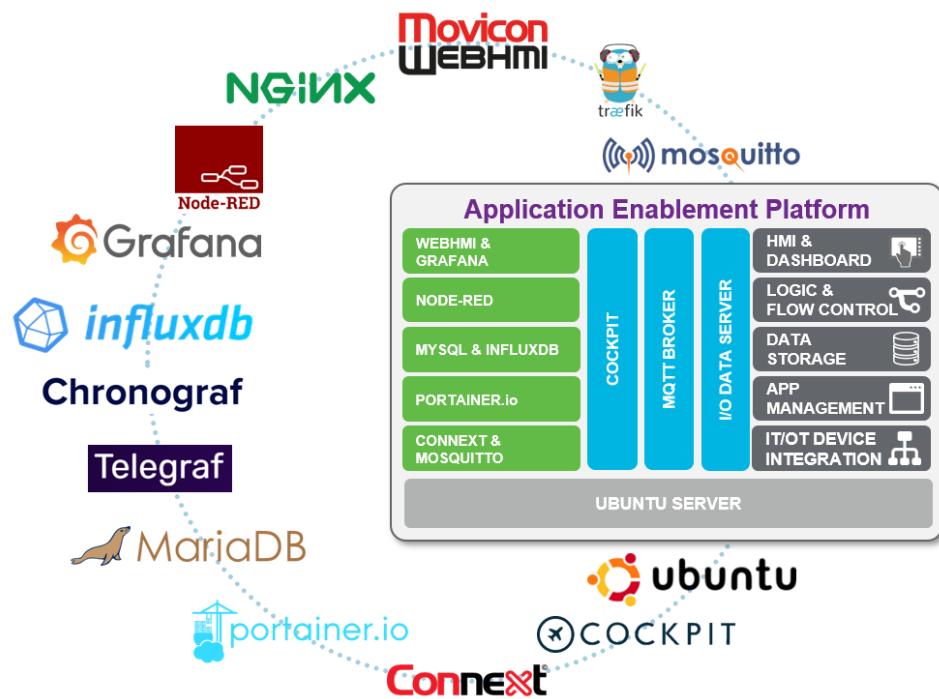


Section 4: PACEEdge Architecture Details

4.1 PACEEdge Applications

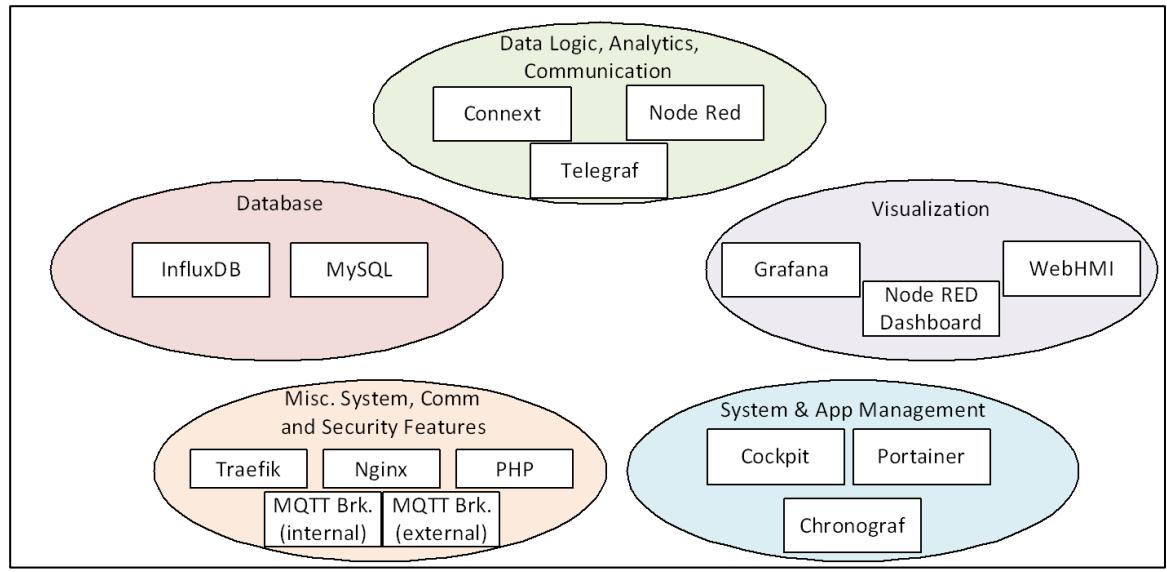
PACEEdge entails several communications, data processing, data storage, and visualization applications that run on the Linux operating system. The following diagram gives an overview of the components that make up PACEEdge:

Figure 21: Software Overview



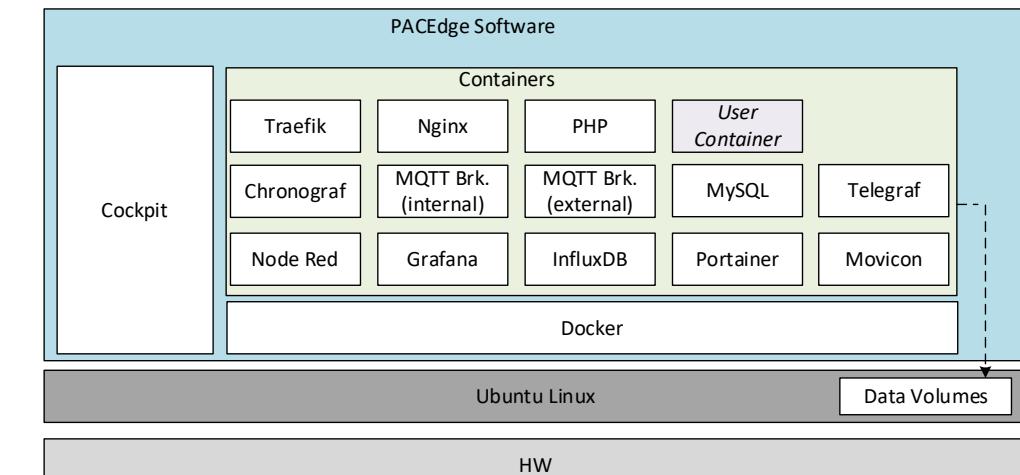
PACEEdge tools and applications, based on the function that they perform, can be grouped into functional categories as follows:

Figure 22: PACEEdge Application Categories



PACEEdge was designed using Docker architecture, in which each application runs in its own Docker container (Figure 23). Since containers are designed to be easily replaceable, they do not retain an internal state between reboots unless specifically designed. With PACEEdge, selected containers will map some of their data to data volumes on the host Linux system so that Node-RED, Grafana, and database changes can be saved between Container restarts and updates.

Figure 23: Containerized PACEEdge Implementation



4.1.1 Application Development Components

Node-RED

Node-RED version: V2.2.2

Node-RED is the logic engine of the PACEdge. It provides a graphical way to wire together different APIs and services, enabling event-driven logic implementations. Node-RED is well known for its broad adoption in the software community and has many freely available nodes that can be easily installed. PACEdge comes with a large selection of pre-installed nodes, allowing users to easily send and receive data via MQTT, OPC-UA, ModbusTCP, and ModbusRTU interfaces. It also allows users to process, visualize data via the dashboard, store data in InfluxDB or MySQL databases, and send email alerts. Node-Red also has nodes for cloud connectivity.

Node-RED is implemented in a Docker Container and can be accessed via browser at *ip_address_of_PACEdge/nodered/*

For more details about Node-RED, please refer to www.nodered.org

Connext and WebHMI

Supported Movicon.NExT version: V4.2

Connext is a data gateway, supporting a large number of field busses and proprietary communication protocols, allowing to receive data, internally share data with other applications, historize data and make it available to other software services via OPC UA Server

WebHMI, in addition to data gateway functionality, adds HMI visualization capability.

Grafana

Grafana version: v9.0.2

Grafana is a visualization tool that lets users view and analyze data and create alerts. Even though the Node-RED Dashboard already has its own data visualization, Grafana brings extra features and ways to scroll and zoom into specific portions of the graph. Grafana works with several databases, but in PACEdge, it is suggested to use InfluxDB and MySQL.

Grafana is implemented in a Docker Container and can be accessed via browser at *ip_address_of_PACEdge/grafana*

For more details about Grafana, please refer to www.grafana.com

MQTT

MQTT version: V2.0.14

PACEEdge implements two MQTT brokers: one is meant to be used for PACEEdge internal communication between Docker containers, and the second one for external communication. Node-RED does have MQTT nodes pre-installed. MQTT brokers are implemented using Mosquitto open-source software.

For more details about MQTT, please refer to www.mqtt.org

Traefik

Traefik version: V2.8.1

Since most of the PACEEdge applications are installed in their own Docker Containers and communicate with each other via network interfaces, Traefik is used as a front-end edge router to receive requests from the external world and route them to an appropriate service. Traefik enables accessing services such as Node-RED by simply adding /nodered at the end of the PACEEdge IP address instead of the port number. In addition to simplifying application access and navigation Traefik also improves cybersecurity by reducing the number of open ports.

For more details about Traefik, please refer to docs.traefik.io/

Nginx and PHP

Nginx version: V2.2.0

NGNIX and PHP are services used to serve an initial landing page, which has important version information, end-user license agreement, password management, and license file management services, as well as shortcuts to services like Cockpit, Node-RED, Grafana, Portainer, Chronograf, WebHMI.

Telegraf

Telegraf version: V1.23.2

PACEEdge implements Telegraf agent, which is plugin-based and can pull data, statistics, and logs from a variety of databases, underlying system resources (CPU, memory, disc, kernel, software logs, docker containers, etc.), and external devices, heavily focusing on IoT protocols, such as MQTT, AMQP, Cloud resources, etc. For the complete list of available plug-ins, please refer to:

<https://www.influxdata.com/products/integrations/? integrations dropdown=telegraf-plugins>

By default, Telegraf is configured to gather system health statistics, listen to a specific MQTT topic on the internal MQTT Broker, and store resulting data in the InfluxDB database.

4.1.2 Database Components

MariaDB (MySQL)

MySQL version: MariaDB v10.5.16

MariaDB (MySQL) is a relational database. Node-RED has a pre-installed node to access this database. I/O Data Server in Connexx can use MariaDB as a database. Grafana can pull, analyze, and help visualize data from MariaDB. MariaDB is implemented in a Docker Container.

For more information about MariaDB (MySQL), please refer to mariadb.org/

InfluxDB

InfluxDB version: v1.8.10

InfluxDB is a time-series database. Node-RED has nodes that enable the user to store and query data to and from InfluxDB, and Grafana connects to InfluxDB and retrieves data for visualization. Telegraf uses InfluxdDB to historize data it receives via input connectors

InfluxDB is implemented in a Docker Container and is expected to be managed by either Node-RED or Chronograf applications.

For more details about InfluxDB, please refer to www.influxdata.com

4.1.3 Administration Components

Cockpit Description

Cockpit version: V215

PACEEdge is designed to offer the user a GUI experience. Even though it is based on a Linux operating system, all main system management tasks can be done via GUI, and a Cockpit is a tool that makes it happen. Cockpit provides system status and health information, resource (CPU, memory, storage, network) usage, network (IP address) management options, Docker Container status and management options, user management options, and different logs for troubleshooting.

Since Cockpit is meant to manage Linux operating system tasks, it runs on Linux as a native application and not in a Docker Container.

Cockpit can be accessed via browser at ip_address_of_PACEEdge:9090/cockpites/

For more details about Cockpit, please refer to www.cockpit-project.org

Portainer

Portainer version: V2.11.1

PACEEdge is heavily utilizing Docker's container-based implementation, allowing users to add their own containers. Even though Cockpit already has Docker Container management features, a dedicated docker management tool Portainer adds additional functions and visualization options. Portainer allows users to monitor, start, and stop containers, check the container log file, configure restart policies, open ports, etc.

For more details about Portainer, please refer to www.portainer.io/

Chronograf

Chronograf version: V1.9.4

Chronograf is a management interface for InfluxDB. With Chronograf, users can search and query data that is stored in InfluxDB, as well as perform database management tasks. Chronograf as well offers data visualization capabilities, similar to Grafana.

Chronograf is implemented in a Docker Container and can be accessed via a browser at ip_address_of_PACEEdge/chronograf/

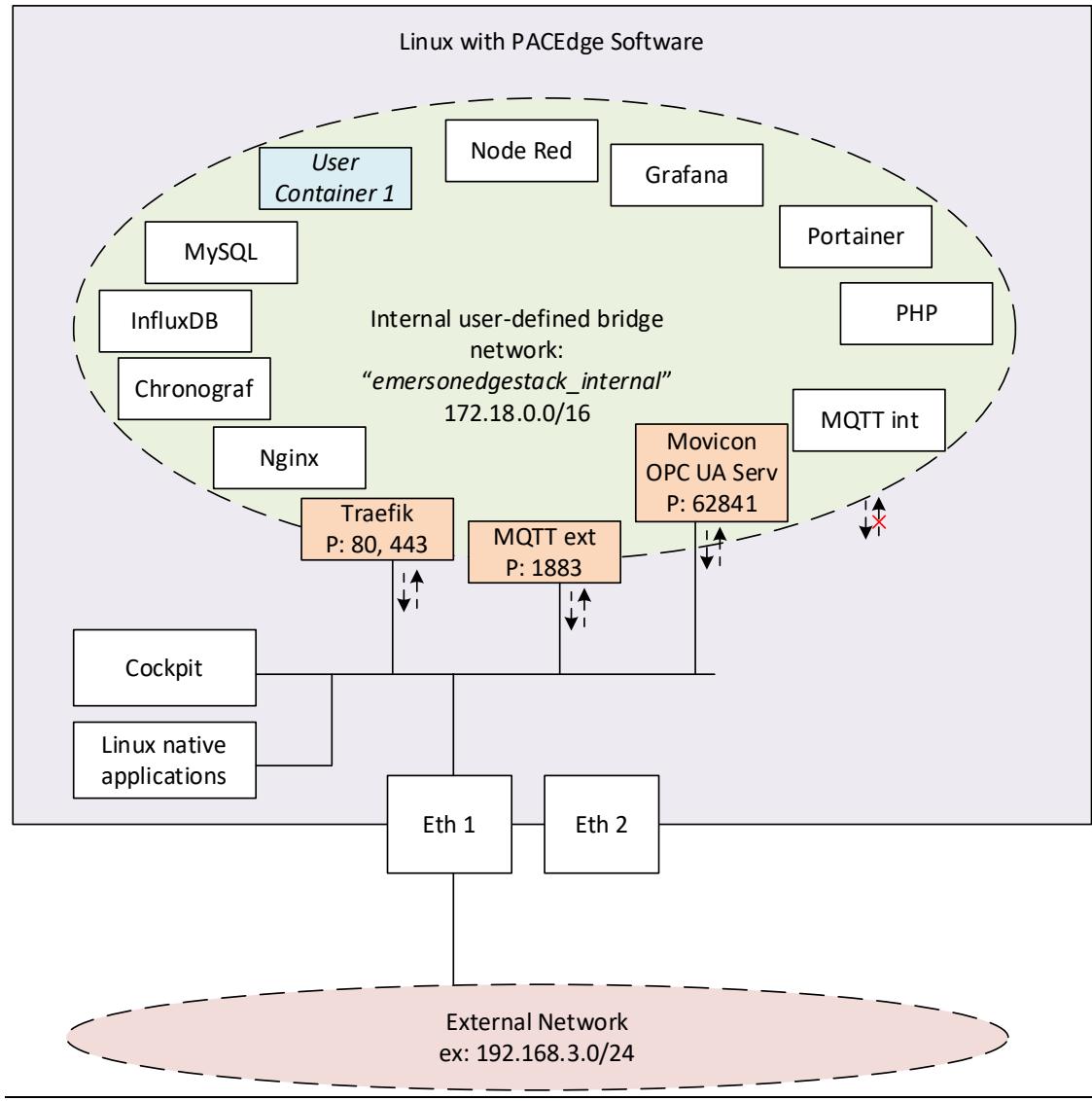
For more details about Chronograf, please refer to www.influxdata.com/time-series-platform/chronograf/

4.2 PACEEdge Communications and Data Flow

Considering that most of the applications in PACEEdge are implemented using Docker Containers, the main communication between them is implemented using network interfaces. For security and traffic segregation reasons, PACEEdge uses an internal user-defined bridge network with IP subdomain address 172.18.0.0/16. Each Docker container is connected to this network and can access external services, such as the Internet, but are not directly accessible from outside. Some services, such as Traefik, external MQTT broker, and OPC UA server, have open ports (80, 443, 1883, 62841), enabling these services to be reached from the outside.

External Ethernet ports and IP addresses can be easily managed via Cockpit or other standard Linux tools.

Figure 24: PACEdge Network Communication Paths



4.2.1 PACEEdge Internal Communication Parameters

To access InfluxDB and MySQL from within the PACEEdge, say from Node-RED flows or Connex, as well as for MQTT communication, both internal and external, certain communication parameters and authorization data need to be specified. Note that user passwords are the ones that have been set up at the beginning of PACEEdge use.

InfluxDB Access Parameters

- Host: influxdb
- Port: 8086 (internal access only)
- Usernames: admin, developer, service
- Passwords: as changed by the user

Grafana communication parameters

Grafana is configured with the following users:

- admin: Full rights, can set up users and their passwords, add databases
- developer, service: can create a dashboard, but cannot set up databases
- operators: read-only access

To configure databases inside Grafana, log in as an admin user.

To set up the InfluxDB database:

- URL: `http://influxdb:8086/`
- Database: test (or your own database name, need to match the InfluxDB database name in Node-RED)
- User: admin or developer
- Password: as changed by the user
- HTTP Method: GET

To set up MySQL database:

- Host: mysql:3306
- Database: test (or your own database name, need to match MySQL database name in Node-RED or Connex)
- User: admin or developer
- Password: as changed by the user

MySQL communication parameters

- Host: MySQL
- Port: 3306 (internal access only)
- Username: admin, developer, service
- Password: as changed by the user

Internal MQTT communication parameters

- Server: emerson-mqtt-internal-ipc
- Port: 1883

4.2.2 PACEEdge External Communication Parameters

PACEEdge Landing page

To access the PACEEdge landing page via browser:

- https://ip_address_of_PACEEdge

Node-RED

To access Node-RED via browser:

- https://ip_address_of_PACEEdge/nodered/
- shortcut on the landing page
- Username: admin, developer, service, operators
- Password: as changed by the user

Grafana

To access Grafana via browser:

- https://ip_address_of_PACEEdge/grafana/
- shortcut on the landing page
- Username: admin, developer, service, operators
- Password: as changed by the user

Cockpit

To access Cockpit via browser:

- [https:// ip address of PACEdge:9090/cockpites/](https://ip_address_of_PACEdge:9090/cockpites/)
- shortcut on the landing page
- Username: admin, developer, service
- Password: as changed by the user

Portainer

To access Portainer via browser:

- [https:// ip address of PACEdge /portainer/](https://ip_address_of_PACEdge /portainer/)
- shortcut on the landing page
- Username: admin, developer, service
- Password: as changed by the user

WebHMI

To access WebHMI via browser:

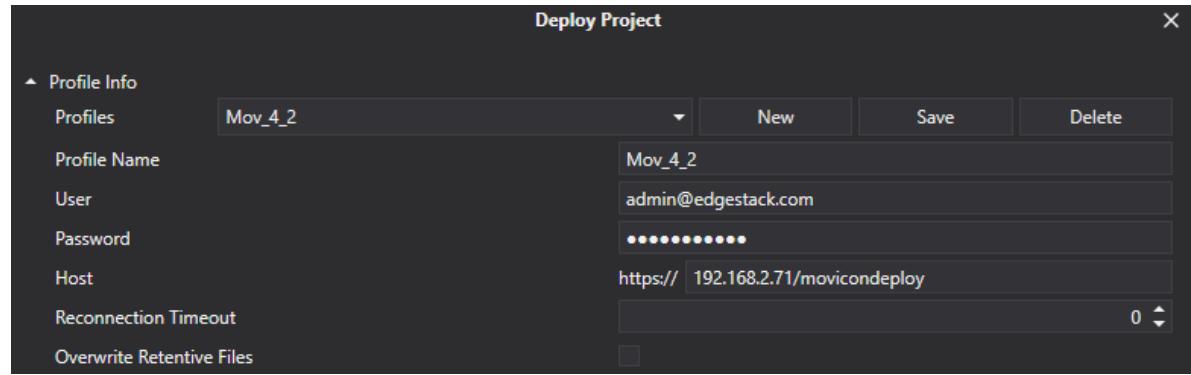
- [https:// ip_address_of_PACEdge /movicon/](https://ip_address_of_PACEdge /movicon/)
- shortcut on the landing page

Movicon.NExT to Connex/WebHMI on IPC

Movicon.NExT software, running typically on Windows computers, is used to develop Connex and WebHMI applications. Once development is done, these applications must be uploaded to an industrial PC running PACEdge. To establish this upload, the following credentials need to be specified in Movicon.NExT:

- User: admin@edgestack.com
- Password: as changed by the user
- Host: xx.xx.xx.xx/movicondeploy
(where xx.xx.xx.xx is an IP address of the remote industrial PC, such as 192.168.3.100)

Figure 25: List of credentials to deploy the Movicon project



Note: User name and password must match what is specified in /home/admin/pacedge/emerson-movicon/appsettings.json file. Starting with PACEEdge release v2.2, please use the Password Management utility to change this password and do not modify this file directly.

External MQTT Communication

PACEEdge has a second MQTT Broker, which is designated to be used for MQTT communication with external devices.

- When accessing (external) MQTT broker in the same PACEEdge unit:
 - Server: emerson-mqtt
 - Port: 1883
- When accessing the MQTT broker in the other PACEEdge unit, attached via Ethernet:
 - Server: xx.xx.xx.xx:1883
(where xx.xx.xx.xx is an IP address of the remote unit, such as 192.168.2.90)
 - Port: 1883

By default, MQTT security (username, password, certificates, encryption) is disabled, but users can choose to enable it via MQTT Broker configuration settings. For details, please refer to: mosquitto.org

4.2.3 Accessing Connext OPC UA Server

Connext has an integrated OPC UA Server, which can be used by internal PACEEdge applications, such as Node-RED, but also accessed externally by using OPC UA Clients and Browsers.

To access the Connext OPC UA server from outside the PACEEdge device, specify the IP address of the PACEEdge device and append the following port number **62841**. The example should look like this:

opc.tcp://192.168.3.100:**62841**

4.2.4 Accessing Connext OPC UA Server from Movicon.NExT Browser

When accessing the Connext OPC UA server on the PACEEdge device from Movicon.NExT browser uses the Hostname instead of an IP address. The Movicon.NExT browser will check for a cybersecurity certificate on the target device, which is issued using the Hostname. Therefore, to pass the security check, the Hostname, not the IP address, has to be used.

To find out the hostname of the Connext docker container, please follow the steps below:

1. Open Cockpit, go to the Terminal tab
2. Type command: docker exec emerson-movicon hostname and hit enter
3. Copy hostname shown

Figure 26: Finding the Hostname with Terminal

```
admin@pacedge-e3c228:~$ docker exec emerson-movicon hostname
pacedge-e3c228
```

Note: Starting with PACEEdge v2.2.0, main Linux and Movicon container hostnames have been changed to be unique. As a result, the Movicon hostname is no longer derived from the container ID and cannot be read using the Cockpit Docker tab.

4. If Movicon.NExT Editor runs on a Windows computer; the Hostname must be associated with the PACEdge device IP address. This association is established in the host's file:

C:\Windows\System32\drivers\etc\hosts

This file is protected and cannot be edited directly, therefore use the following procedure:

- a. Copy the host file to the Desktop folder.
- b. Open the host file with Notepad or another editor.
- c. Add a line with the PACEdge IP address, space, and Hostname of the container, such as:

192.168.3.100 130d8143d8bd

- d. Save the file and copy it back to the original location, overwriting the original file.
- e. Restart Windows.

4.2.5

Accessing OPC UA Server via Port Forwarding

Starting with version v2.2 PACEdge implements the OPC UA Port Forwarding feature. The OPC UA Port Forwarding feature enables temporary forward OPC UA traffic from one Ethernet port to the other on the PACEdge device. This feature is very handy when using Movicon.NExT editor to browse OPC UA variables on the PLC, which is connected to PACEdge but is not directly accessible from the user PC where Movicon is.NExT editor is running. This is the case on CPL410/CPE400 device, where PLC's OPC UA server is accessible via an internal virtual Ethernet port (VNIC) or in RXi2-BP/RXi2-LP systems where one Ethernet port is used to connect to PLC and another Ethernet port is used to connect to engineering PC where Movicon.NExT is running.

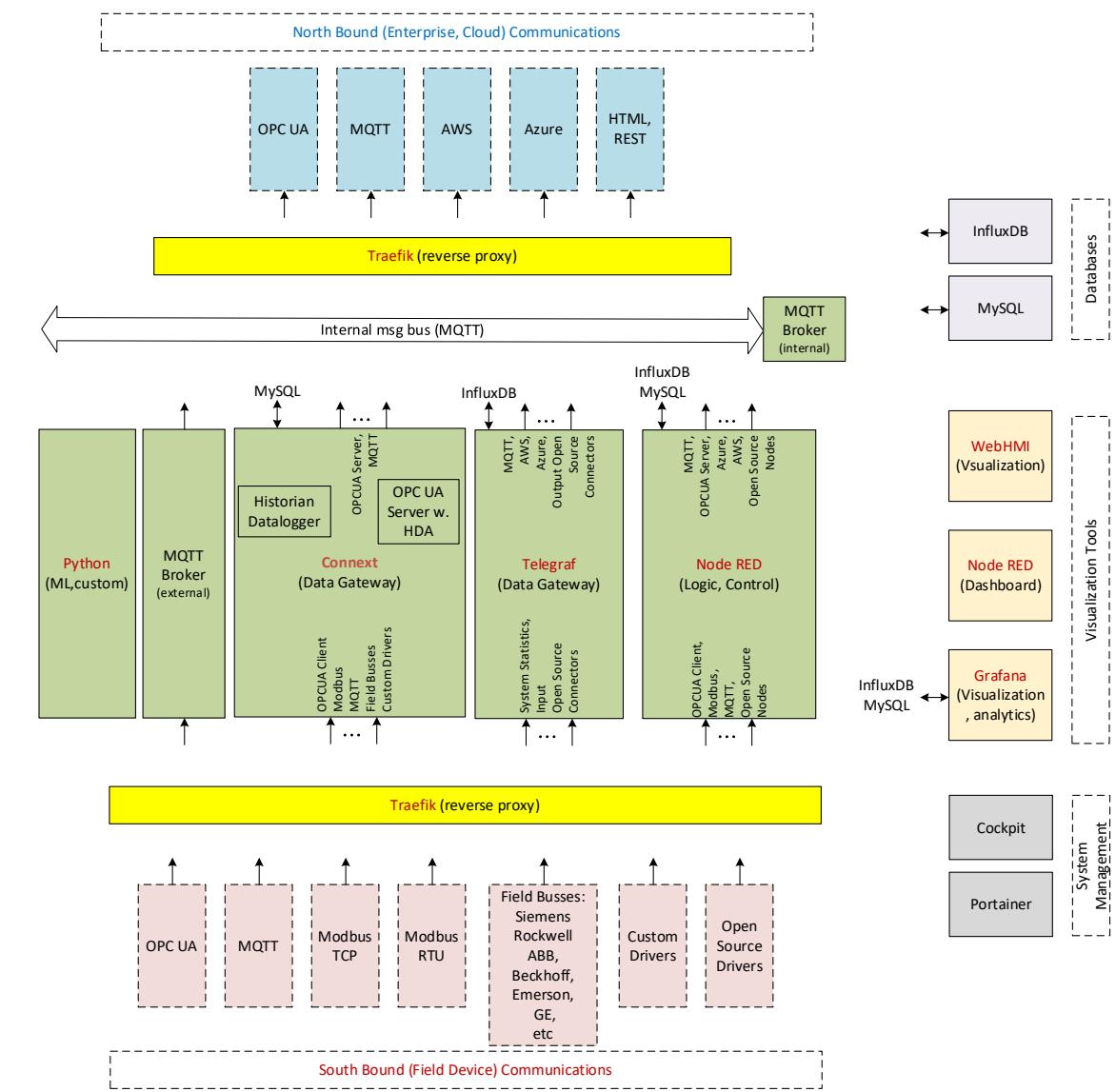
To use the OPC UA Port Forwarding feature, please follow the steps documented in the example: Section 5.6.2 - Example Walk Through, Enable OPC-UA Port Forwarding Feature.

Note: Example is based on CPL410 device, using internal VNIC. If using a different device and Ethernet port, please substitute IP Address and Port Number as required.

4.2.6 PACEdge Suggested Data Flows

In Edge applications, it is common to define data flow as “Southbound” (from field devices, sensors, and PLCs to the gateway) and “Northbound” (from the gateway to SCADA, Enterprise applications, and cloud applications). PACEdge implementation is ideally suited for such data flow scenarios. The following diagram shows different components and their communication capabilities:

Figure 27: PACEdge Communication Capabilities



4.2.7

Southbound Communication Capabilities

To communicate with different field devices, aka Southbound communication, PACEEdge has the following capabilities:

Connext I/O Drivers

Connext data gateway component implements the so-called I/O Driver infrastructure, which is designed to communicate with many different field devices, supporting open as well as proprietary communication protocols, such as MQTT, OPC UA Client, Modbus, IEC 60870-5-104, IEC 61850 MMS, Siemens, Beckhoff, GE, Hilscher, Mitsubishi, Omron, Phoenix and many more. For the complete list, please refer to: <https://www.progea.com/i-o-driver-list-movicon-next/?lang=en>. Furthermore, I/O Driver infrastructure was designed to add new and custom drivers supporting efficient customization efficiently. Data that the driver is receiving is internally stored in the form of a data tag, which can be historized in MySQL, sent out via another driver (such as MQTT), or made available via the OPC UA Server.

Node-RED Nodes

Node-RED has a large open-source community developing and constantly adding new communication nodes. Node-RED supports OPC UA clients and servers, MQTT, Modbus TCP, Modbus RTU, etc. Once data is received by Node-RED, it can be easily processed, stored in MySQL or InfluxDB databases, and sent out via another interface

Telegraf

PACEEdge implements Telegraf agent, which is plugin based utility and can pull data, statistics, and logs from a variety of databases, underlying system resources (CPU, memory, disc, kernel, software logs, docker containers, etc.), and external devices, heavily focusing on IoT protocols, such as MQTT, AMQP, Cloud resources, etc. For the complete list of available plug-ins, please refer to:

https://www.influxdata.com/products/integrations/? integrations_dropdown=telegraf-plugins

For more details on how Telegraf is configured, please refer to an example later in this document: Section 5.7 - Configuring Telegraf.

Custom Driver

Given the open nature of PACEEdge, users can also add their existing communication drivers, for example, written in Python, that run in the separate customer-specific Python docker container.

4.2.8 North Bound Communication Capabilities

To communicate with upper software layers, such as SCADA, enterprise systems, and Cloud, aka Northbound communication, PACEEdge has the following capabilities:

OPC UA Server

PACEEdge has different options to set up an OPC UA Server and make data available to other applications and systems.

Connext comes with an integrated high-performance OPC UA Server and is a recommended option. All data tags within the Connext environment are automatically published via OPC UA Server, which can be accessed at the address:
`opc.tcp://xx.xx.xx.xx:62841`, where xx.xx.xx.xx is an IP address of the IPC.

Alternatively, Node-RED comes with pre-installed OPC UA Nodes, which include OPC UA servers.

MQTT

PACEEdge comes with an MQTT Broker, which can be accessed outside IPC. MQTT Broker used is based on Mosquitto open-source implementation and supports Sparkplug B specification and security and encryption features. For access details, please refer to Section *External MQTT Communication*.

Cloud Connectivity

Cloud connectivity can be established using Node-RED nodes, which readily support AWS, Azure, and other Cloud providers.

Another option is to use Telegraf plug-ins.

Yet, another option is to use a custom docker container and install cloud agents or Python libraries

4.2.9 PACEEdge Internal Communications and Data Flow

PACEEdge can be seen as a Swiss army knife when solving a particular problem. Typically, there are more than one solution and implementation option. It is recommended that the user first understands what tools and options are available and then selects the most appropriate implementation to solve a particular problem.

MQTT – Internal Communication Bus

PACEEdge has two MQTT Brokers, one of them is dedicated to internal communication only and not accessible from outside. MQTT protocol using an Internal MQTT Broker is a recommended internal data bus within the PACEEdge. MQTT is fast, simple to set up, and has low overhead, saving CPU resources. Most agents and applications within PACEEdge,

such as Connex, Node-RED, Telegraf, and Python, readily support MQTT communication. For example, data received by one of the Connex drivers can be automatically published via MQTT, and Node-RED can subscribe to it. Alternatively, Node-RED can publish MQTT data, and Connex can receive it via the MQTT driver. Telegraf can also publish received data via MQTT, such as CPU utilization statistics. Starting with PACEEdge v2.2.0, Telegraf is configured by default to subscribe to Internal MQTT broker, topics starting with emr_v1/..., parse received data, and store it in InfluxDB. This setup can store Connex data in InfluxDB by publishing it to MQTT internal broker with the appropriate topic.

OPC UA – Alternative for Internal Communication

OPC UA can also be used for internal data communication as an alternative to MQTT. For instance, Connex has an OPC UA server, and all data variables that Connex receives are published via OPC UA Server. Consequently, Node-RED can use the OPC UA Client and get data from Connex. Compared with MQTT, note that the OPC UA protocol is significantly heavier on CPU resources and more complicated to set up.

Sharing data via Databases

For not-so-real-time-critical data sharing, an excellent way to consider is storing data in the database and letting other applications extract data directly from it. For example, Connex has a Historian and Datalogger functionality, which can keep the data in a MySQL database. Consequently, Node-RED or Grafana can directly access the data with MySQL and perform further processing or offer visualization.

4.3 PACEEdge Remote Access

PACEEdge is designed to operate within the internal networks, protected by firewalls and other cyber security devices, the so-called on-premise use model. Nevertheless, there are use cases where remote access to the PACEEdge system is required.

Following are a few remote access options that have been tested to work well with PACEEdge

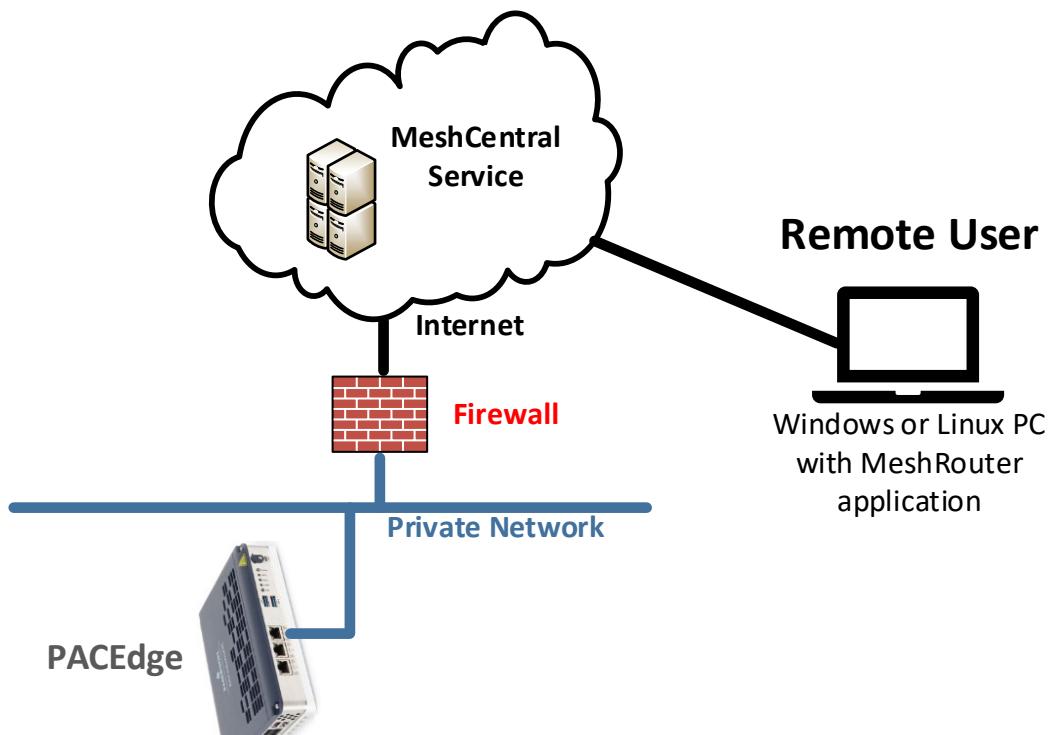
4.3.1 Remote Access using MeshCentral

MeshCentral is an open-source project based on the concept of having a MeshCentral Server hosted on the internet, either a publicly freely available server or your instance of the server, say on AWS or Azure cloud. PACEdge devices, located on a private network, behind NAT and Firewall, would connect to this server. A remote user would log into the server and gain access to the desired device. Supported access type:

- Command line terminal,
- file browser, transfer,
- desktop view (RXi2-BP and RXi2-LP when the actual monitor is plugged into DisplayPort on the PACEdge device).

If an additional MeshCentral Router application is installed on the remote user's PC, web-based access to PACEdge applications (Node-RED, Grafana, Cockpit, etc.) is also supported.

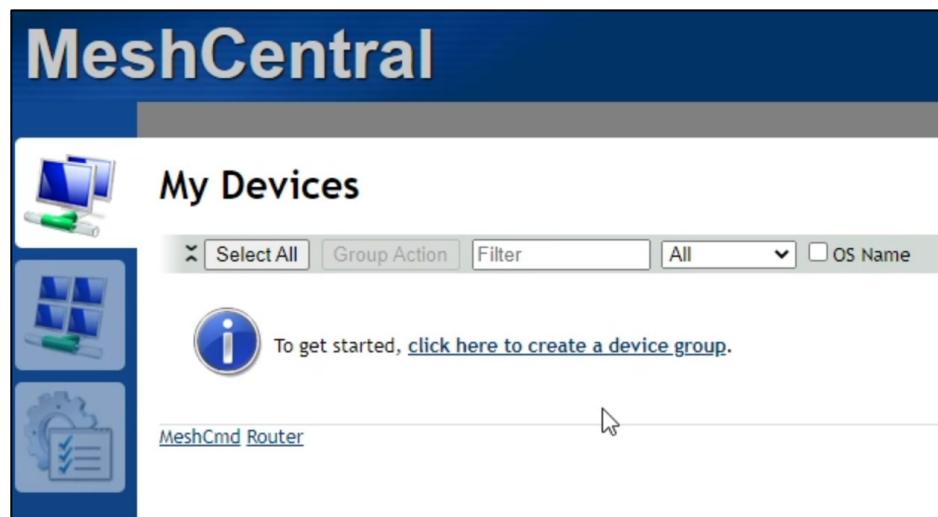
Figure 28 Remote Access using MeshCentral Software



4.3.2 Steps to setup MeshCentral: Create MeshCentral Account

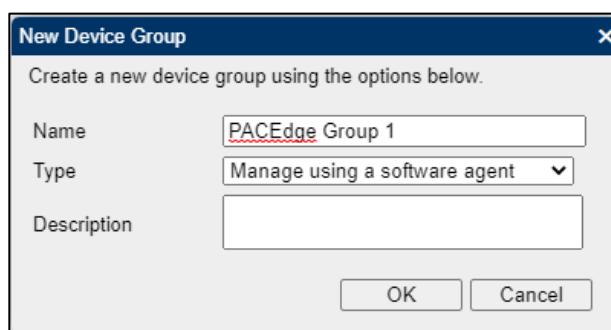
1. Go to meshcentral.com website, click Public Server Login, and **Create one** link.
2. Fill in the required email address, user name, and password, and verify your email address as requested.
3. Log in to your newly created account at MeshCentral. Once logged in, you will get the following view:

Figure 29: Mesh Control View



4. Click on the link to create a device group, give it a name and keep other values as default

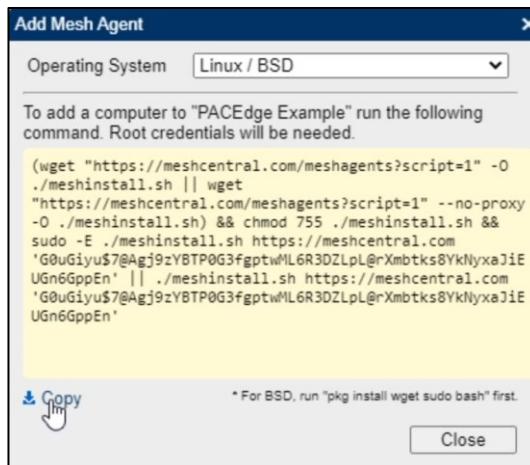
Figure 30: Device Group Name



5. Click on the link to add a device.

6. Select Linux/BSD for the operating system and click on **Copy** to copy the link that gets automatically created.

Figure 31: Add Mesh Agent



Add PACEEdge Device to MeshCentral Server

1. Using your web browser, connect to your PACEEdge device, go to Cockpit, and log in as an admin.
2. Go to Terminal, right-click into the Terminal window, and select Paste to paste the link you copied in the above step. Alternatively, you can use a CTRL+V shortcut to paste the link.
3. Hit enter and enter the admin password once requested.

Figure 32: Enter credentials

```
admin@pacedge:~$ (wget "https://meshcentral.com/meshagents?script=1" -O ./meshinstall.sh || wget "https://meshcentral.com/meshagents1.sh https://meshcentral.com 'G0uGiyu$7@Agj9zYBTP0G3fgptwML6R3DZlpL@rXmbtks8YkNyxaJiEUGn6GppEn' || ./meshinstall.sh https://meshcen--2022-08-18 09:36:32-- https://meshcentral.com/meshagents?script=1Resolving meshcentral.com (meshcentral.com)... 52.32.58.254Connecting to meshcentral.com (meshcentral.com)|52.32.58.254|:443... connected.HTTP request sent, awaiting response... 200 OKLength: 5466 (5.3K) [application/octet-stream]Saving to: './meshinstall.sh'

./meshinstall.sh                                         100%[=====] 2022-08-18 09:36:33 (467 MB/s) - './meshinstall.sh' saved [5466/5466]

[sudo] password for admin:
```

4. Once the installation of the MeshCentral agent is complete, a message similar to the one below will be shown:

Figure 33: Complete Message

```
admin@pacedge:~$ (wget "https://meshcentral.com/meshagents?script=1" -O ./meshinstall.sh https://meshcentral.com 'G0uGiyu$7@Agj9zYBTP0G3fgptwML6R3DZLpLqrXmbtks8YkNyxaJ'; ./meshinstall.sh)
--2022-08-18 09:36:32-- https://meshcentral.com/meshagents?script=1
Resolving meshcentral.com (meshcentral.com)... 52.32.58.254
Connecting to meshcentral.com (meshcentral.com)|52.32.58.254|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5466 (5.3K) [application/octet-stream]
Saving to: './meshinstall.sh'

./meshinstall.sh                                [=====] 100%[=====]
2022-08-18 09:36:33 (467 MB/s) - './meshinstall.sh' saved [5466/5466]

[sudo] password for admin:
Downloading agent #6...
--2022-08-18 09:36:40-- https://meshcentral.com/meshagents?id=6
Resolving meshcentral.com (meshcentral.com)... 52.32.58.254
Connecting to meshcentral.com (meshcentral.com)|52.32.58.254|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3724624 (3.6M) [application/octet-stream]
Saving to: './meshagent'

./meshagent                                         [=====] 100%[=====]
2022-08-18 09:36:43 (2.02 MB/s) - './meshagent' saved [3724624/3724624]

Agent downloaded.
--2022-08-18 09:36:43-- https://meshcentral.com/meshsettings?id=G0uGiyu$7@Agj9zYBTP0G3fgptwML6R3DZLpLqrXmbtks8YkNyxaJ
Resolving meshcentral.com (meshcentral.com)... 52.32.58.254
Connecting to meshcentral.com (meshcentral.com)|52.32.58.254|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26924 (26K) [application/octet-stream]
Saving to: './meshagent.msh'

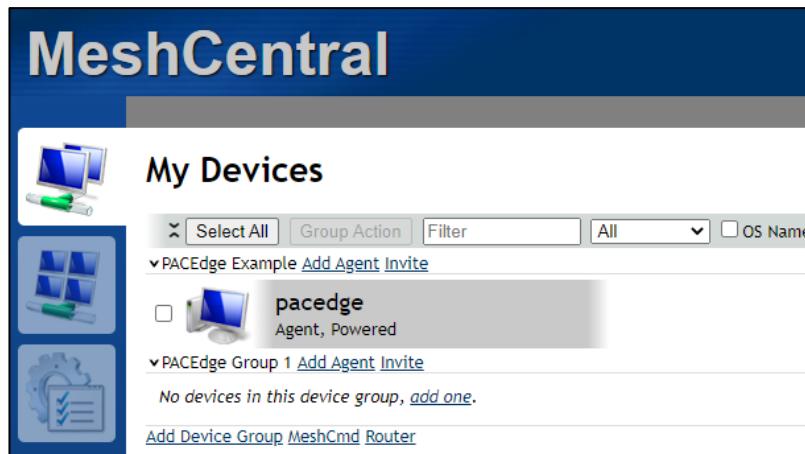
./meshagent.msh                                     [=====] 100%[=====]
2022-08-18 09:36:44 (45.7 MB/s) - './meshagent.msh' saved [26924/26924]

...Checking for previous installation of "meshagent" [NONE]
...Installing service [DONE]
    -> Starting service... [OK]
admin@pacedge:~$
```

Test Remote Access to PACEEdge

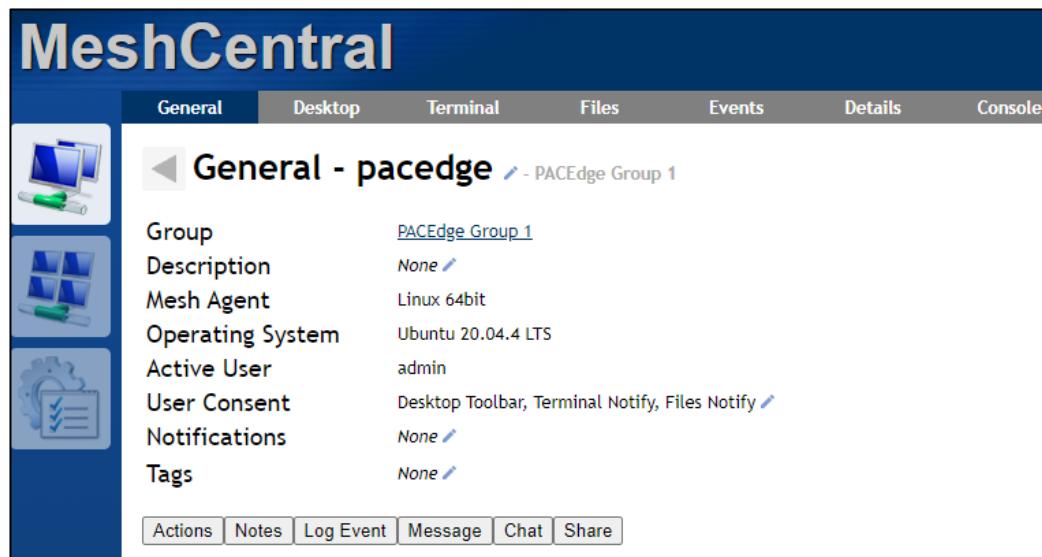
1. Go back to MeshCentral Server, make sure you are logged in, and you will see a PACEEdge device:

Figure 34: PACEEdge Device



2. Click on this device, and you will get a screen that allows you to gain command line and file transfer access

Figure 35: File Transfer Access

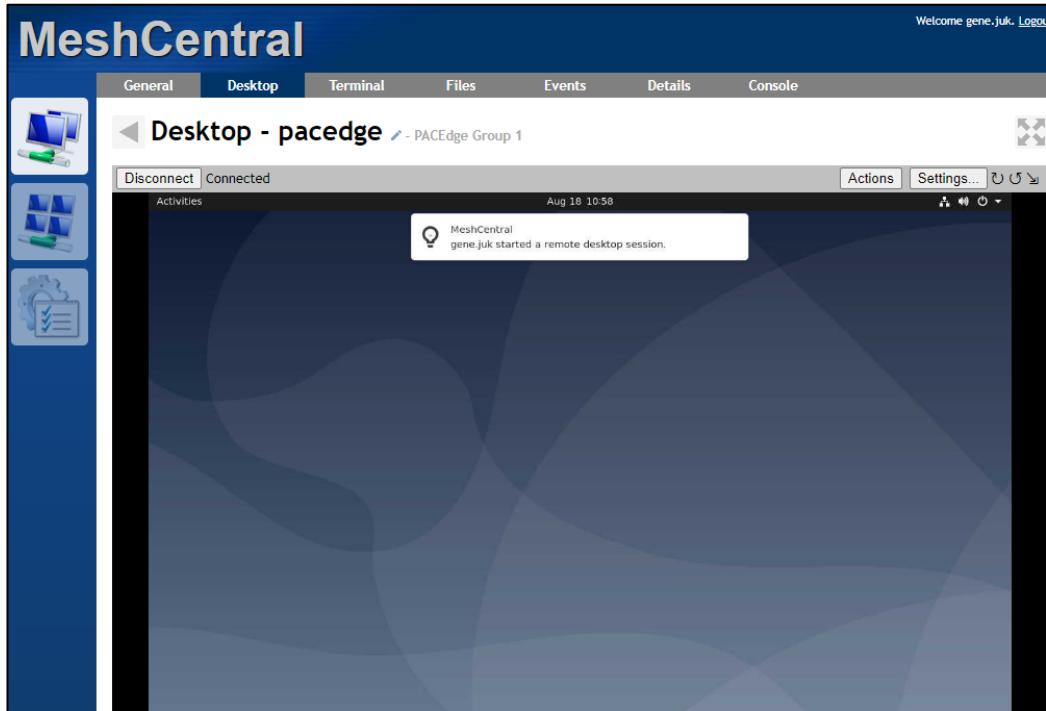


NOTE: depending on the PACEEdge HW SKU and monitor being plugged into the PACEEdge device via DisplayPort interface, the Desktop option might not be available

3. To gain Desktop access to PACEdge, click on **Desktop** and then on **Connect** button.

Note: for this connection to work, a monitor needs to be plugged into the PACEdge device via DisplayPort interface, and the user requires to be logged in to PACEdge via the local monitor, keyboard, mouse interface

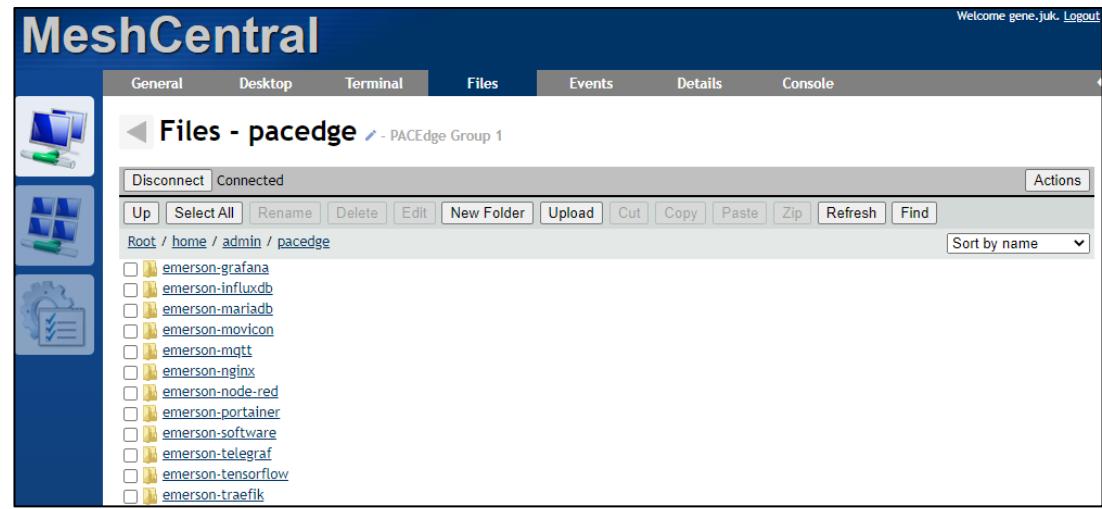
Figure 36: Connected Device



4. To gain command line access to PACEdge, click Terminal and then the **Connect** button.

5. To gain file browser/transfer access to PACEdge, click on **Files** and then on **Connect** button

Figure 37: Gaining file Browser/Transfer Access to PACEdge



Setup Remote Access to PACEdge Applications (Node-Red, Grafana, Cockpit)

1. Go to meshcentral.com web site and before logging in, click on the Downloads link

Figure 38: Downloads Link



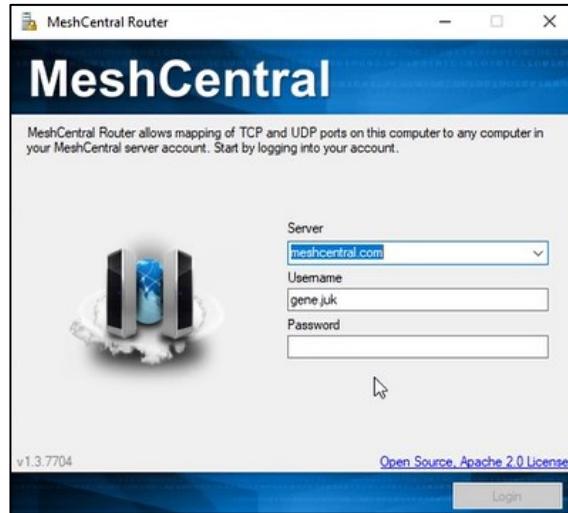
2. From the Tools Downloads section, click on the Win32 MeshCentral Router link (assuming you are using Windows based PC)

Figure 39: Tools Downloads



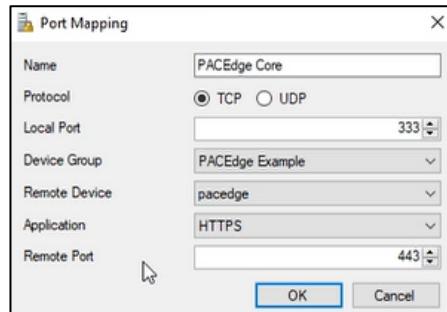
3. Save MeshCentral Router locally on your PC and execute it
4. Log in with your MeshCentral credentials:

Figure 40: Log in



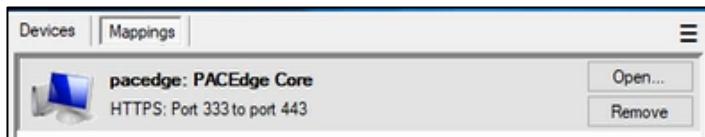
5. Once logged in, make sure you can see your PACEdge device under the **Devices** tab
6. Click on the **Mappings** tab
7. Click on **Add Map** button
8. Fill in details by choosing a port number that will be used locally (on your PC, use a large number 300-1000 range), selecting your remote PACEdge device, selecting HTTPS, and keeping the Remote Port number 443.

Figure 41: Port Mapping



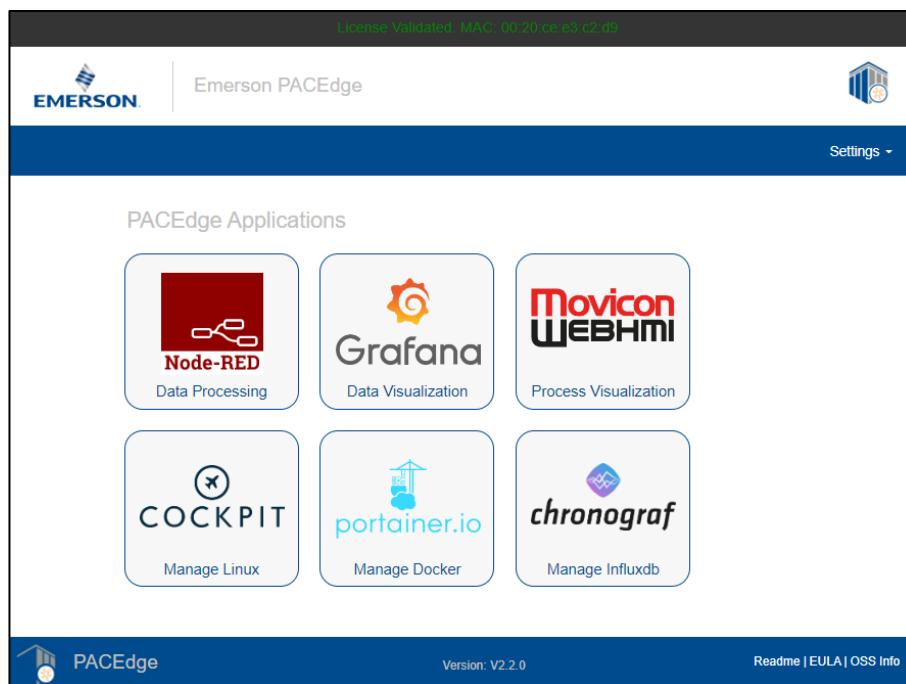
9. Click the OK button and then click on the Open button in the dialogue below:

Figure 42: Mappings



10. A new browser window will open, giving you access to the PACEEdge landing page

Figure 43: PACEEdge Landing Page

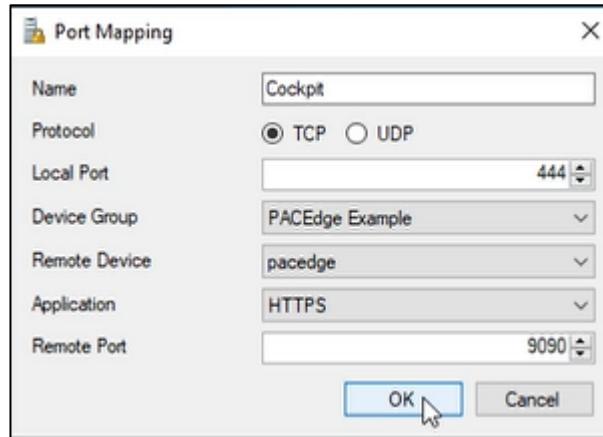


11. Now you can access Node-RED, Grafana, Portainer, and Chronograf

Note: access to Cockpit via the link will not work because Cockpit requires using port number 9090. The following section will describe how to gain access to Cockpit.

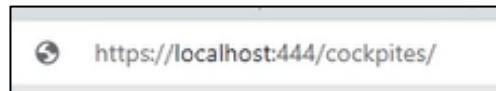
12. To gain access to Cockpit, create a second Port Mapping and make sure that Remote Port is manually set to 9090:

Figure 44: Port Mapping



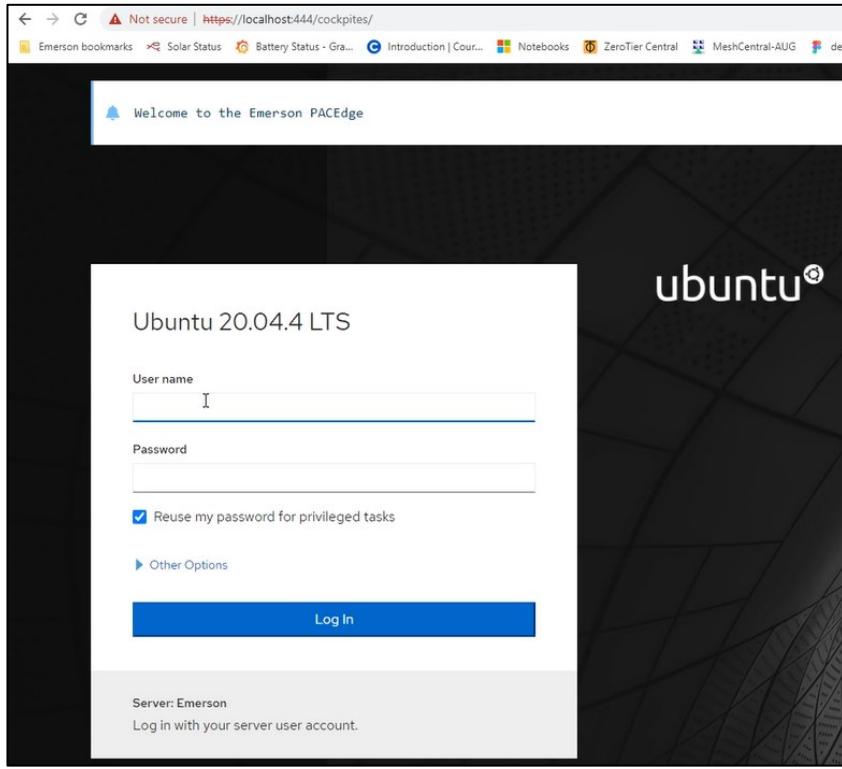
13. Click on the Open button for this new Port Mapping, and when you receive an error page “Not Found,” modify the address in the browser by adding /cockpites/ at the end:

Figure 45: Localhost



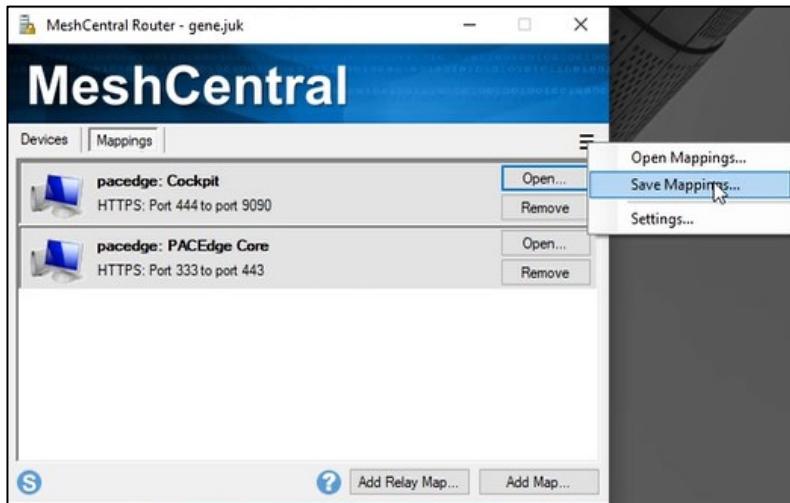
14. Hit enter, and you will have access to the Cockpit on your remote PACEEdge device

Figure 46: Access Cockpit on Remote PACEEdge Device



15. To save the newly created Port Mappings, click on the three lines in the upper right corner and choose Save Mappings:

Figure 47: Save Mappings



16. Next time MeshCentral Router is started, select Open Mappings to restore your Port Mapping.

Setup Private MeshCentral Server (Optional)

A Public MeshCentral server is suitable for experimentation, but it is highly recommended to set up your private MeshCentral server in a production environment. This can be quickly done by using a compute instance on, say, AWS cloud (EC2) or similar. Detailed step-by-step instructions are provided in MeshCentral Install Guide on the MeshCentral website.

Figure 48: Documentation



4.3.3 Remote Access using ZeroTier

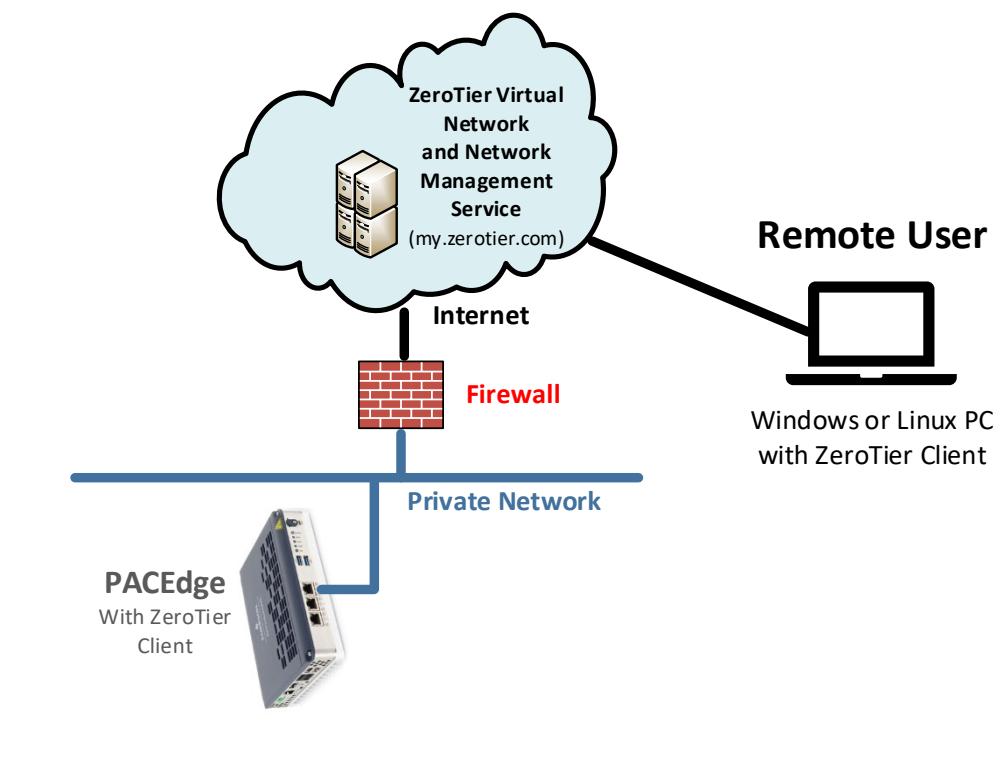
ZeroTier is a commercial service offering allowing the user to create a virtual network that spans different physical sites and devices behind NAT and firewalls. This can connect a user's PC to one or more remote PACEdge devices as if plugged into the same local physical network.

Although ZeroTier is a commercial offering, the Basic version, limited to 25 nodes, is free of charge and is well suited for testing.

To setup ZeroTier following steps need to be executed:

1. Create a ZeroTier account
2. Download and install ZeroTier onto your PC
3. ZeroTier is already pre-installed on PACEdge version 2.2 or later devices but deactivated by default.
4. Configure devices to join the same virtual network

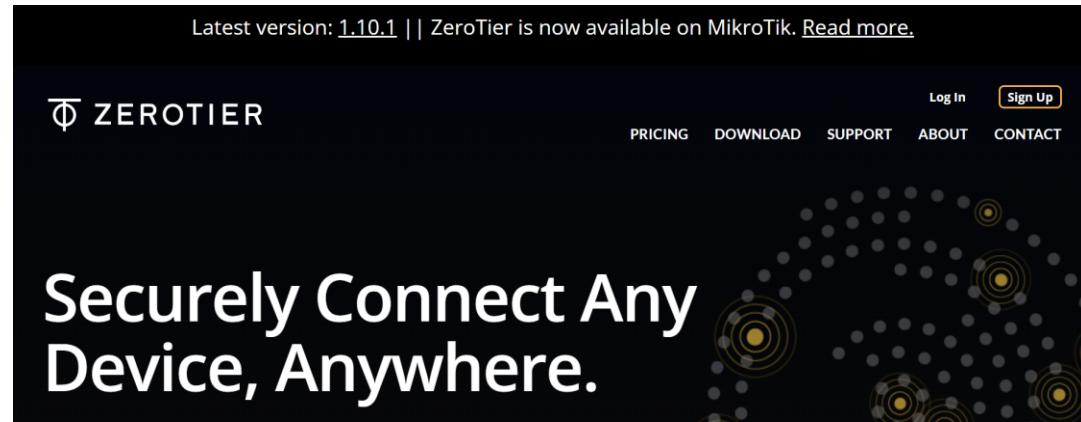
Figure 49 Remote Access using ZeroTier Software



Create ZeroTier account

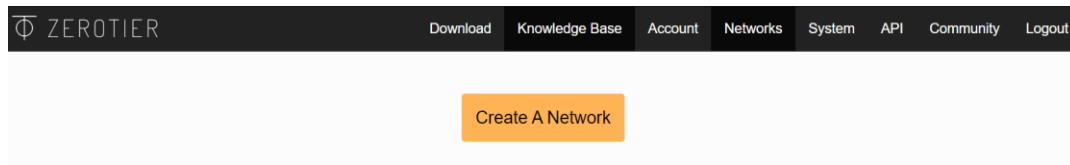
1. Using a browser on your PC, go to www.zerotier.com and click on **Sign Up**

Figure 50: ZeroTier Account



2. Once signed up, go to my.zerotier.com and log in
3. Click on Create a Network button

Figure 51: Create a Network



4. If you want to edit the network name or other properties, click on the created new network.
5. Copy the **Network ID** and keep it handy for the next steps

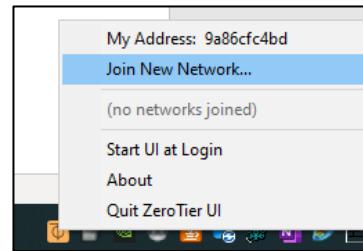
Figure 52: Network ID

NETWORK ID	NAME↑	DESCRIPTION	SUBNET	NODES	CREATED
93afae59630d9b06	PACEdge_zerotier_1		172.27.0.0/16	0	2022-08-18

Install ZeroTier on PC

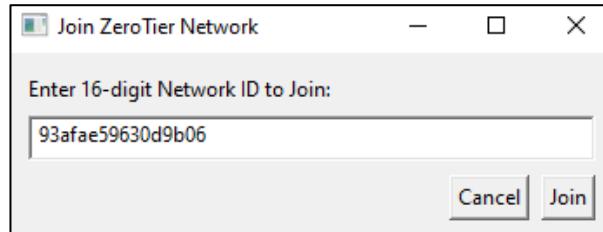
1. Go to www.zerotier.com/download/, select your operating system, download and install ZeroTier
2. Launch ZeroTier on your PC
Note: ZeroTier service might appear as one of the icons in the lower right corner of Windows. In such case, right-click on the ZeroTier icon and select Join New Network

Figure 53: Join New Network



-
3. Enter **Network ID** that was copied in the previous steps and click on **Join**

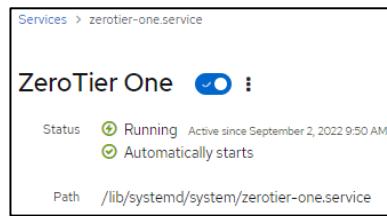
Figure 54: Enter Network ID



Configure PACEEdge Device to Join ZeroTier Network

1. By default, the ZeroTier service is installed in PACEEdge but is disabled. To enable it, go to Cockpit, Services tab, scroll down to zerotier-one service, and click on it.
2. Click on the button to **Start and Enable**. The controller will turn blue, and the status will show Running

Figure 55: ZeroTier Enabled Toggle



3. Copy **Network ID**, steps above, go to Cockpit, Terminal, and type the following:
`sudo zerotier-CLI join Network_ID`
Where **Network_ID** is a 16-digit number from the steps above, enter the password if requested. The result will be: **join OK**

Figure 56: Join OK

```
admin@pacedge-e3c228:~$ sudo zerotier-cli join 93afae59630d9b06
200 join OK
```

Configure and Start Virtual Network

1. Log into your account on my.zerotier.com site
2. Click on the network

Figure 57: Click on Network

Your Networks		SEARCH 1 networks...					
Networks: 1		NETWORK ID	NAME↑	DESCRIPTION	SUBNET	NODES	CREATED
93afae59630d9b06	PACEEdge_zerotier_1	192.168.192.0/24	2	2022-08-18			

3. Scroll down to where both devices (your PC and PACEEdge) are listed with the option to Authenticate access to the network

Figure 58: Locate Devices

Auth?	Address	Name/Description	Managed IPs	Last Seen	Version	Physical IP
<input checked="" type="checkbox"/>	6249788735 00:02:44:12:04:3b	(short-name) (description)	<input checked="" type="checkbox"/> 192.168.192.10 <input type="checkbox"/> 192.168.192.x	ONLINE	1.10.1	84.158.246.137
<input checked="" type="checkbox"/>	9a86cf4bd 00:02:44:12:04:3b	(short-name) (description)	<input checked="" type="checkbox"/> 192.168.192.20 <input type="checkbox"/> 192.168.192.x	ONLINE	1.10.1	84.158.246.137

4. Check **Auth?** Box for each device or optionally assign a specific IP address for each device. **Last Seen** status will be showing: **ONLINE**

Accessing PACEEdge Remotely

1. From your PC, open a browser and type in the IP address assigned in the step above. In this particular example, it would be: 192.168.192.10
2. The PACEEdge landing page will open with access to all PACEEdge applications and Cockpit.

4.3.4

Remote Access using WireGuard

WireGuard is another VPN software that can establish a secure point-to-point connection between PACEEdge and remote computers. WireGuard is a relatively simple and easy setup service, which does not require an external Server as a rendezvous point; however, it requires that the desired network port be opened in the exterior router or firewall, which gates access to the internal network where PACEEdge is located.

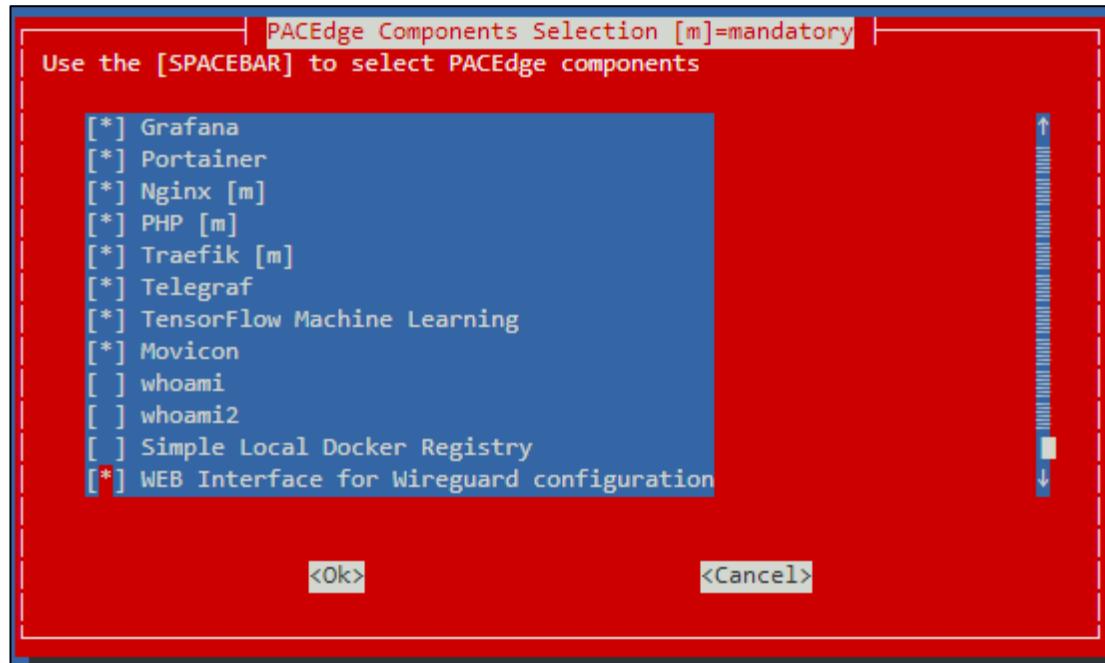
PACEEdge comes pre-installed with WireGuard service and an optional graphical WireGuard UI interface, which runs in its container. WireGuard UI container is disabled by default but can be enabled using the config-compose script. To enable the WireGuard UI container, please do the following:

1. Open Cockpit and go to Terminal
2. Type: cd /home/admin/pacedge
3. Type: ./config-compose.sh

Select **Yes** and hit **Enter** key to shut down PACEEdge containers

In the list of available containers, using **arrow** keys, scroll down, and using the **space** key, select: **WEB Interface for Wireguard configuration**

Figure 59: Select WEB Interface for Wireguard Configuration



4. Hit the **Tab** key to select OK and then enter; respond with enter key to start the PACEEdge again.
5. Once all containers are up and running again, you can access WireGuard graphical configuration utility by entering the PACEEdge IP address, followed by port number 5005, for example:
http://192.168.3.100:5005
user: admin
password: edgestack
6. For details on how to configure WireGuard service on both ends, please consult online documentation at www.wireguard.com
7. **NOTE:** For Cybersecurity reasons and CPU resource optimization, once the WireGuard configuration is finished, please remove the WireGuard UI container by following the same steps as above, only this time removing the check mark in WEB Interface for Wireguard configuration.

4.3.5 Remote Access using OpenVPN

Starting with version 2.2.0, PACEdge comes with pre-installed OpenVPN software.

OpenVPN software can be used to create secure point-to-point connections. PACEdge can be configured to act as both OpenVPN Server and a Client. If the user desires to use the OpenVPN service, it must be properly configured and started.

Note: if configured as Server, a network port in the router/firewall will have to be opened so that the external Client can reach PACEdge and establish a secure tunnel.

For OpenVPN configuration and use details, please consult the following:

- github.com/OpenVPN/openvpn

openvpn.net (commercial offering based on OpenVPN)

Section 5: PACEEdge Usage Examples

5.1 Node-RED Getting Started

5.1.1 Example Description

This example shows how to create a basic Node-RED flow that generates sine values and displays them via Node-RED Dashboard

5.1.2 Prerequisites

None

5.1.3 Example Walkthrough

Getting Started with Node-RED

1. Open the Node-RED application from the list of Nodes on the left side.
2. Drag and drop the **inject** node and **debug** node into the middle of the page (flow editor).

Figure 60:Inject Node and Debug Node



3. Connect **inject** node to **debug** node by dragging the line between grey connection points.
4. Open the debug window by clicking on the bug symbol  in the upper-right corner.
5. Click on the red **deploy** button in the upper-right corner
6. At this point, the flow has been deployed and is active. Clicking on the square icon in front of the **inject** node will generate a time stamp and will send it to **debug** node, which will display it in debug window, as shown below:

Figure 61: Timestamp



7. Double-click the **inject** node, and you will get a configuration dialogue where you can change the functionality of the **inject** node. For instance, you can send a string or JSON structure instead of sending the time stamp. For example, let us configure a repeat mode by selecting **interval** and setting it to every 1 sec. Click on **Done**.
8. Click on **Deploy** after making changes so that they take effect.
9. In the debug window, you will get a new timestamp every second

Adding a Function Node

1. Drag and drop a function node and put it in between the **inject** and **debug** nodes. Connect all three nodes in a series. The message the inject node sends will go to the function node and then **debug** node.

Figure 62: Place the Function Node Between the Timestamp and Debug Nodes



2. Double-click on the function node and write the following JavaScript code:

Figure 63: JavaScript Code

```
1 var a = msg.payload;
2 msg.payload = Math.sin(a/1000/10);
3 return msg;
```

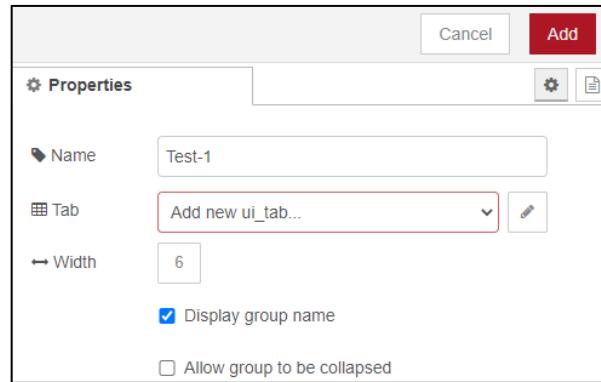
This code will first assign the received timestamp to **variable a**, then scale it down and calculate its Sine value. The result will be sent out to the next node.

3. Once deployed, you will see a value slowly changing between -1 and 1 in the debug window.

Adding Node-RED Dashboard

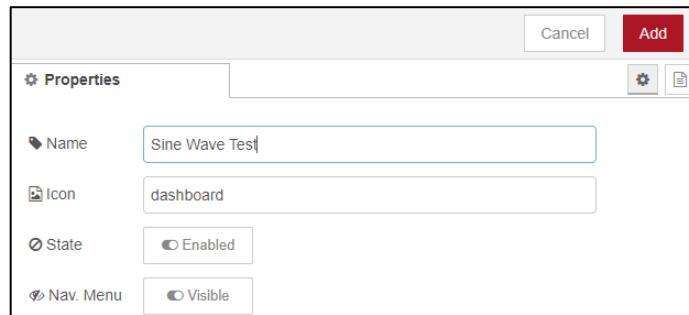
1. Drag and drop a **chart** node and connect it to the function node.
2. Double-click the chart node and make the following changes:
 - a. In the Group, select **Add new ui_group** and click on the pen symbol to edit.
 - b. In the Properties Window, provide a name (such as "Test-1") in the Name field, and in the Tab field, select **Add new ui_tab** and click on the pen symbol to edit.

Figure 64: Properties



-
- c. In the new dialogue, give a name, such as **Sine Wave Test**, and click on the red **Add** button.

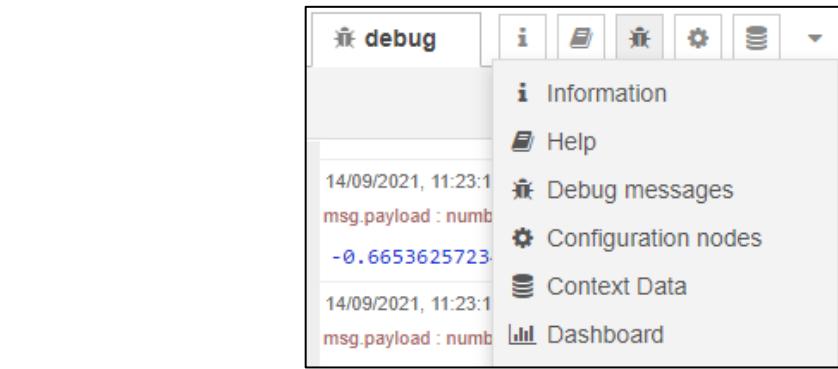
Figure 65: Sine Wave Test



- d. Click on **Add** button again
3. Back in the Edit chart dialogue, in X-axis, select the **last 1 minute** so that the data for the previous 1 minute is visible.

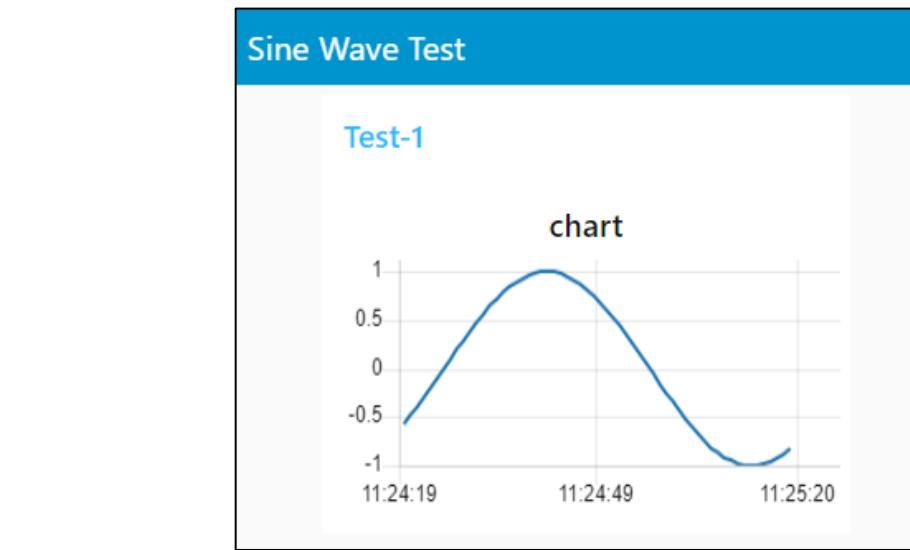
4. Give the chart a name and click **Done**.
5. Click on **Deploy** to save the changes.
6. Open the Dashboard window by clicking on the drop-down symbol in the upper-right corner and select **Dashboard**.

Figure 66: Dashboard Menu



- a. Click on the square with the arrow  symbol.
- b. A new browser tab will open, and a sine wave will display:

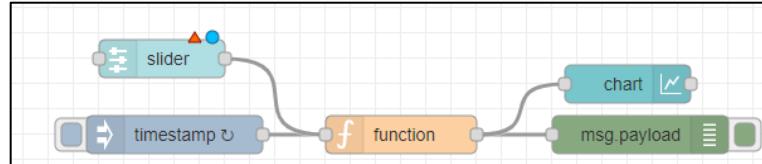
Figure 67: Sine Wave Test



Adding Control to the Node-RED Dashboard

1. Add a slider control that can increase or decrease the sine wave frequency. To do so, drag and drop a **slider** node and connect it so that the output from the slider goes into the **function** node (in parallel with the **inject** node).

Figure 68: Slider Node



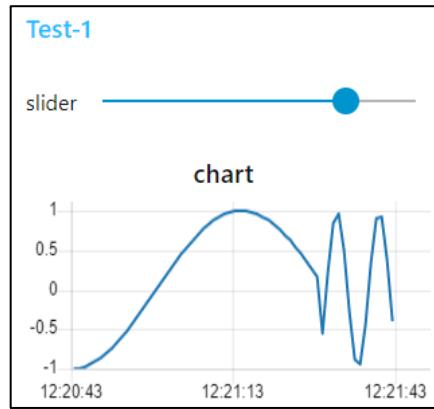
2. Double-click the **slider** node and select the group created earlier [Sine Wave Test]Test-1.
3. Note the Range. Set it to 1 to 10 in steps of 1.
4. In the Topic field, from the drop-down list, select String and then type in: **Freq**.
5. Click on **Done** to close the **slider** node.
6. Now we have a problem: two different messages are coming into the **function** node. We need to change the code to check for the message topic and take action accordingly. Change the code inside the function node as follows:

Figure 69: Function Node's Code Change

Setup	On Start	On Message	On Stop
		<pre> 1 // initialize retentive variable 2 context.data = context.data {}; 3 if(context.data.freq == null) context.data.freq = 1; 4 5 // check if message has topic Freq and then assign the value to 6 // persistent variable. Other messages are from inject function 7 // and should generate new sinus value 8 if(msg.topic == "Freq"){ 9 context.data.freq = msg.payload; 10 }else{ 11 var a = msg.payload; 12 msg.payload = Math.sin(a/1000/10*context.data.freq); 13 return msg; 14 } 15 return; </pre>	

7. Once deployed, check the dashboard and use the slider to change the multiplier inside the Sinus function. As a result, you will get a sine wave of different frequency

Figure 70: Changed Sinus Value



5.2 Using the InfluxDB Database

Starting with PACEdge version 2.2, new Emerson-specific Node-RED nodes have been introduced that simplify access to the InfluxDB database. These nodes are not a replacement for the generic InfluxDB nodes and have only a subset of functionality, but they are pre-configured and easier to use for basic tasks. The sections below will explain how to use both Emerson-customized InfluxDB nodes and standard InfluxDB nodes.

Note: to get Emerson customized InfluxDB nodes added to the Node-RED node palette, import an example **PacedgeInfluxDB**, as described below.

5.2.1 Using Emerson Customized InfluxDB Nodes

Example Description

This example shows how to access the InfluxDB database within the Node-RED flow using Emerson-customized Node-RED nodes. It shows how to install the nodes, create a new database, write to it, and read from it.

Prerequisites

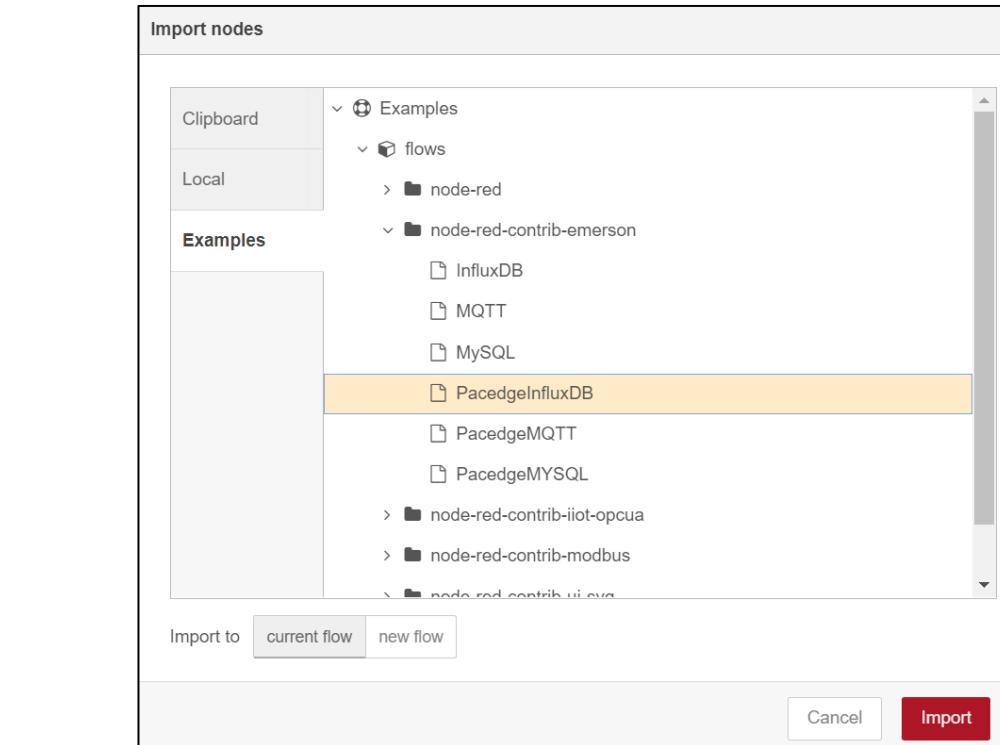
PACEdge version 2.2 or newer.

If new to Node-RED, completion of the 5.1 is highly recommended, as it shows some Node-RED usage basics and tips

Example Walkthrough

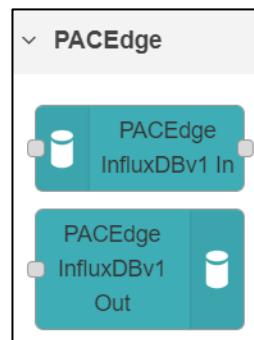
1. In the Node-RED, upper right corner, click on the three-line symbol  and select Import.
2. In the Import Nodes dialog that opens, go to Examples, expand **node-red-contrib-emerson** and, select **PacedgeInfluxDB**, click on the Import button.

Figure 71: Select Import



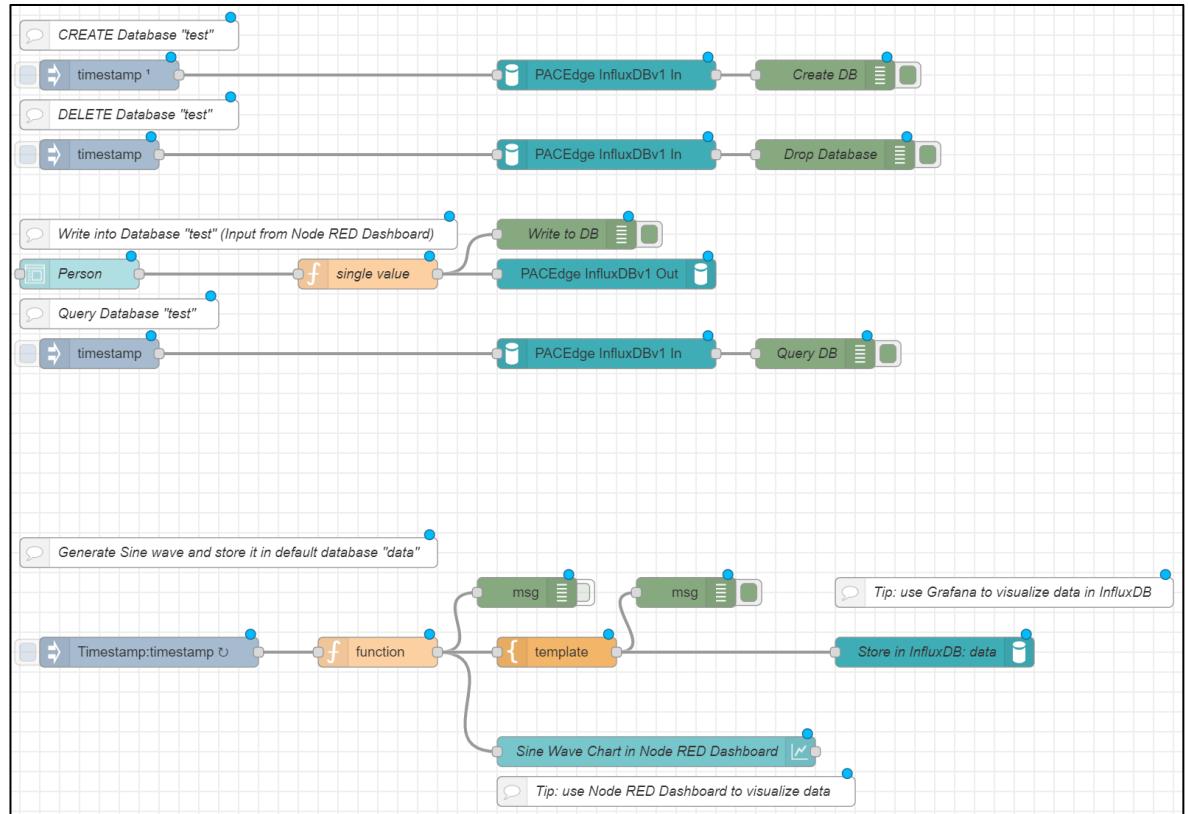
3. Now you have two new nodes in the node pallet on the left side.

Figure 72: Example Nodes



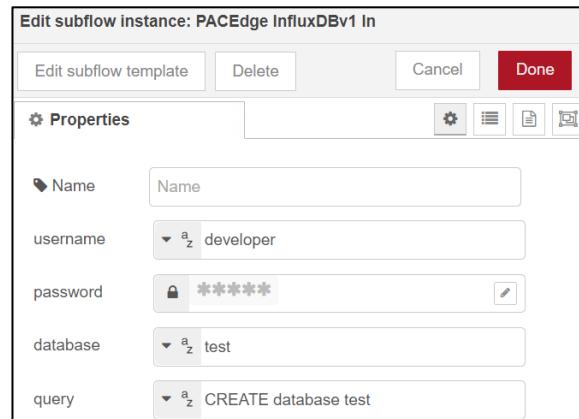
The following is an example flow showing how to write to InfluxDB and read from InfluxDB.

Figure 73: Example flow



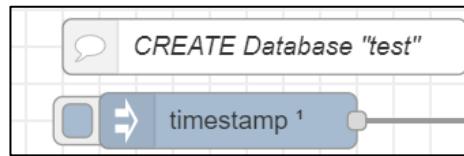
- For each instance of the PACEEdge InfluxDB node, configure username and password. You can use **admin**, **developer**, or **service** users.

Figure 74: Configure Users



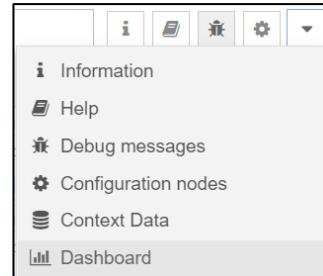
5. Once finished, click on the Deploy button 
6. At this point, the example flow is active. It consists of two parts:
 - a. The bottom part generates and stores a new value every 200msec, representing the sine wave. Values will be stored in the InfluxDB database called **data**, which is present by default. You can observe it in Node-RED Dashboard and via Grafana.
 - b. The upper part shows how to create a new database in InfluxDB called **test**, write data into it, query the data, and delete the database. Data entry will be done via Node-RED Dashboard, as explained below
Make sure to first create a test database by clicking on the button

Figure 75: Create Database



7. Open Node-RED Dashboard by clicking on the triangle icon in the upper right corner, then selecting Dashboard.

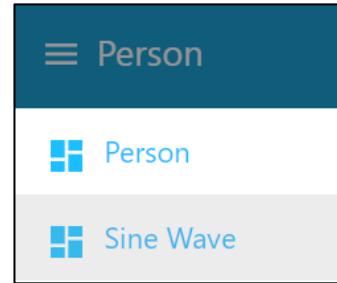
Figure 76: Dashboard



8. In the right-side pane that opens, click on the icon  to open another browser tab that displays Node-RED Dashboard

9. In the Node-RED Dashboard, you will have two tabs accessible on the upper left side of the window:

Figure 77: Person Tab



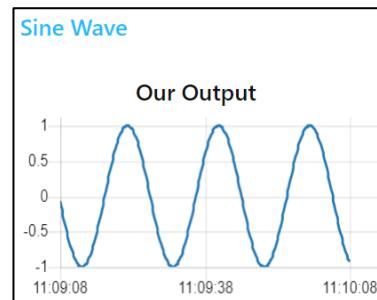
You will default see the **Person** tab, which allows you to enter data into the InfluxDB database called **test** (upper portion of the Node-RED flow).

Figure 78: Person Tab

A screenshot of a "Person" form. The title "Person" is at the top. Below it are three input fields: "Last Name *", "First Name *", and "Birthday *". The "Birthday" field includes a calendar icon. At the bottom are two blue buttons: "SUBMIT" on the left and "CANCEL" on the right.

Once you click on the **Sine Wave** tab, you will see the sine wave that is being generated and stored in the InfluxDB database called **data**

Figure 79: Data



10. Once you enter one or more personal data using the **Person** tab, you can go back to the Node-RED flow browser tab and click on the button to Query database.

Note that you must first create a database test; otherwise, you will get **Error 404 Not Found**.

Note: In the Node-RED flow tab, please select Debug tab / so you can see query results and errors, if any.

5.2.2 Using Node-RED Standard InfluxDB Nodes

Example Description

This example shows how to access the InfluxDB database within the Node-RED flow using Node-RED standard InfluxDB nodes. It shows how to create a new database, write to it, and read from it.

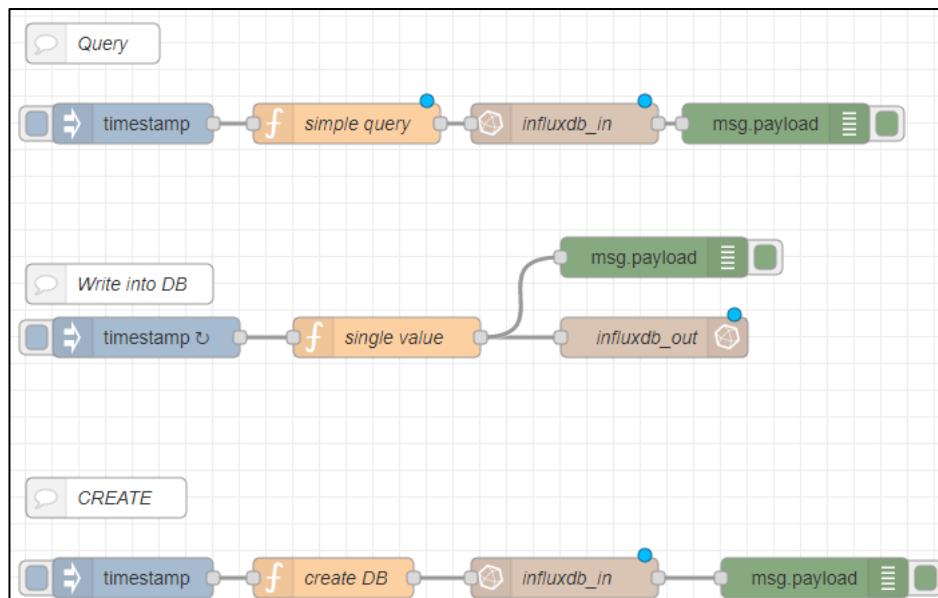
Prerequisites

If new to Node-RED, completion of the 5.1 is highly recommended, as it shows some Node-RED usage basics and tips

Example Walkthrough

In the Node-RED flow window place **inject**, **function**, **influxdb_in**, **influxdb_out**, and **debug** nodes and interconnect them as shown. Note that the example below also uses **Comment** nodes to make it more readable.

Figure 80: Example Walkthrough



1. Double-click on the function node in the Query section and enter the following code. It will be used to query the database:

Figure 81: Query Section: Function Node's Code

Setup	On Start	On Message	On Stop
<pre>1 msg.query="select * from demo_1;"; 2 return msg;</pre>			

2. Double-click on the function node in the Write into DB section and enter the following code. It will generate random values and send them to be stored in the database.

Figure 82: DB Section: Function Node's Code

Setup	On Start	On Message	On Stop
<pre>1 msg.payload = Math.random()*10; 2 return msg;</pre>			

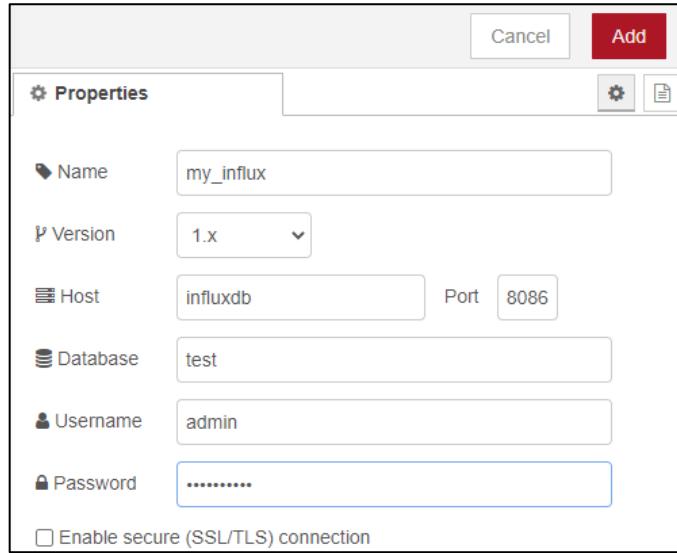
3. Double-click on the function node in the Create section and enter the following code. It will be used to create a new database within InfluxDB.

Figure 83: Create Section: Function Node's Code

Setup	On Start	On Message	On Stop
<pre>1 msg.query="CREATE database test;"; 2 return msg;</pre>			

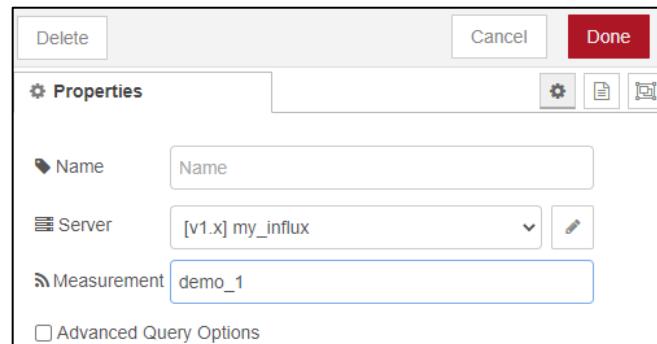
4. Now, the first instance of the InfluxDB node (in the Query section) needs to be configured as follows:
 - In the Server field, select **Add new influxdb** and click on the pen symbol to edit.
 - In the Name, give the desired name, such as **my_influx**.
 - In Host, enter: **influxdb** and keep the port number as 8086.
 - In the Database, enter the **test**.
 - In Username, enter either **admin** or **developer** or **service**.
 - In the Password field, enter the password for the user that was entered above.

Figure 84: User Properties



- g. Click on **Add**, then on **Done**.
5. Double-click on the influx_out node in the Write into DB section. In the Server drop-down list, you should now see the newly configured database server with the name my_influx; select it since this is an output node that will write data into the database; we also need to provide the name for the Measurement. Please enter the desired name, such as **demo_1**. Click **Done** afterward.

Figure 85: Demo_1



6. In the third **influxdb** node, select **my_influx** in the list of servers and click **Done**.
7. Deploy the newly configured flow.
8. Now you can experiment by triggering actions when clicking on the trigger nodes. You will notice that if you try to read or write to the database, you will get an error stating that the database does not exist. Once you trigger the flow to create a database, you can write to and read from the database.

9. Note that the database is referenced by name in our flow tests. This name is entered in the InfluxDB nodes Database field, but it must also match in the function node create DB. Therefore, if you want to change the database name, you need to change it in the create database flow function node.
10. If you want to modify query parameters, edit the function node.

5.3 Using MySQL Database

Starting with PACEEdge version 2.2, new Emerson-specific Node-RED nodes have been introduced that simplify access to MySQL database. These nodes are not a replacement for the generic MySQL nodes and have only a subset of functionality, but they are pre-configured and easier to use for basic tasks. The sections below will explain how to use both Emerson customized MySQL nodes and standard MySQL nodes.

Note: to get Emerson customized MySQL nodes added to the Node-RED node palette, import an example **PaceedgeMySQL**, as described below.

5.3.1 Using Emerson Customized MySQL Nodes

Example Description

This example shows how to access MySQL Database from Node-RED.

Prerequisites

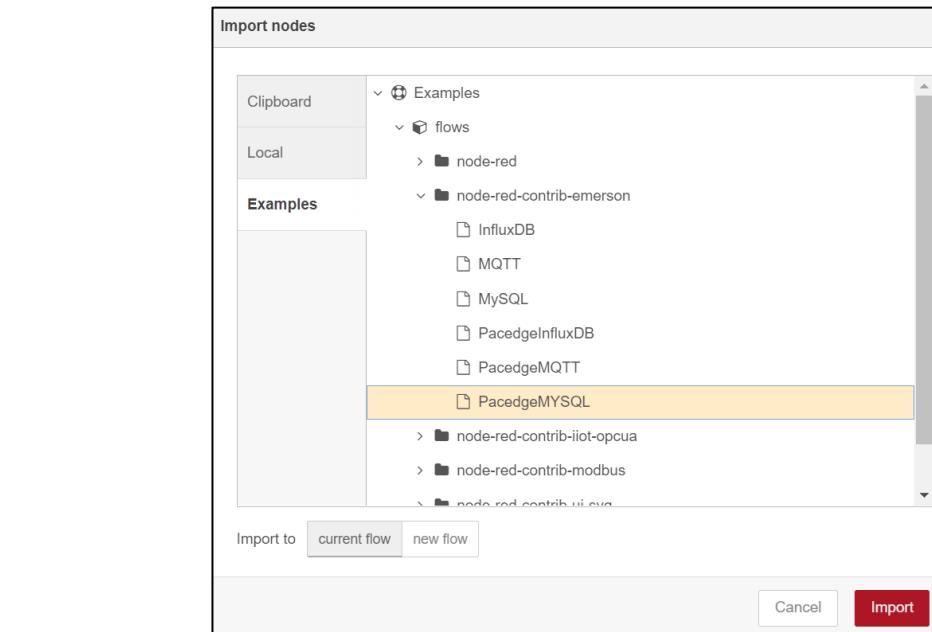
PACEEdge version 2.2 or newer.

Completing the 5.1 Getting Started with Node-RED is highly recommended if new to Node-RED, as it shows some Node-RED usage basics and tips.

Example Walkthrough

1. In the Node-RED, upper right corner, click on the three-line symbol  and select Import.
2. In the Import Nodes dialog that opens, go to Examples, expand **node-red-contrib-emerson** and, select **PacedgeMySQL**, click on the **Import** button.

Figure 86: Import Nodes

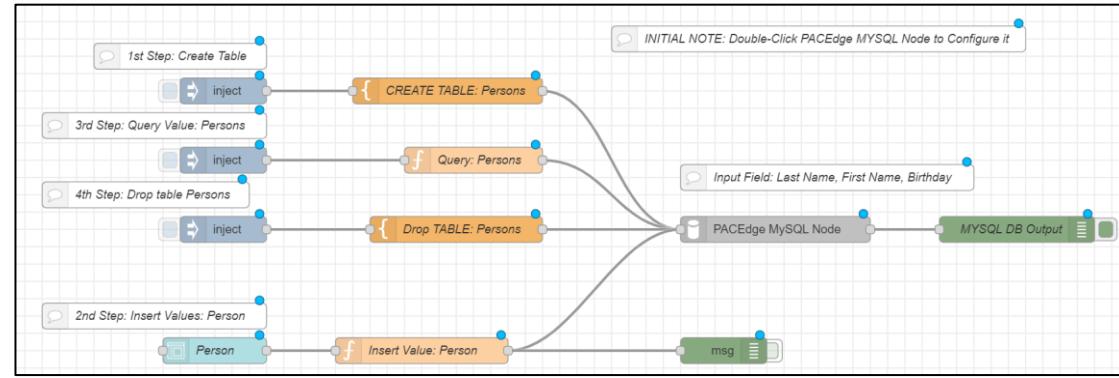


- Now you have one node in the node pallet on the left side



And an example flow that shows how to write to MySQL and how to read from MySQL

Figure 87: MySQL Example



- Double click on PACEEdge MySQL node and configure username and password.
You can use **admin**, **developer**, or **service** users.

Figure 88: PACEEdge MySQL node

Edit subflow instance: PACEEdge MySQL Node

Edit subflow template	Delete	Cancel	Done
Properties			
Name	Name		
username	admin		
password		
database	data		

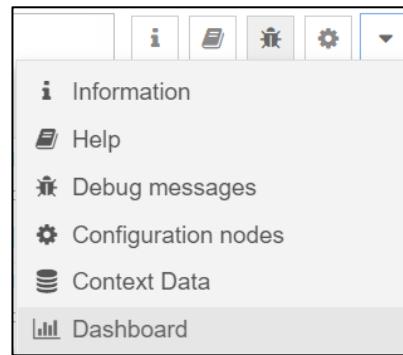
- Once finished, click on Deploy button
- At this point, the example flow is active. In the flow, start by creating a new table in the MySQL database. To do so, click on Inject button

Figure 89: Inject Button



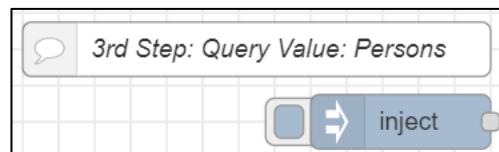
7. Try entering some data into the MySQL table. In the example, this is done by using Node-RED Dashboard. To open Dashboard, first, click on the triangle icon in the upper right corner, then select Dashboard

Figure 90: Node-Red Dashboard



8. In the right-side pane that opens, click on the icon  to open another browser tab that displays Node-RED Dashboard
9. In the Dashboard, please enter one or more person details
Go back to the browser tab with Node-RED flow and click on the button to Query the database.

Figure 91: Query Database



Note: In the Node-RED flow tab, please select Debug tab / so you can see query results and errors, if any.

10. Feel free also to try deleting the MySQL table

5.3.2 Using Node-RED Standard MySQL Nodes

Example Description

This example shows how to access MySQL Database from Node-RED using standard MySQL nodes.

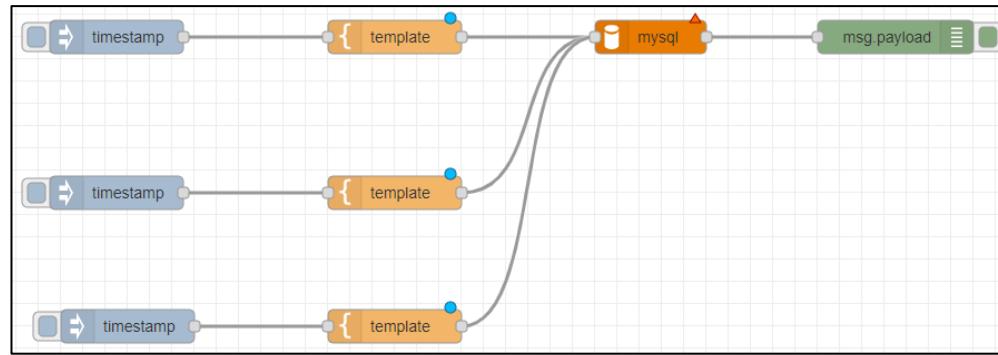
Prerequisites

Reading section 5.1, Getting Started with Node-RED, is highly recommended if new to Node-RED. It shows some Node-RED usage basics and tips.

Example Walkthrough

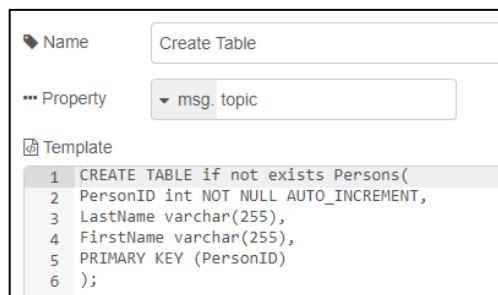
We will use a template node instead of a function node in this example. This is to show that there are alternative ways to implement the same functionality. Drag and drop inject, template, MySQL, and debug nodes and interconnect them as shown. In contrast to InfluxDB nodes, for MySQL, the same node is being used for both writing and querying the database, so it is more efficient to instantiate it only once and just send different commands.

Figure 92: Walkthrough Example



1. Double-click the first **template** node, name it **Create Table** in the Property field, type in `msg.topic`, and paste the following code into it:

Figure 93: Template Node



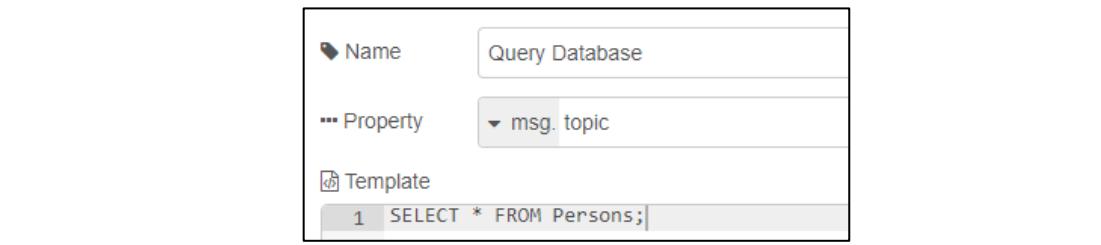
2. Double-click the second **template** node, name it **Write to Database**, enter msg.topic, and fill in the following code.

Figure 94: Write to Database Node



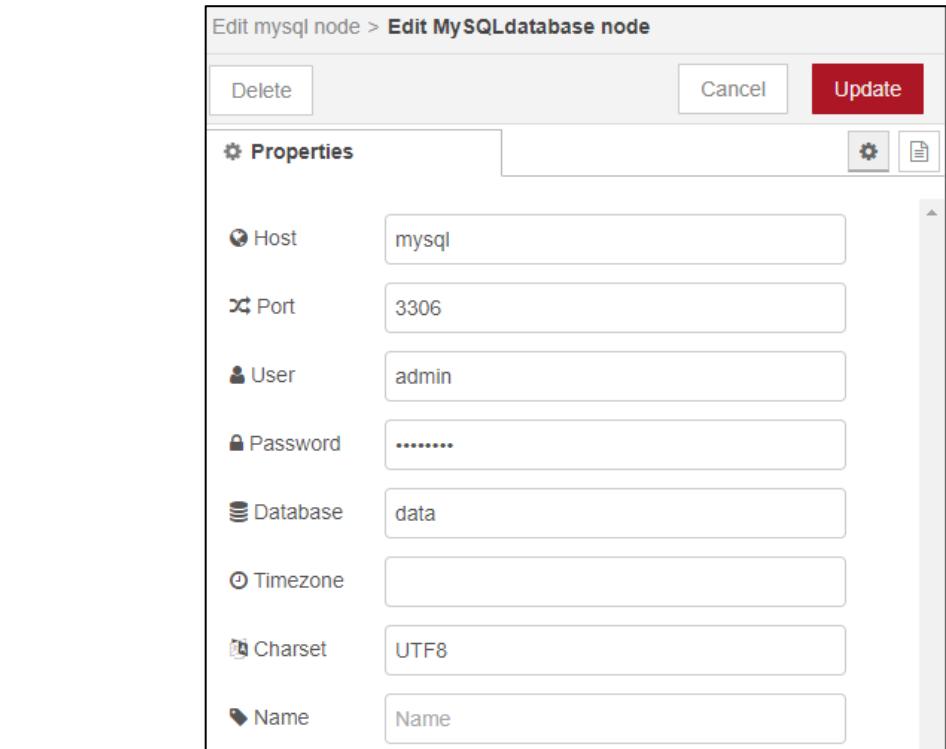
3. Double-click the third **template** node, name it **Write to Database**, enter msg.topic, and fill in the following code.

Figure 95: Write to Database Node



4. Double-click the MySQL node in the Database, select Add new MySQL database and click on the pen icon to edit.
5. In the Host field, enter **mysql**. In the User field, enter either **admin** or **developer**. In the Password field, enter the associated password with the user you choose and the Database field value data. Click on **Update** and then on **Done**.

Figure 96: My SQLDatabase Node



6. Once all the edits are done, click on the **Deploy** button. If the MySQL node is properly configured, you will see a green dot with the label OK at the bottom of it.
7. Click on the first **inject** node to create a new table, then on the second **inject** node to enter data, and on the **third** inject node to read all entries from the table.

5.4 Using MQTT Communication

Starting with PACEEdge version 2.2, new Emerson-specific Node-RED nodes have been introduced that simplify MQTT communication. These nodes are not a replacement for the generic MQTT nodes and have only a subset of functionality, but they are pre-configured and easier to use for basic tasks. Sections below will explain how to use both Emerson customized MQTT nodes as well as standard MQTT nodes.

Note: to get Emerson customized MQTT nodes added to the Node-RED node palette, import an example **PaceedgeMQTT**, as described below.

5.4.1 Using Emerson Customized MQTT Nodes

Example Description

This example shows how to use internal and external MQTT brokers and communication when using Emerson-customized MQTT nodes.

Prerequisites

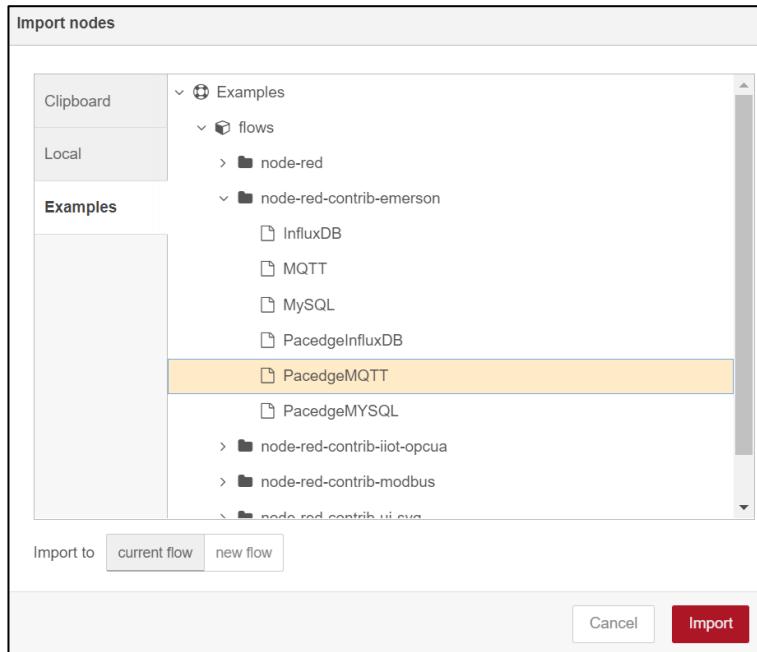
PACEEdge version 2.2 or newer.

Completing section 5.1, *Getting Started with Node-RED* example, is highly recommended. It shows some Node-RED usage basics and tips.

Example Walk Through

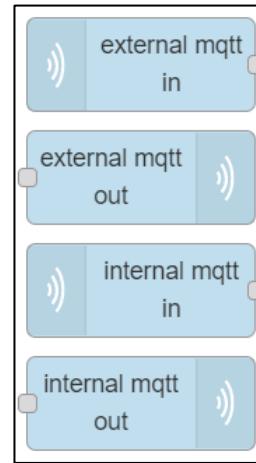
1. In the Node-RED, upper right corner, click on the three-line symbol  and select Import.
2. In the Import Nodes dialog that opens, go to Examples, expand **node-red-contrib-emerson** and, select **PaceedgeMQTT**, click on the Import button.

Figure 97: PACEdge MQTT



3. Now you have four new nodes in the node pallet on the left side and an example flow that shows how to publish and subscribe to internal and external MQTT brokers in MySQL Database

Figure 98: Example Nodes

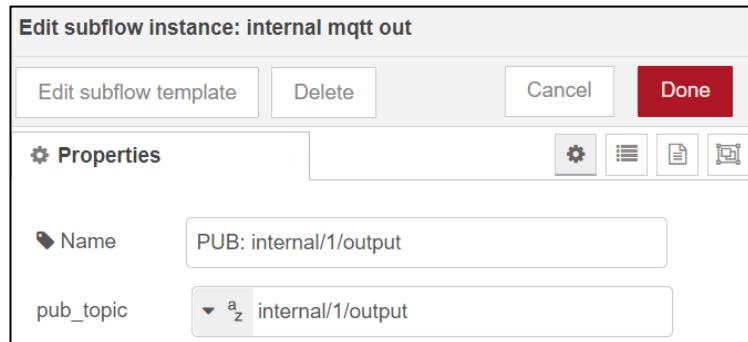


4. Click on Deploy button 
5. At this point, the example flow is active, and you can send a timestamp by clicking on Inject node button while observing the received message in the Debug window



6. Optionally, try changing the MQTT Topic that messages are being published to. To do so, first, click once on the MQTT PUB node (to select a segment) and then double-click to open the properties

Figure 99:pub_topic



Note: if you change the topic messages are being published to, you also need to change the topic you are subscribing to.

5.4.2 Using Node-RED Standard MQTT Nodes

Example Description

This example shows how to use internal and external MQTT brokers and communication when using Node-RED standard MQTT nodes.

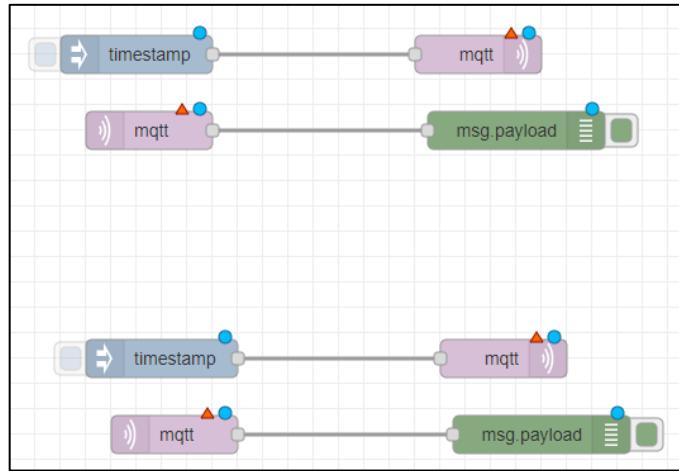
Prerequisites

Completing section 5.1, *Getting Started with Node-RED* example, is highly recommended. It shows some Node-RED usage basics and tips.

Example Walk Through

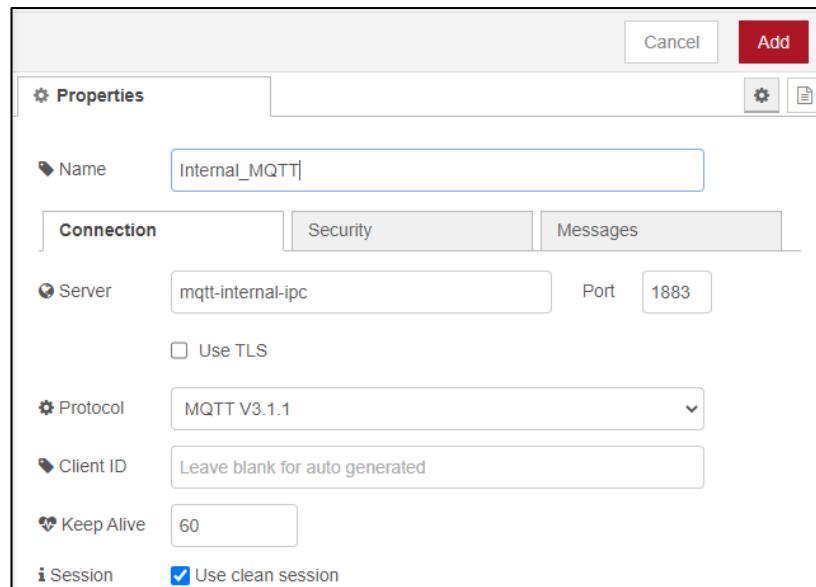
1. Drag and drop **inject**, **MQTT output**, **MQTT input**, and **debug** nodes and interconnect them as shown in the diagram.

Figure 100: MQTT Nodes



2. We will use top nodes for internal MQTT communication. Double-click on the first MQTT output node, and in the Server window, select **Add new MQTT-broker** and click on the pen icon to edit.
3. Enter a name for this broker, such as **Internal_MQTT**, in the Server field, enter **mqtt-internal-ipc**, then click on **Add**.

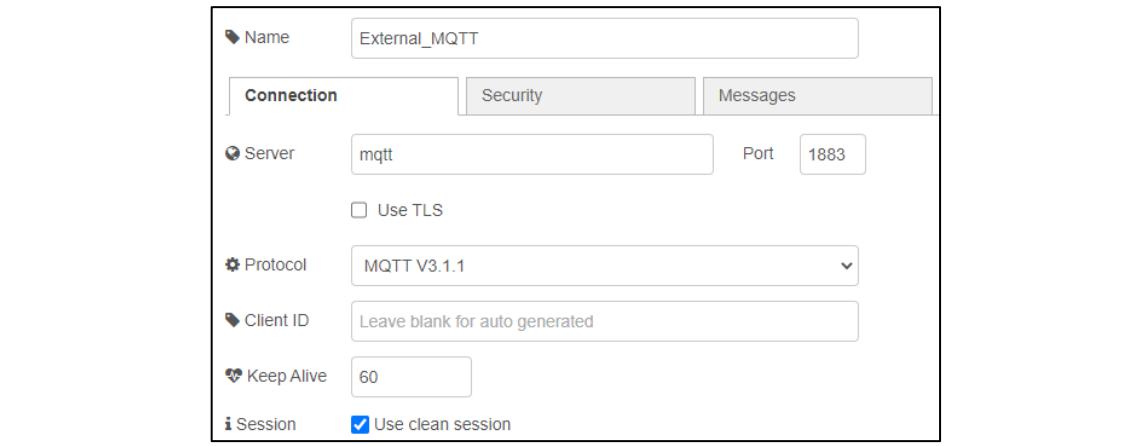
Figure 101: Name of Broker



4. Enter the MQTT Topic name, **Internal_Test**, and click **Done** to close the window.
5. Then, double-click on the first MQTT input node and select the same server created above. Also, enter the same Topic as above, and click **Done**.

6. We will use bottom nodes for external MQTT communication. Double-click on the second **MQTT output** node, and in the Server window, select **Add new MQTT-broker** and click on the pen icon to edit.
7. Enter a name for this broker, such as **External_MQTT**, in the Server field, enter **mqtt**, then click on **Add**.

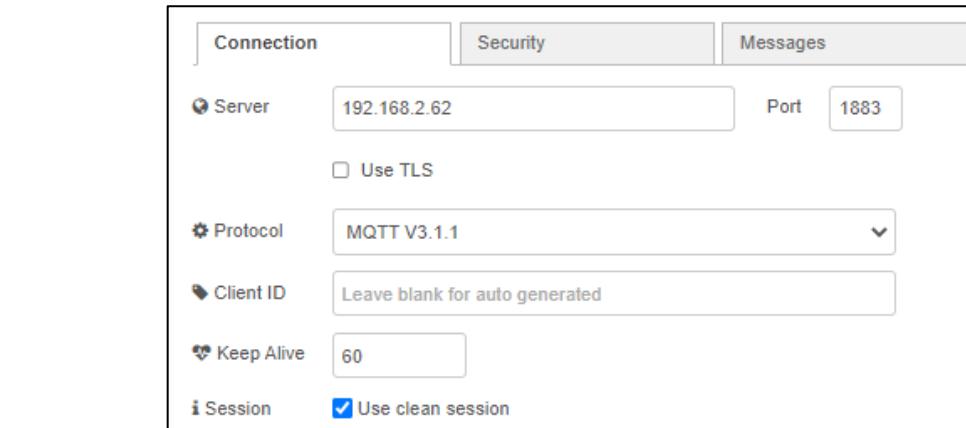
Figure 102: External MQTT Properties



8. Enter the MQTT Topic name, **External_Test**, and click **Done** to close the window.
9. Double-click on the second MQTT input node and select the same external Server that was created above. Also, enter the same Topic as above, and click **Done**.
10. After all the edits are done, click on the **Deploy** button
11. When triggering each flow, clicking on the Inject node will show that a time stamp is generated, published to the respective MQTT Broker, and specified the topic. Then flow below, which connects to the same Broker and subscribes to the same topic, will receive it and print it in the debug window.

Note: The outside of this IPC internal MQTT Broker is not accessible, but an external broker is. If you have another device with MQTT, say another PACEEdge system, then you can connect it to this setup by configuring the MQTT broker, where 192.168.2.62 is an IP address of the first PACEEdge system.

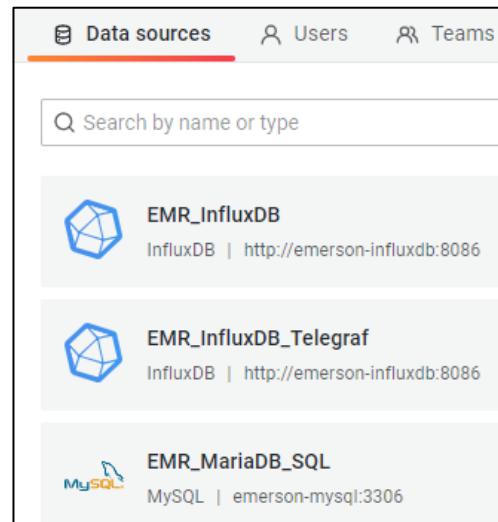
Figure 103: Connection Settings



5.5 Using Grafana

Grafana is a graphical application that lets users quickly visualize the data and perform different analytics. Grafana supports a large number of databases, including InfluxDB and MySQL. Starting with PACEEdge version 2.2, Grafana comes pre-configured with default InfluxDB and MySQL databases **data**, named **EMR_InfluxDB** and **EMR_MariaDB_SQL**, which were used in the previous examples. In addition, by default, PACEEdge is using Telegraf to collect system health data and to store it in InfluxDB database **telegraf_metrics**. This database is also pre-configured in Grafana under the name: **EMR_InfluxDB_Telegraf**.

Figure 104 Pre-configured Databased in Grafana



5.5.1 Example Description

This example will walk through using Grafana to analyze data in pre-configured databases and how to configure Grafana to use the new InfluxDB database.

5.5.2 Prerequisites

PACEEdge version 2.2 or newer.

Data has been written into InfluxDB and MySQL databases using Node-RED examples in the previous section.

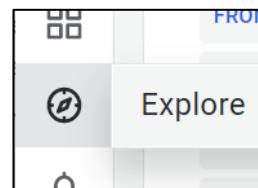
Completing section 5.1, *Getting Started with Node-RED* example, is highly recommended. It shows some Node-RED usage basics and tips.

5.5.3 Example Walk Through

Using Grafana to Analyze the pre-configured InfluxDB database

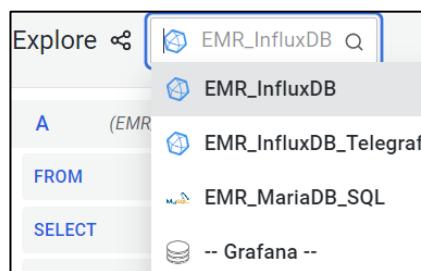
1. Open Grafana and log in as an admin user
2. Click on Explore icon in the upper left corner

Figure 105: Explore Button



3. Select the EMR_InfluxDB database in the drop-down list

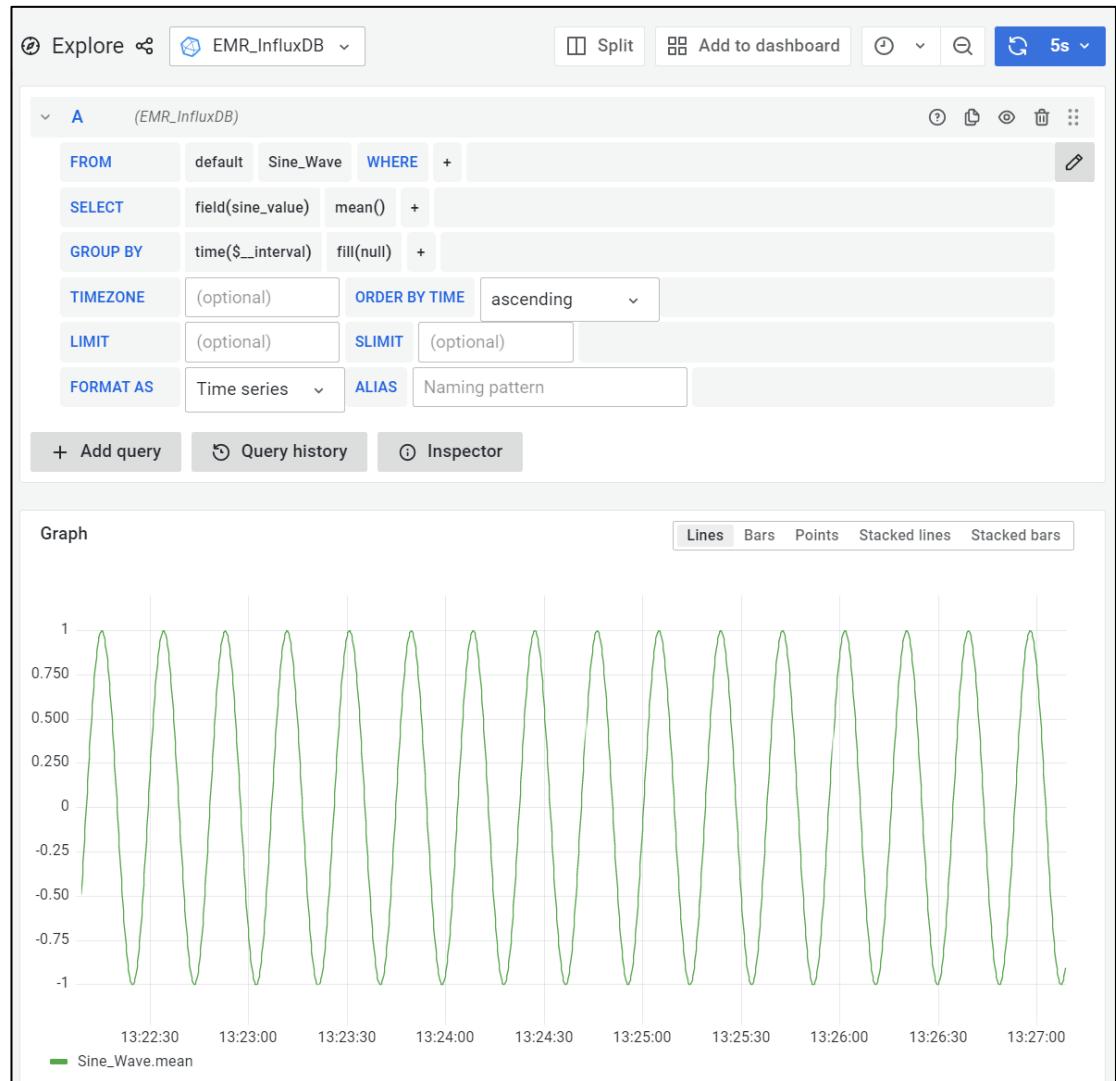
Figure 106: EMR_InfluxDB



4. In the Measurement, select **Sine_Wave**; in Field, select **sine_value**. All other values can be left at default. The measurement will show the sine wave generated by Node-RED and stored in InfluxDB.

Note: Play around by changing the time to the last 5 minutes and setting the query to run every 5 sec.

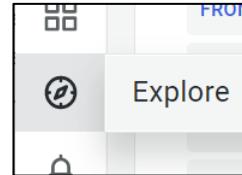
Figure 107: EMR_InfluxDB



Using Grafana to Analyze pre-configured MySQL database

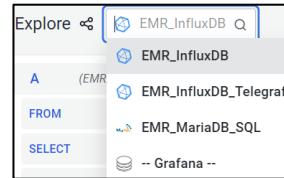
1. Open Grafana and log in as an admin user
2. Click on Explore icon in the upper left corner

Figure 108: Explore Button



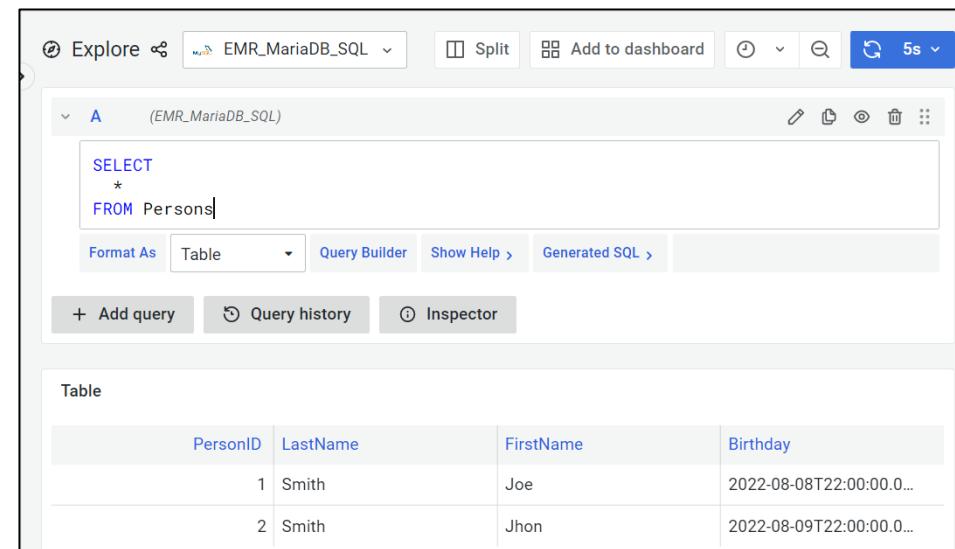
3. Select the EMR_MariaDB_SQL database in the drop-down list

Figure 109: EMR_InfluxDB



4. For this example, switch to text query mode by clicking on the pencil icon  on the right side of the screen.
5. Modify query entry to show all values in the table called Persons and format output data as Table

Figure 110: Format Data as a Table

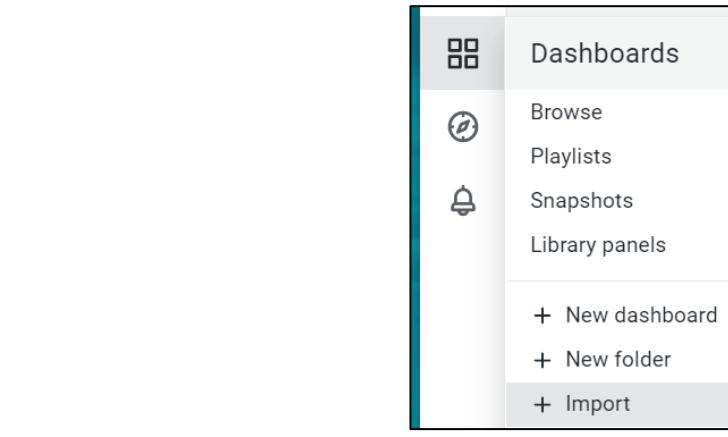
A screenshot of the Grafana Explore screen. The query entered is "SELECT * FROM Persons". The "Format As" dropdown is set to "Table". The resulting table has four columns: PersonID, LastName, FirstName, and Birthday. There are two rows of data: PersonID 1, LastName Smith, FirstName Joe, and Birthday 2022-08-08T22:00:00.0...; and PersonID 2, LastName Smith, FirstName Jhon, and Birthday 2022-08-09T22:00:00.0....

Install Open Community Grafana Dashboard

By default, PACEEdge uses Telegraf to collect system health data and store it in InfluxDB. This data is stored in a database called telegraf_metrics under Grafana name EMR_InfluxDB_Telegraf. Few ready-to-use Grafana dashboards are available from the open community to visualize this data.

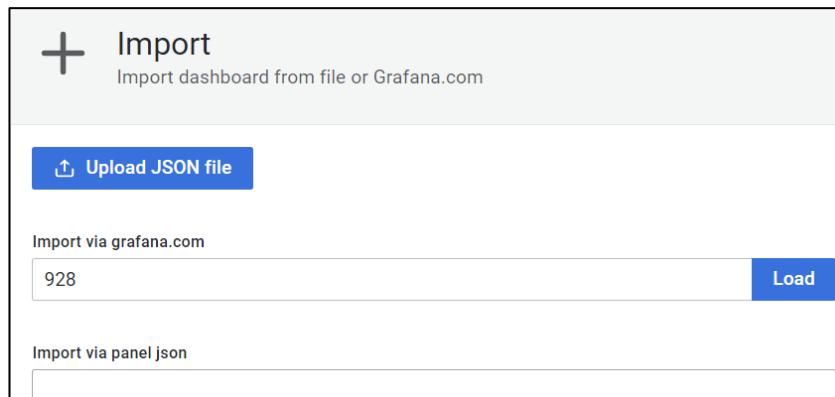
1. Open Grafana and log in as an admin user.
2. Hover the mouse over the Dashboards icon and select **+Import**

Figure 111: Dashboard



3. In the Import dialogue, enter number **928** into Import via the Grafana.com field and click the **Load** button.

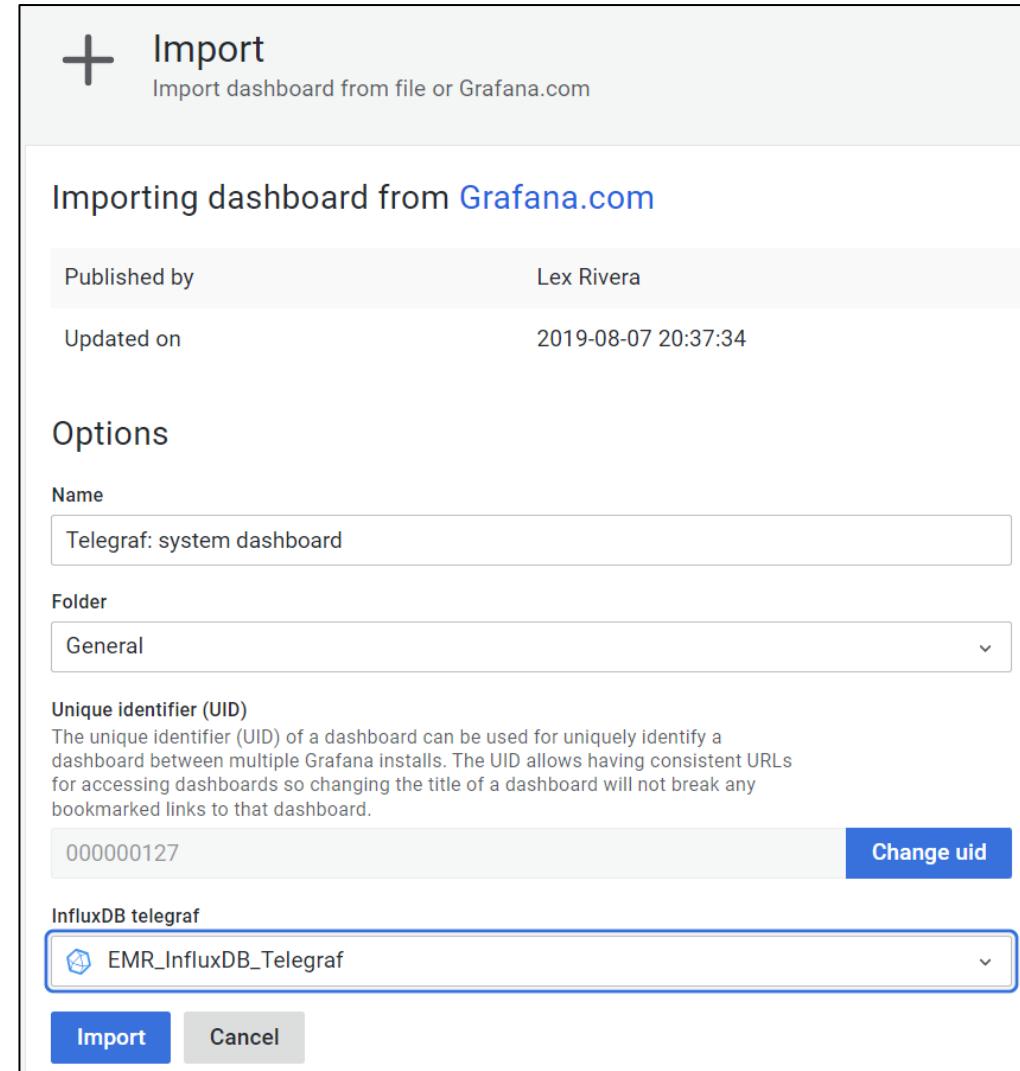
Figure 112: Import Button



Note: PACEEdge device requires access to the Internet for this operation

4. Select **EMR_InfluxDB_Telegraf** database from the drop-down menu at the bottom and click Import.

Figure 113: Importing dashboard from Grafana



5. A dashboard showing the variety of PACEdge system functions will be shown

Figure 114: Dashboard



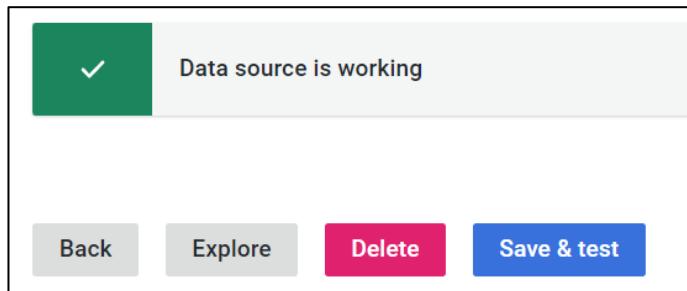
Configuring Grafana with the new InfluxDB database

This example will demonstrate how to access the InfluxDB database test created in the Node-RED example above.

Note: This example will fail if a database test does not exist.

1. Open Grafana from the PACEdge landing page and log in as the admin user.
2. Move the cursor over the Configuration icon on the left side and select **Data Sources**.
3. Click on Add Data Source and select InfluxDB.
4. Enter values as shown, the admin password, and click on **Save & Test** at the bottom. You should get a message **Data Source is working**.

Figure 115: Data Source



Note: If the database in InfluxDB has not been created yet, then trying to add it in Grafana will fail. Make sure that a particular database already exists before adding it to Grafana.

Figure 116: Grafana Configuration Screen for InfluxDB

Name ⓘ Test Example Default

Query Language

HTTP

URL ⓘ influxdb:8086

Access

Allowed cookies

Timeout ⓘ Timeout in seconds

Auth

Basic auth With Credentials

TLS Client Auth With CA Cert

Skip TLS Verify

Forward OAuth Identity

Custom HTTP Headers

InfluxDB Details

Database Access
Setting the database for this datasource does not deny access to other databases.
example: `SHOW MEASUREMENTS ON _internal` or `SELECT * FROM "_internal".`
To support data isolation and security, make sure appropriate permissions are configured.

Database

User

Password

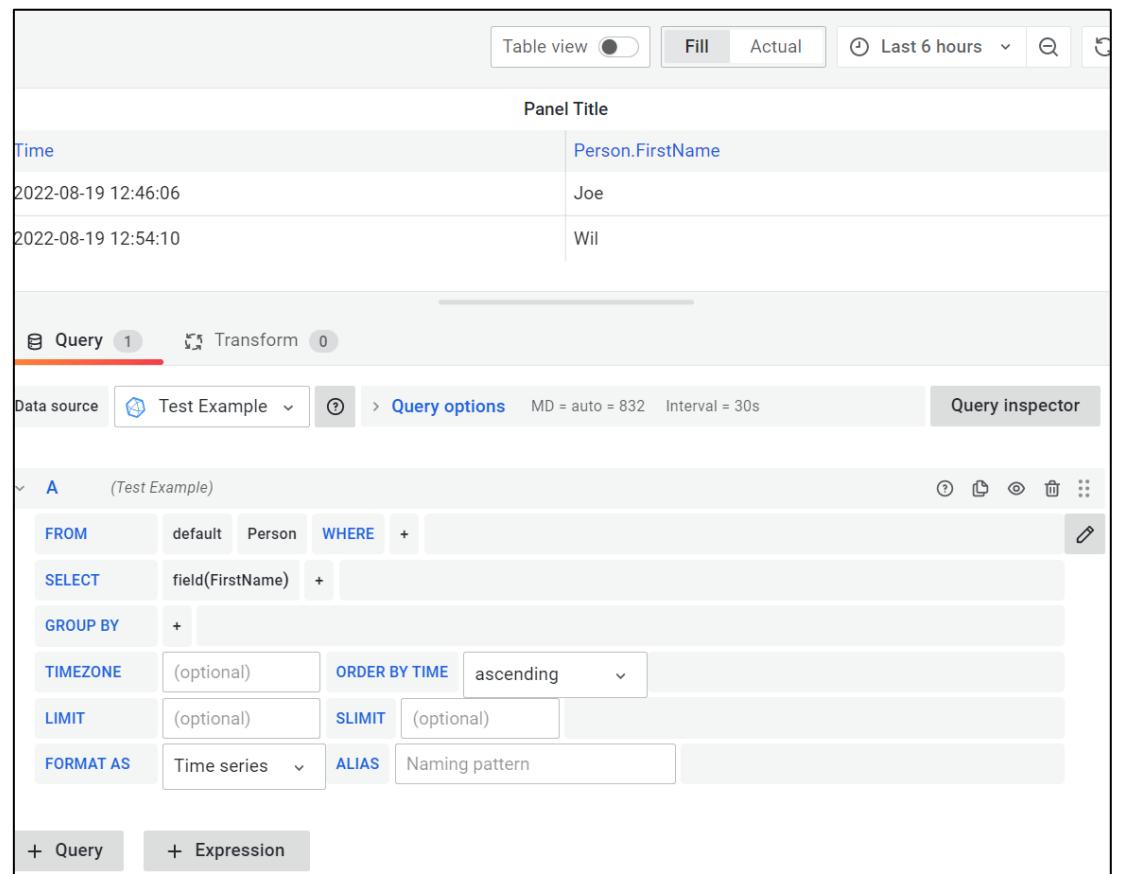
HTTP Method ⓘ

Min time interval ⓘ

Max series ⓘ

1. To create a new dashboard in Grafana, hover your mouse over the Dashboards symbol and click on **+New Dashboard**, then select the **Add a New Panel** option.
2. Select your configured InfluxDB database, **Test Example** in this case.
3. In line item **FROM** from the drop-down list, pick **Person**
4. In the **SELECT** line item, pick the field value **FirstName**
5. In the **GROUP BY** line, click on the word **Time** and select the option **Remove**
6. Click on the **Switch to table** option that will be shown.
7. At this point, you should see the first names of the persons you have entered in the Node-RED example.

Figure 117: Telegraf Metrics



The screenshot shows the Grafana interface with a panel titled "Panel Title". The top bar includes "Table view" (switched off), "Fill" (Actual), a time range selector ("Last 6 hours"), and a search icon. Below the title, there's a table with two rows:

Time	Person.FirstName
2022-08-19 12:46:06	Joe
2022-08-19 12:54:10	Wil

Below the table is a query editor:

- Query 1** (selected)
- Transform 0**

Query details:
Data source: Test Example
Query options: MD = auto = 832, Interval = 30s
Query inspector

The query builder shows the following configuration:

- FROM**: default Person WHERE +
- SELECT**: field(FirstName) +
- GROUP BY**: +
- TIMEZONE**: (optional) ORDER BY TIME ascending
- LIMIT**: (optional) SLIMIT (optional)
- FORMAT AS**: Time series ALIAS Naming pattern

Buttons at the bottom: + Query, + Expression.

Configuring Grafana with MySQL database

This example will show how to add a new MySQL database. For this purpose, we will use an already pre-configured database and add it under a different name.

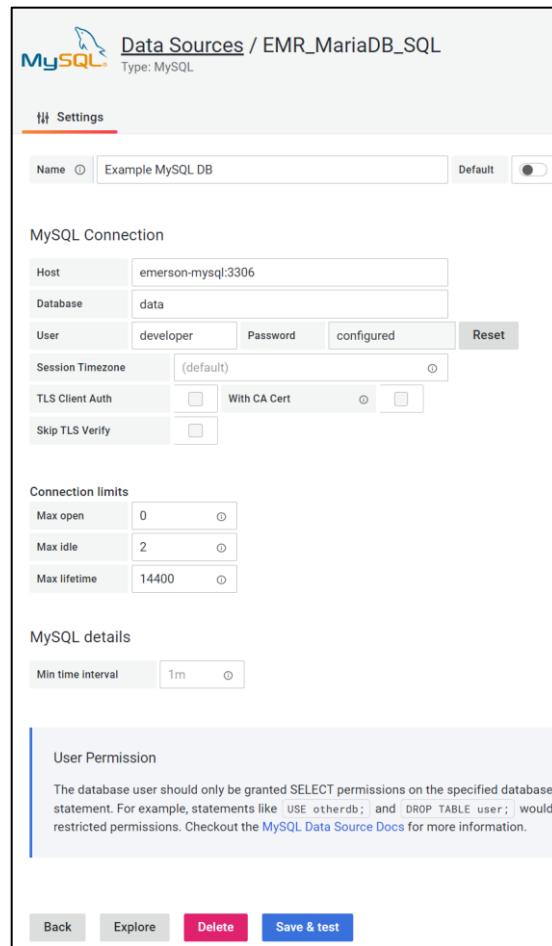
1. From the PACEEdge landing page, click on the **Grafana** link and log in as admin.
2. Move the cursor over the Configuration icon on the left side and select **Data Sources**.
3. Click on **Add Data Source** and select **MySQL**.

Note: if the database in MySQL has not been created yet, then trying to add it in Grafana will fail. Make sure that a particular database already exists before adding it to Grafana.

The following steps assume that the database Movicon has been created in MySQL.

4. Enter values as shown and the admin or developer password, and click on Save & Test at the bottom. You should get a message **Database Connection OK**.

Figure 118: MySQL Connection



5.6 Using Connex and WebHMI

This example shows how to instantiate the OPC UA client in Connex, get data from OPC UA Server on the CPL410 PACs Controller, and visualize this data via both WebHMI and Grafana.

5.6.1 Prerequisites

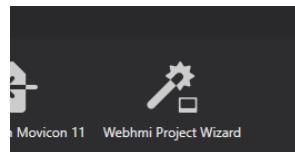
PACEdge version 2.2 or newer.

Basic knowledge of Movicon.NExT is required.

5.6.2 Example Walk Through

Movicon Connex and WebHMI functionality must first be created on the user workstation using Movicon.NExT software, typically running under Windows. Once required data connectivity, internal data tags, historian settings, and WebHMI screens are designed, a project can be uploaded to the unit running PACEdge. Note that when creating a new project with Movicon.NExT you need to select Webhmi Project type.

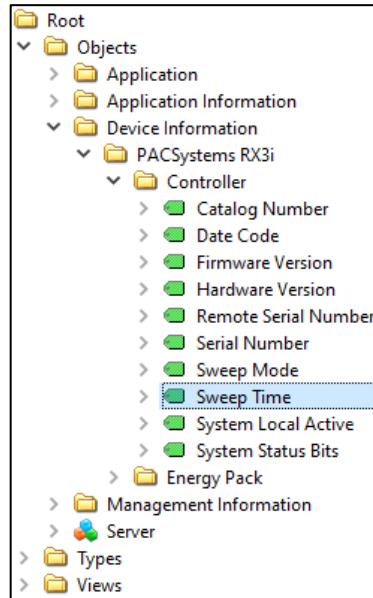
Figure 119: WebHMI Project



For detailed tutorials and documentation on creating Connex and WebHMI projects, please consult the [Movicon documentation](#). The following sections will provide only fundamental guidance.

For example, let us assume we have a CPL410 and want to use OPC UA to get data from the CPL410 PACs OPC UA server. PACs controller typically has a system tag called Sweep Time, shown in the UA Expert figure below:

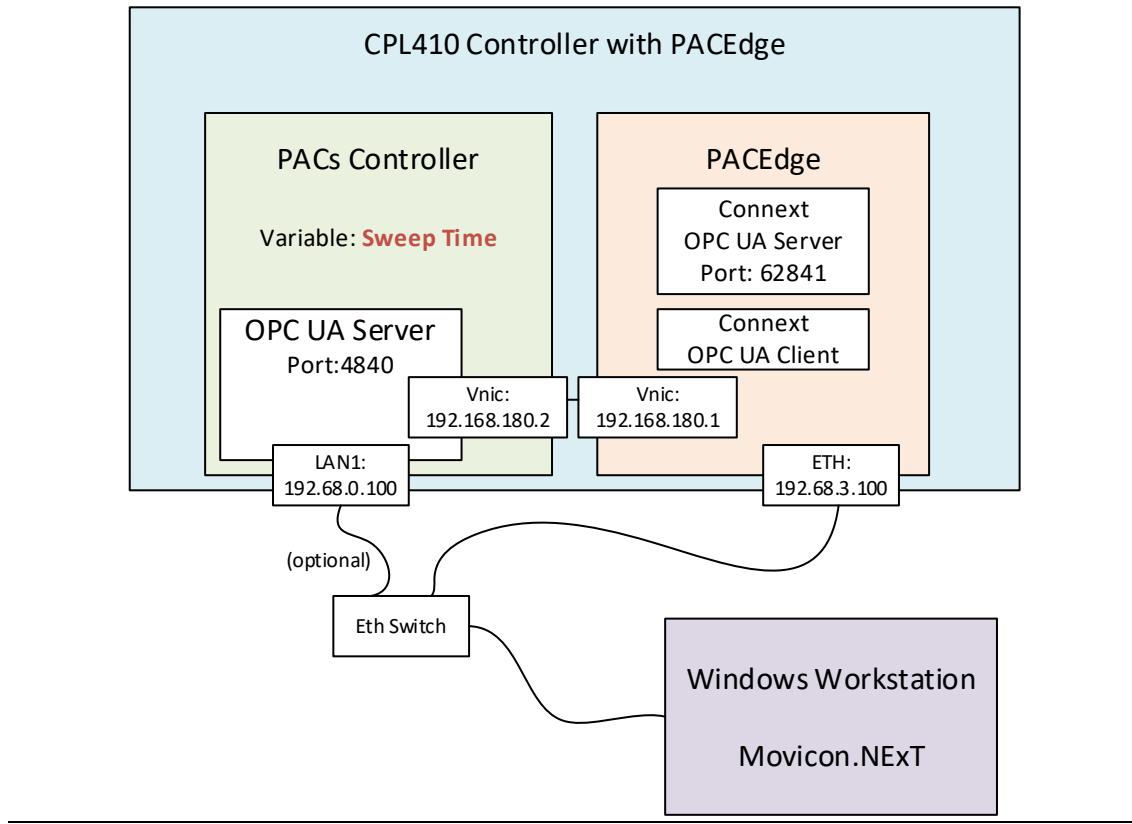
Figure 120: UA Expert view of CPL410 data tags



In our example, we will read this tag using I/O Data Server in WebHMI, store it with Historian in the PACEdge MySQL database, and visualize it via both WebHMI and Grafana.

CPL410 is a hybrid device containing both PACs automation controller and PACEdge. A PACSystems controller has its own OPC UA server, which enables access to different data tags. This OPC UA server can be accessed by PACEdge using an internal virtual Ethernet port (VNIC) and an external Ethernet LAN1 port.

Figure 121: CPL410 Internal Block Diagram



There are different ways to configure OPC UA Client, but for this example, we will utilize the new OPC-UA Port Forwarding feature that was added in the PACEdge version 2.2. The example will use the following steps:

- Use Movicon.NExT on the workstation to browse through OPC UA data tags on the PACs OPC UA Server using the new OPC-UA Port Forwarding feature via the internal VNIC port.
- Configure WebHMI project
- Test the WebHMI project locally on the workstation
- Transfer WebHMI project to target CPL410
- Test WebHMI project running on CPL410

Example Walk Through

Enabling the OPC UA Port Forwarding Feature

The OPC UA Port Forwarding feature enables to temporarily forward port used for OPC-UA communication via internal VNIC to an external Ethernet port. This allows the user to configure the Movicon OPC-UA driver without needing to plug an Ethernet cable into the LAN1 port on the CPL410 or CPE400 controller. To enable OPC-UA Port Forwarding, follow these steps:

1. Log into Cockpit as an **admin** user
2. Open the OPC-UA Port Forwarding tab
3. If using this feature for the very first time, a manual step accepting the security implications of such port forwarding is required. To do so, highlight the text shown in the Cockpit and use the **CTRL+C** key combination to copy:

Figure 122: Enable Access to a Remote OPC UA Server

Enable Access to a Remote OPC-UA Server

Create a Secure Shell (SSH) Tunnel to make remote OPC-UA Server accessible via the local PACEdge Ethernet interface. Once the Tunnel is created, the data can be accessed via the IP-Address of the PACEdge device and port 4840 i.e., IP-Address:4840.

NOTE:

Admin Privileges are required to run create the SSH Tunnel.

For security purposes, the SSH tunnel will be closed automatically after 1 Hour.

INITIAL STEPS

When using port forwarding to a given remote OPC-UA server for the first time a security consent needs to be given. Please make sure you trust the remote system with OPC-UA Server and perform following steps:

1. Copy following text: `ssh -g -L 4840:192.168.180.2:4840 admin@localhost sleep 1`
2. In Cockpit, open the Terminal Tab, copy the text above with **CTRL + C** and paste the text in the terminal via **CTRL + V** and press **Enter** button. **Note:** Change IP address as required if not using CPL410 or CPE400 PACEdge device.
3. Upon receiving authenticity warning type in **yes** and press **Enter** button.
4. Upon request enter your admin password
5. Wait until command prompt returns (about 1 sec), return to OPC-UA Port Forwarding tab in Cockpit and continue with setup.

OPC-UA Server	192.168.180.2
OPC-UA Port	4840
Login Timeout (in seconds. Max value: 3600)	60
Linux Admin Password	<input type="password"/>
Create Tunnel Close Tunnel	
Tunnel Status	

-
4. Open Terminal in Cockpit and Paste this line by using the **CTRL+V** key combination, hit enter

5. A message stating that the authenticity of the localhost can not be established asking if you want to proceed with this operation. Enter yes.
When asked, enter the admin password.

Figure 123: Enter Password

```
admin@pacedge:~$ ssh -g -L 4840:192.168.180.2:4840 admin@localhost sleep 1
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:NtzEp4aeb30j7svXjtgNVoMK0caCBJQQ6mR0Cw7tkd4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
admin@localhost's password:
admin@pacedge:~$
```

This operation set up a port forwarding for 1 sec and will close afterward. Wait a couple of seconds until the command prompt is back

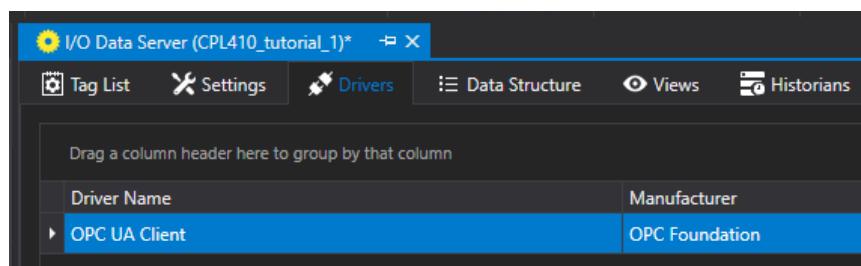
6. Return to the OPC-UA Port Forwarding tab in Cockpit, enter the admin password in the field, and click Create Tunnel. You will get the message: Tunnel Created. At this point, you can proceed with the next steps
Note: port forwarding is default enabled for 60 seconds and for only one session. It will automatically get disabled afterward. This is done for security reasons.

Configuring I/O Data Server

Following are the steps to configure the OPC UA Client in I/O Data Server to read a variable from PACs Controller:

1. In Movicon.NExT use a wizard to create new WebHMI project
2. In Movicon.NExT, in the I/O Data Server section, double-click on **Drivers**.
3. In the main window, right-click, select **Add New Driver**, and select **OPC UA Client** from the list, click on the > button.
4. Keep the default settings in all steps and click on the > button to proceed.
5. At the last step click on **Finish**. You will see an OPC UA Client driver listed in the driver window.

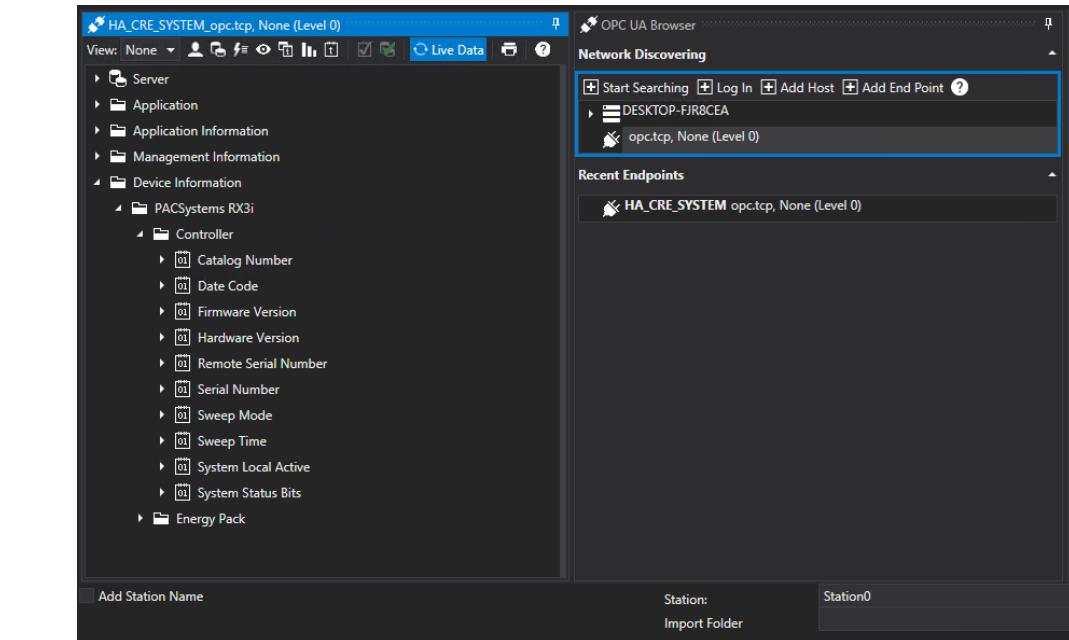
Figure 124: OPC UA Client



6. Right-click on the OPC UA Client entry and select **Import Tags**.

7. Select the **Browse Server** tab and then either:
 - a. **Add End Point:** `opc.tcp://ip_address_of_ETH_port:4840`
(where IP address is the address of ETH port, such as 192.168.3.100 if static default IP is being used)
OR
 - b. **Add End Point:** `opc.tcp://192.168.0.100:4840`
8. Double-click on the newly created server entry to connect to OPC UA Server, and browse for available tags.

Figure 125: Browse for Available Tags



9. Click on **Sweep Time** and click **OK**. A new data tag will be added to the Connect Tags List.

Configuring WebHMI Screen

1. In Movicon.NExT, right-click on **Screens** and select **New** to configure the HMI Screen.
2. Choose the screen size, enter the screen's name and click **OK**.
3. On the right edge of the screen, click on **ToolBox**, then on **Displays**, and then drag a **display** to the HMI screen.
4. Double-click on the display. In the Tag field, select the **Sweep_Time** tag. Expand the Display Settings, and in the Precision Digits, enter **6**. Click on the checkmark in the upper-left corner to close the Properties dialogue.
5. At this point, we can test to see if our setup is working on the workstation locally. To do so, save the project and click on the **Start Runtime** button. Once HMI starts locally on the workstation, you will see controller sweep time being updated once a second.

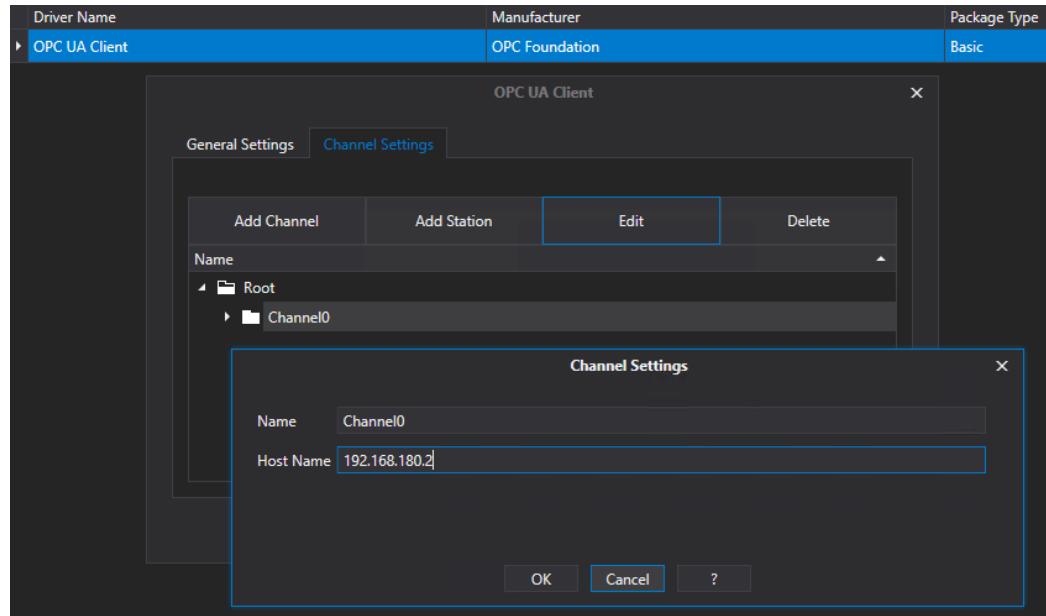
Note: To stop the HMI on the workstation, press the Alt+F4 keys.

Configure Connex/WebHMI Project for Deployment on CPL410 Controller

To transfer the WebHMI project from Movicon.NExT development tool to CPL410 controller, we need to change the ethernet interface to an internal VNIC inside the CPL410. Perform the following steps:

1. In Movicon.NExT I/O Data Server section, double-click on **Drivers**.
2. Double-click on **OPC UA Client**.
3. Select **Channel Settings**, select **Channel0**, and click **Edit**.
4. Change the Host Name to CPL410 VNIC IP address: 192.168.180.2.
5. Click **OK** to close all dialogues.

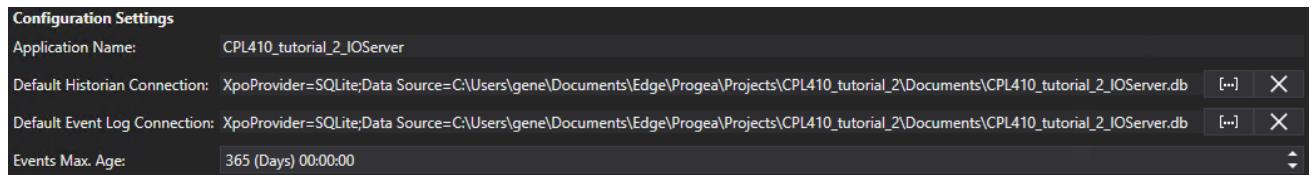
Figure 126: Host Name



In addition, configure the Historian in I/O Data Server to use the PACEdge MySQL database. To do so:

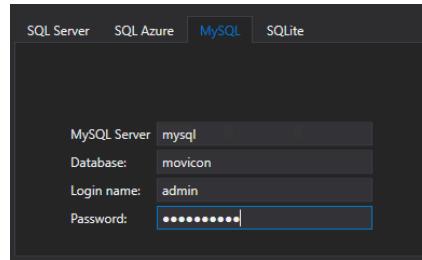
1. In Movicon.NExT I/O Data Server section, double-click on **Settings**.
2. In Default Historian Connection, click on the three dots on the right side to access settings.

Figure 127: Configuration Settings



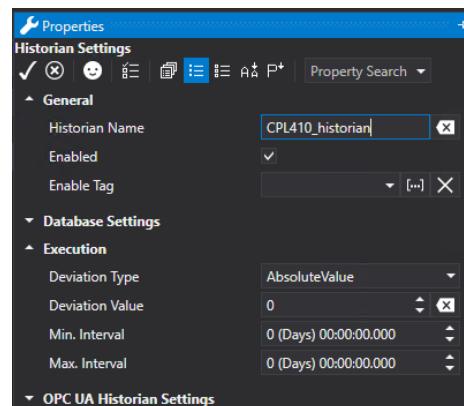
3. Select the MySQL tab, enter **mysql** as the server name, choose a name for the database, login name as **admin**, and choose the admin password you have configured for the PACEEdge. Click **OK**.

Figure 128: Password



4. At this point, you will get a message saying Unable to connect to the database, which is expected since you cannot connect to MySQL inside the PACEEdge from outside. Click **OK** to choose this dialogue.
5. Double-click on **Historians in the I/O Data Server** section, then in the main window, right-click on **Historian** and choose **Add Historian Settings**. Keep default settings, give some name to the Historian instance and click on the checkmark in the upper-left corner.

Figure 129: Properties



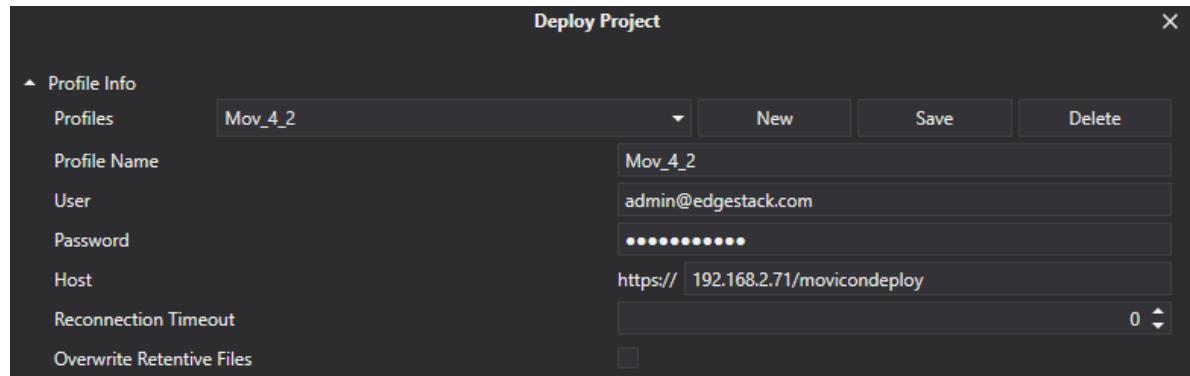
6. Right-click on the newly created historian, select Assign Tag, navigate to the **Sweep Time** variable, and click **OK**.
7. Save the Movicon.NEXT project

Deploy Movicon Project to CPL410 PACEEdge Controller

1. Click on the button **Create WebHMI**. Click on **Yes** to run WebHMI Server from remote.
2. In the Deploy Project window that opens, enter the username and password assigned using the Password Management utility.

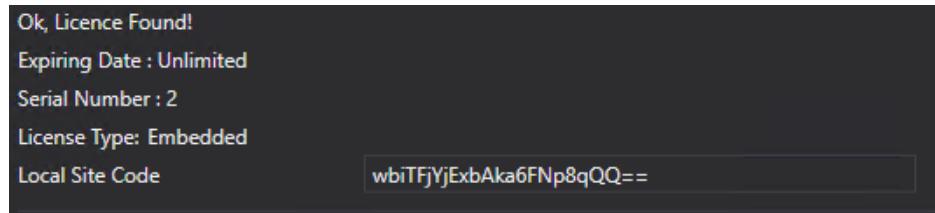
Note: username and password are stored in the appsettings.json file as described in section 4.2.2 PACEdge External Communication Parameters

Figure 130: Connxet Login



3. Click on the **Connect** button, respond with Yes to Deploy the Project, and upload I/O Data Server and WebHMI questions the first time you run.
4. Once everything is uploaded, expand Advanced Servers Start features and click on Start I/O Data Servers and Start WebHMI.
Note: if you are making changes to the Movicon project and re-deploying the new version onto CPL410, make sure to stop both I/O Data Server and WebHMI and start them again so that changes take effect.
Note: please be cautious when using the **Start Servers** button to start all servers (instead of starting each server individually, as described above). It was observed that this might lead to new instances of the servers being started without replacing the old versions.
5. Scroll down, expand **Remote Device** Info, and make sure both servers and one instance of each server are running.
Note: if you are changing from one Movicon project to another, please make sure to stop I/O Data Server and WebHMI (from within the old project), then download the new project and start servers. Failure to do so will result in multiple instances of WebHMI and I/O Data Server running, leading to unpredictable behavior.
6. You can also check the license status from the same Deploy Project window. Expand Remote License Info and then click on Check Remote License. The CPL410 license will look like the following:

Figure 131: License Found



At this point, go to the PACEdge landing page and refresh the browser. WebHMI Icon should become active, bringing you to the WebHMI screen.

Verifying OPC UA Servers

At this point, we have two OPC UA Servers active, one in CPL410 PACs Controller and one in PACEdge I/O Data Server. We can verify both by using an OPC UA Client on the workstation, such as UA Expert. In both cases, you will see the Sweep Time variable and can observe value changes:

- To connect to OPC UA Server in PACs Controller, either:
 - Enable the OPC-UA Port Forwarding feature, the open UA Expert, and connect to `opc.tcp://ip_address_ETH_port:4840`, where `ip_address_ETH_port` is an IP address of ETH port
 - Or, ensure the Ethernet cable is plugged into the LAN1 port on the controller, and the workstation is configured to communicate on the 192.168.0.x subnet. Open UA Expert and connect to `opc.tcp://192.168.0.100:4840`
- To connect to OPC UA Server in PACEdge I/O Data Server, ensure the Ethernet cable is plugged into the ETH port on the controller and the workstation is configured to communicate on the subnet you have configured PACEdge for (by default: 192.168.3.x). Open UA Expert and connect to `opc.tcp://192.168.3.100:62841`

Visualizing via Grafana

To visualize the Sweep Time variable in Grafana:

1. From the PACEdge landing page, click on the **Grafana** link and log in as an admin.
2. Move the cursor over the Configuration icon on the left side and select **Data Sources**.
3. Click on **Add Data Source** and select **MySQL**.
4. Enter values as shown and the admin password, and click on Save & Test at the bottom. You should get a message **Database Connection OK**.

Figure 132: MySQL

The screenshot shows the MySQL configuration interface. At the top, there is a header with a 'Name' field containing 'MySQL', a 'Default' toggle switch (which is turned on), and a help icon. Below the header, the title 'MySQL Connection' is displayed. The connection details are as follows:

Host	mysql		
Database	movicon		
User	admin	Password
TLS Client Auth	<input type="checkbox"/>	With CA Cert	<input type="checkbox"/>
Skip TLS Verify	<input type="checkbox"/>		

Below the connection details, the 'Connection limits' section contains the following values:

Max open	unlimited
Max idle	2
Max lifetime	14400

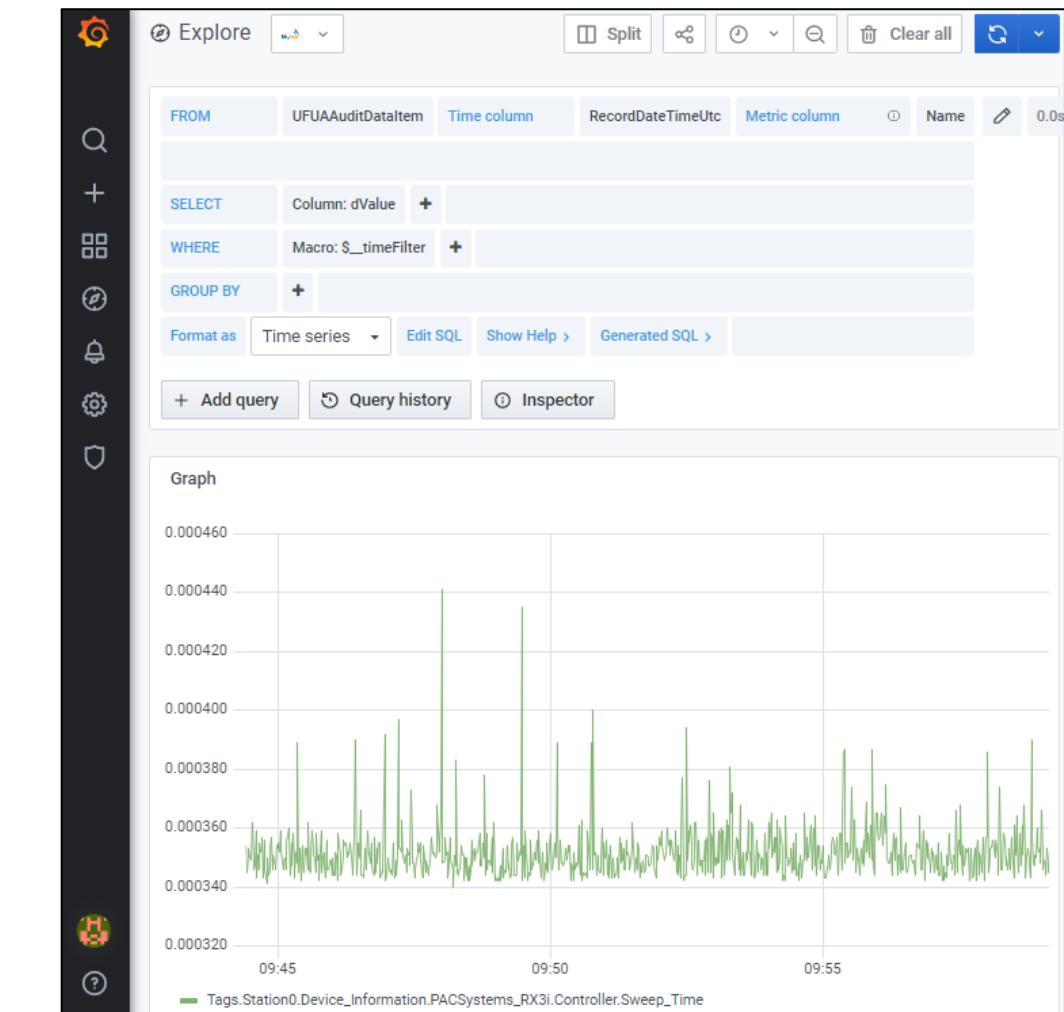
The 'MySQL details' section includes a 'Min time interval' field set to '1m'. In the bottom right corner of the main configuration area, there is a 'User Permission' box with the following text:

The database user should only be granted SELECT permissions on the specified database & table. Please validate that queries are safe so queries can contain any SQL statement. For example, statements like `DROP TABLE user;` would be executed. To protect against this we Highly recommend you create a new user with minimal permissions. Checkout the [MySQL Data Source Docs](#) for more information.

At the bottom of the configuration area, there are three buttons: 'Save & Test' (blue), 'Delete' (red), and 'Back' (grey).

5. On the left sidebar, click on the **Explore** icon.
6. Make sure the newly configured MySQL database is selected, then:
 - a. In FROM entry: make sure **UFUAAuditDataItem** is selected (default)
 - b. In Time column entry: choose what time stamp should be used, can leave at default
 - c. In Metric column entry: select **Name**
 - d. In SELECT entry: select **dValue**
 - e. You should see the historical values of the Sweep Time data tag.

Figure 133: Sweep Time Data

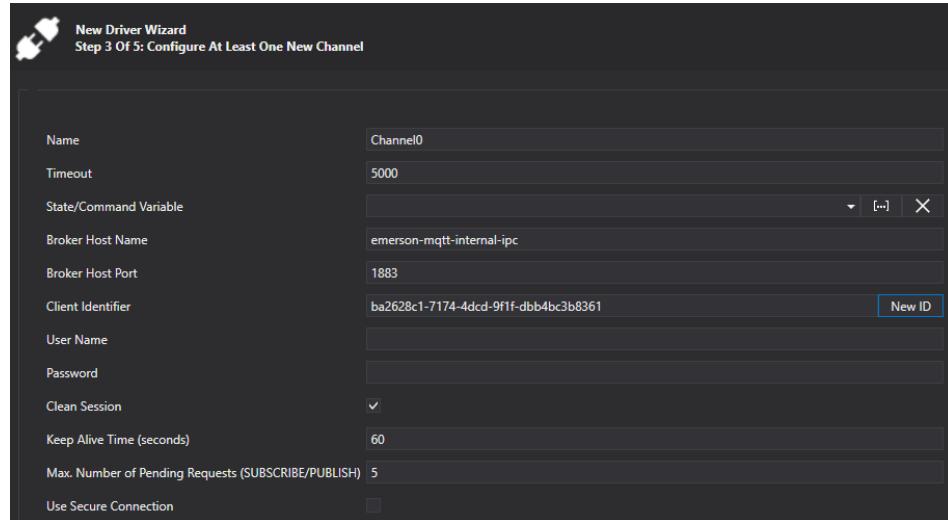


Storing Connex Data in InfluxDB via Telegraf

Connex can historize and store data directly in MySQL database; however, MySQL is a relational database, not well suited to high volumes of time series data. Telegraf is already pre-configured to store data automatically into InfluxDB. The following steps show how to extend the Connex configuration to store data in InfluxDB.

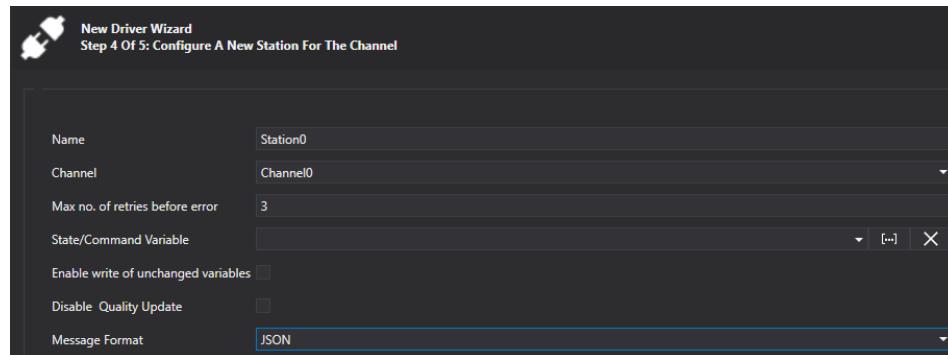
1. In the Movicon project, add an MQTT driver by double-clicking on the **Drivers** (left side of the screen), then right-click in the main window, select **Add New Driver**, scroll the list and select **MQTT Client** driver.
 - a. While configuring the MQTT driver, in step 3, enter Broker Host Name: **emerson-mqtt-internal-ipc**, then click on the **New ID** button to generate Client Identifier. Click on **>** to go to the next page.

Figure 134: MQTT Driver Configuration



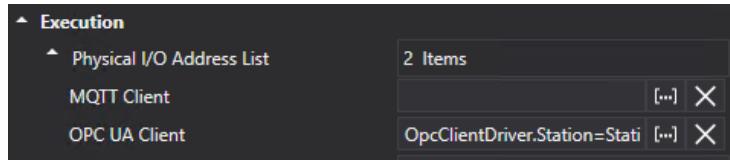
- b. In step 4, select **JSON** for Message Format, then **>** and **Finish**.

Figure 135: Select JSON Message



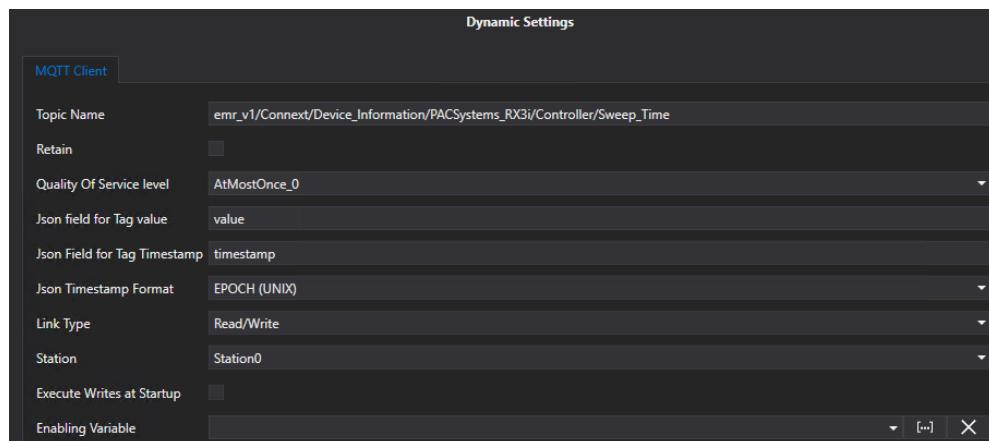
2. In Movicon Editor, navigate to your tag **Sweep_Time** and double-click on it to open Properties
3. In tag properties, Execution section, expand Physical I/O Address List and click on ... next to MQTT Client entry

Figure 136: Execution Section



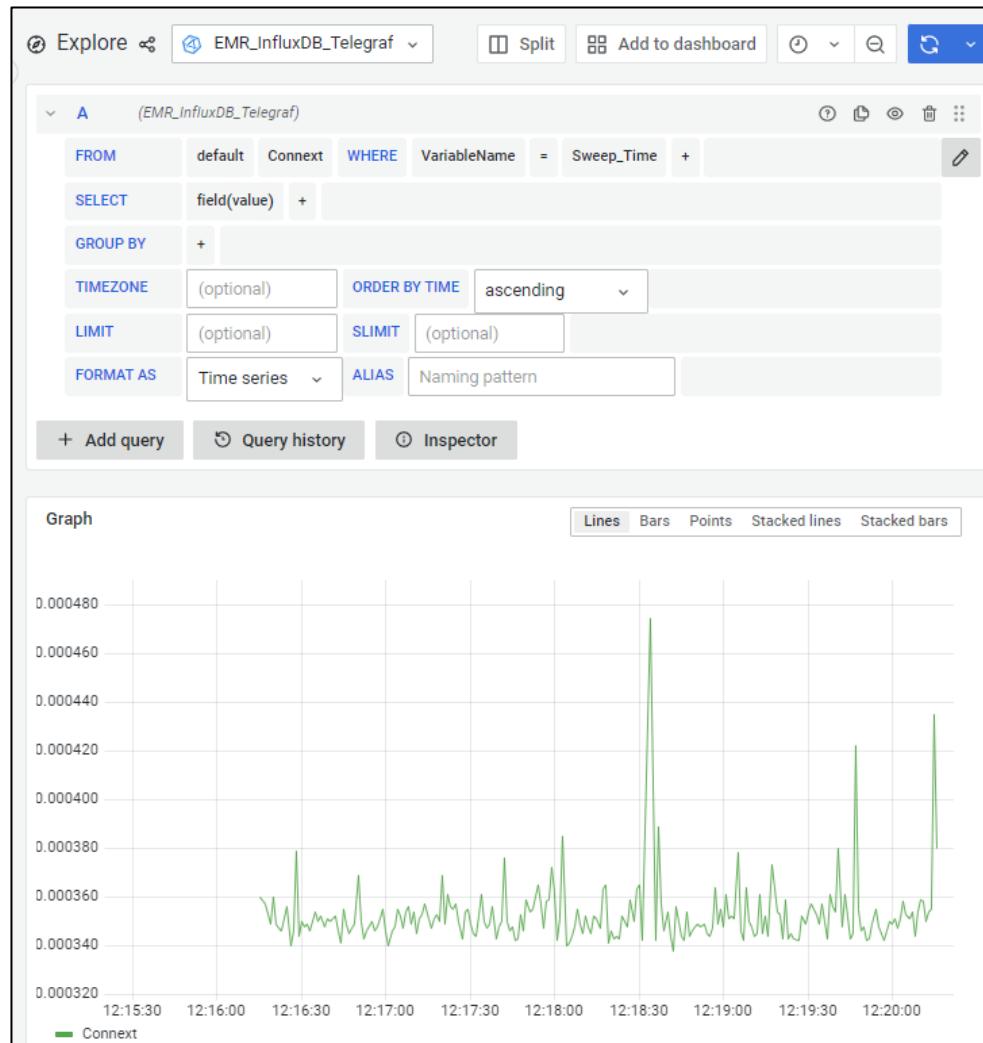
4. In the MQTT Client Dynamic Settings dialogue, enter the following:
 - a. Topic Name:
emr_v1/Connext/Device_Information/PACSystems_RX3i/Controller/Sweep_Time
Note: it is mandatory that the topic starts with **emr_v1** and has at least 3, but no more than eight segments.
 - b. In the Station field, select the newly created MQTT Client station; by default, it is **Station0**
 - c. Put in Json Field for Tag value: **value**
 - d. Put in Json Field for Tag Timestamp: **timestamp**
 - e. Select in Json Timestamp Format: **EPOCH(UNIX)**
 - f. Select in Link Type: Read/Write
 - g. Click OK and then check the mark to Accept Changes

Figure 137: Dynamic Settings of MQTT Client



5. Upload the newly modified project to the CPL410 PACEdge device by following the same steps as in Deploy Movicon Project to CPL410 PACEdge Controller
6. Open Grafana and select **Explore** option on the left side of the window
7. Select the pre-configured **EMR_InfluxDB_Telegraf** database
8. In the FROM field, select **Connext** as your Measurement name
9. In the SELECT field, choose **value**. You will see the Sweep Time values graphically shown. Feel free to experiment with WHERE fields, where you can find the variable name and other information about the data tag that was configured in Connext

Figure 138: EMR-InfluxDB_Telegraf



5.7 Configuring Telegraf to Log Data into InfluxDB

5.7.1 Example Description

PACEEdge implements a Telegraf agent, which is plugin-based and can pull data, statistics, and logs from a variety of databases, underlying system resources (CPU, memory, disc, kernel, software logs, docker containers, etc.), and external devices, heavily focusing on IoT protocols, such as MQTT, AMQP, Cloud resources, etc. For the complete list of available plug-ins, please refer to:

https://www.influxdata.com/products/integrations/? integrations_dropdown=telegraf-plugins

Telegraf runs in his container and has a configuration file, which users can access and modify based on the requirements.

Starting with PACEEdge v2.2.0, Telegraf comes pre-configured to:

- Pull system statistics and health information and store it in InfluxDB database **telegraf_metrics**, in Grafana mapped under the name: **EMR_InfluxDB_Telegraf database**
- Subscribe to Internal MQTT broker, all topics starting with: emr_v1/, parse the message and store data in InfluxDB database **telegraf_metrics**, in Grafana mapped under the name: **EMR_InfluxDB_Telegraf database**

5.7.2 Prerequisites

If new to Node-RED, completing 5.1 *Getting Started with Node-RED* and 5.4, *Using MQTT Communication* is highly recommended, as it shows some Node-RED usage basics and tips.

5.7.3 Example Walk Through

Steps to access the Telegraf configuration file:

1. From the PACEEdge landing page, open Cockpit and log in as an admin user.
2. Go to Navigator, navigate to /home/admin/pacedge/Emerson-telegraf/ folder, and open telegraf.conf file

If making configuration changes, save to changed telegraf.conf file and restart the Telegraf container using either Cockpit-> Docker Containers or Portainer. Alternatively, the whole IPC can be rebooted.

Storing System Statistics in InfluxDB

By default, Telegraf is configured to pull the system it runs on statistics and dump them into InfluxDB every 10 seconds. Password to gain access to InfluxDB is now configured by the Password Management utility, so please make no password changes in this file.

By default, Telegraf is configured to use InfluxDB database **telegraf_metrics** (which in Grafana is mapped to EMR_InfluxDB_Telegraf), where all data will be saved.

Figure 139: Telegraf Configuration File, database setting

```
### OUTPUT
# Configuration for influxdb server to send metrics to
[[outputs.influxdb]]
  urls = ["http://emerson-influxdb:8086"]
  database = "telegraf_metrics"
```

In the telegraf.conf file, there is an Output MQTT section, which is currently commented out but can be enabled by the user. Once enabled, all data that is being stored in InfluxDB will also be published via MQTT

Further down in the config file is a section on what inputs should be captured and stored in InfluxDB.

Figure 140 Defining Inputs to be Captured and Stored in InfluxDB (partial list)

```
# Read metrics about cpu usage
[[inputs.cpu]]
    ## Whether to report per-cpu stats or not
    percpu = true
    ## Whether to report total system cpu stats or not
    totalcpu = true
    ## Comment this line if you want the raw CPU time metrics
    fielddrop = ["time_*"]

# Read metrics about disk usage by mount point
[[inputs.disk]]
    ## By default, telegraf gather stats for all mountpoints.
    ## Setting mountpoints will restrict the stats to the specified mountpoints.
    # mount_points = ["/"]

    ## Ignore some mountpoints by filesystem type. For example (dev)tmpfs (usually
    ## present on /run, /var/run, /dev/shm or /dev).
    ignore_fs = ["tmpfs", "devtmpfs", "devfs", "iso9660", "overlay", "aufs", "squashfs", "snap"]

# Read metrics about disk IO by device
[[inputs.diskio]]
    ## By default, telegraf will gather stats for all devices including
    ## disk partitions.
    ## Setting devices will restrict the stats to the specified devices.
    # devices = ["sda", "sdb"]
    ## Uncomment the following line if you need disk serial numbers.
    # skip_serial_number = false

# Get kernel statistics from /proc/stat
[[inputs.kernel]]
    # no configuration

# Read metrics about memory usage
[[inputs.mem]]
    # no configuration
```

By default, you can see that CPU, disk usage, and disk performance statistics are enabled. Users can manipulate these settings and tune what statistics they are interested in.

Storing System MQTT Data in InfluxDB

By default, Telegraf is configured to subscribe to the internal MQTT broker: **emerson-mqtt-internal-ipc**, topics starting with **emr_v1/**. Once a message with the matching topic is received, Telegraf will parse the rest of the topic, attempting to extract the InfluxDB Measurement name, Tags, and Fields. The following data parsing algorithm is applied:

- Measurement: next topic field after emr_v1/
- VariableName tag: last field in the topic
- Level-0 through Level-4 tags: fields in between Measurement and VariableName

MQTT message itself needs to be a JSON object, consisting of **value** and **timestamp**, where value is a number and timestamp is in seconds.

An example:

The following MQTT message would be parsed and stored in InfluxDB as follows:

```
topic: "emr_v1/connext/level-1/level-2/var-3"
payload: object
  value: 977
  timestamp: "1662710488"
```

- Measurement: **connext**
- Tag VariableName: **var-3**
- Tag Level-0: **level-1**
- Tag Level-1: **level-2**
- Timestamp: 1662710488
- Field: Value = **977**

Note that by default Telegraf is configured to parse MQTT topics that are a minimum of three and a maximum of eight segments long. This can be changed by editing the Telegraf configuration file

Figure 141 Telegraf Configuration File for MQTT to InfluxDB Logging

```
[[inputs.mqtt_consumer]]
servers = ["emerson-mqtt-internal-ipc:1883"]  ## use emerson-mqtt-internal-ipc MQTT Broker
topics = ["emr_v1/#"]      ## subscribe to all topics starting with emr_v1
#topic_tag = ""           ## If uncommented, remove whole topic as a tag

data_format = "json_v2"
[[inputs.mqtt_consumer.json_v2]]
timestamp_path = "timestamp"
timestamp_format = "unix_ms"
timestamp_format = "unix"
[[inputs.mqtt_consumer.json_v2.field]]
path = "value"
#rename = "temp_avg"
type = "float"
[[inputs.mqtt_consumer.topic_parsing]]
topic = "+/+/"      ## parse topics with 3 segments
measurement = "/measurement/_/"    ## measurement name will be 2nd segment in topic
tags = "/_/VariableName"          ## name of the variable path
[[inputs.mqtt_consumer.topic_parsing]]
topic = "+/+/+/"    ## parse topics with 4 segments
measurement = "/measurement/_/_/"  ## measurement name will be 2nd segment in topic
tags = "/_/Level-0/VariableName"  ## name of the variable path
[[inputs.mqtt_consumer.topic_parsing]]
topic = "+/+/+/_/"    ## parse topics with 5 segments
measurement = "/measurement/_/_/_/"  ## measurement name will be 2nd segment in topic
tags = "/_/Level-0/Level-1/VariableName"  ## variable path and variable name
[[inputs.mqtt_consumer.topic_parsing]]
topic = "+/+/_/_/_/"    ## parse topics with 6 segments
measurement = "/measurement/_/_/_/_/"  ## measurement name will be 2nd segment in topic
tags = "/_/Level-0/Level-1/Level-2/VariableName"  ## variable path and variable name
[[inputs.mqtt_consumer.topic_parsing]]
topic = "+/_/_/_/_/_/"    ## parse topics with 7 segments
measurement = "/measurement/_/_/_/_/_/"  ## measurement name will be 2nd segment in topic
tags = "/_/Level-0/Level-1/Level-2/Level-3/VariableName"  ## variable path and variable name
[[inputs.mqtt_consumer.topic_parsing]]
topic = "+/_/_/_/_/_/_/"    ## parse topics with 8 segments
measurement = "/measurement/_/_/_/_/_/_/"  ## measurement name will be 2nd segment in topic
tags = "/_/Level-0/Level-1/Level-2/Level-3/Level-4/VariableName"  ## variable path and variable name
```

Section 6: PACEEdge Performance Indication

PACEEdge software is based on flexible and easy-to-use components, enabling fast application development. This section provides performance expectations and guides the user in selecting a suitable PACEEdge hardware option and software implementation choices.

The subsections below will document the OPC UA Client in Node-RED performance, Node-RED Dashboard performance, Node-RED Modbus TCP, and Connex Historian using MySQL performance. It is expected that the Connex OPC UA Client and Modbus TCP client will result in higher performance as in Node-RED.

Note: Interacting with PACEEdge via the locally (DisplayPort) attached monitor and local browser puts extra workload on the PACEEdge unit compared to the remotely accessing PACEEdge via Ethernet. Due to the lower computation performance of RXi2-LP, it is not recommended to use RXi2-LP in this mode.

In the test results below, "Loc" indicates that PACEEdge is being used with a locally attached Monitor, keyboard, mouse, and browser inside the PACEEdge, whereas "Rem" indicates that PACEEdge is being accessed remotely via Ethernet.

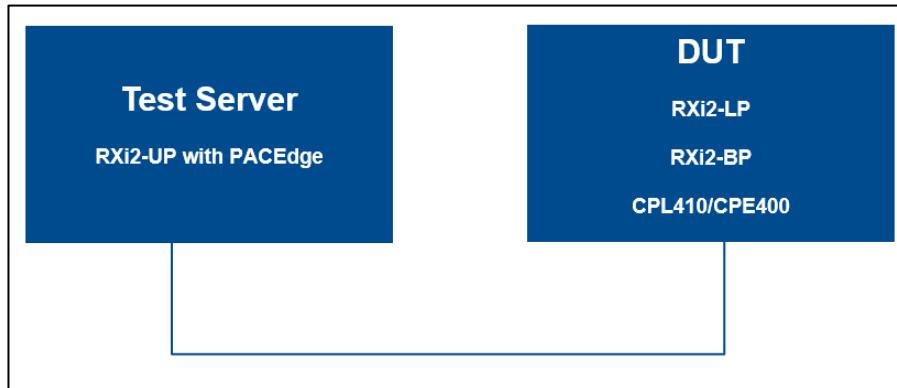
6.1 OPC UA Performance Indication

THE OPC UA Performance test was done using two PACEEdge devices: one configured to act as OPC UA Server (highest performance IPC, RXi2-UP) and the other as an OPC UA Client. The server was set up to generate a large number of variables and to have them updated at the specified rate. The client was using two ways of getting variable updates:

1. Using single variable read operation:
 - a. The client sends 100 single-variable read requests
 - b. The server responds with 100 replies
 - c. Once all replies are received, the Client repeats the sequence.
2. Using batch read request:
 - d. The client sends one read request containing 100 variables
 - e. The server responds with 10 replies, each containing 10 variable updates
 - f. Once all replies are received, the Client repeats the sequence

In addition to performing OPC UA read transactions, PACEdge stores all variables in the InfluxDB database and uses one graph and one dial to show performance via the Node-RED dashboard and the Grafana dashboard.

Figure 142: Performance Indication



The following table provides test results of the Node-RED OPC UA Client performance test:

Table 5 OPC UA Variable Read Performance (variables/sec)

OPC UA Client Functions	Units	LP (Rem)	BP (4-core) (Rem)	BP (4-core) (Loc & Rem)	CPL410/ CPE400 (Rem)
Test: OPC UA Single Var Read InfluxDB Node-RED Dashboard Grafana Dashboard	Var/sec	100	1000	1000	90
Test: OPC UA Batch Var Read InfluxDB Node-RED Dashboard Grafana Dashboard	Var/sec	450	3100	2900	350

6.2 Node-RED Dashboard Performance Indication

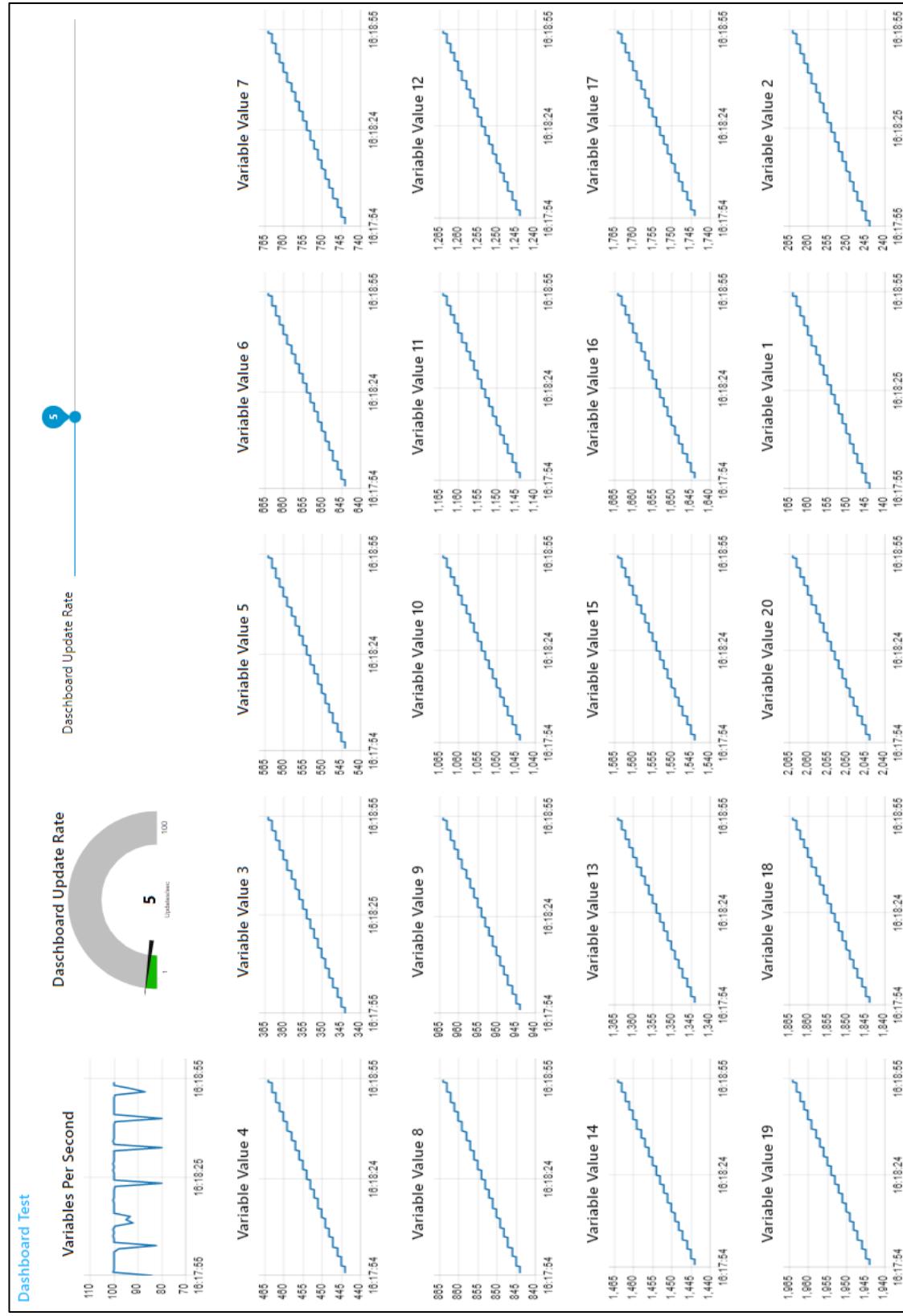
For this test, PACEdge with OPC UA Client reads 20 variables from OPC UA Server, stores them in the InfluxDB, and displays them in 20 Node-RED Dashboard graphs. The read and graph update rate is increased to the point where further increases result in noticeable control latency and lag in graphs being updated. In addition, Grafana is being used to display CPU, Memory, and Disk utilization statistics.

Note: in this particular test, a heavy load is being placed on a computer where the web browser is running, which might become a performance limiting factor.

Table 6 Node-RED Dashboard Performance (variable updates per second)

Functions	Units	LP (Rem)	BP (4- core) (Rem)	BP (4- core) (Loc & Rem)	CPL410 (Rem)
OPC UA Client Node-RED DashboardGrafana	Dashboard Update Rate: (Var/sec)	80	100	80	50

Figure 143: Dashboard Update Rate



6.3 Node-RED Modbus TCP Performance Indication

In this test, the PACEdge device with Modbus TCP (Server) reads 1000 variables from Modbus TCP (Client). The server issues 10 FC3 read requests for 100 variables each. After all, responses are received, the read cycle is repeated. All received variables are stored in InfluxDB; the performance graph is displayed in Node-RED Dashboard; IPC load is being captured via Telegraf and displayed via Grafana

Table 7 Node-RED Modbus TCP Read Performance

Functions	Units	LP (Rem)	BP (4-core) (Rem)	BP (4-core) (Loc & Rem)	CPL410 (Rem)
Modbus TCP Server InfluxDB Node-RED Dashboard Telegraf Grafana	Variable Read: (Var/sec)	8K	40K	40K	4K

6.4 Node-RED Connexx Historian Performance Indication

In this test, the PACEdge device uses the Modbus TCP Client driver within the Connexx data gateway to read variables and store data via Historian inside the MySQL database. The test measures how much data Historian can store in MySQL. In parallel, Telegraf collects the IPC load, and Grafana displays it.

Client Functions	Parameter	LP (Rem)	BP_1605 (Rem)	BP_1605 (Loc & Rem)	CPL410 (Rem)
Connexx Modbus TCP driver MySQL Telegraf Grafana	Variable store rate: (Var/sec)	50	750	750	50

Section 7: Saving and Restoring User Data

PACEEdge software consists of several applications, such as Node-RED, InfluxDB, Grafana, MySQL, Connex, WebHMI, etc. When a user creates his specific flows and stores data in the databases, all this user-specific data will be stored in the Linux file system. This data can be periodically copied and stored in other locations, such as a USB stick, for backup purposes.

User data location is at /home/admin/docker/volumes/.

The following are the most critical applications/folders:

- emersonedgestack_grafana-storage
- emersonedgestack_influxdb
- emersonedgestack_mysql
- emersonedgestack_nodered
- emersonedgestack_portainer
- emersonedgestack_movicon

Note: The data stored/restored is in the **/home/admin/docker/volumes/application-name/_data/** folder. When restoring data, copy back folders into this location. Note access requires admin user rights.

Note: Cockpit/Navigator can be used to copy files and folders using the graphical user interface

Alternatively, Node-RED flows and Grafana dashboards can be stored (exported) and later installed (imported) from each application natively. Connex and WebHMI projects are, by definition, developed on engineering workstations using Movicon.NExT software can be restored by performing a new upload to the PACEEdge device.

7.1 Exporting and Importing Flows in Node-RED

7.1.1 Export Flow

While in Node-RED application:

1. Click on the menu icon on the right side of the screen 
2. Select either the **current flow** or **all flows** option.
3. Click on the **Download** button.
4. Select the location (on your client system, not PACEEdge) where to store the **flow.json** file.

7.1.2 Import Flow

While in Node-RED application:

1. Click on the menu icon on the right side of the screen  .
2. Click on select a file to import.
3. Navigate to your client system's desired flow xxx.json file (not PACEdge).
4. Click on **Open**.
5. Click on **Import**.

7.2 Exporting and Importing Views in Grafana

7.2.1 Export Flow

While in the Grafana application:

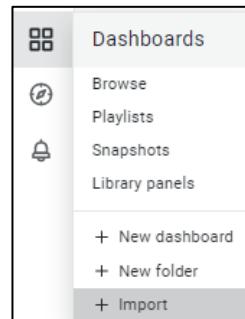
1. Navigate to your desired Dashboard.
2. Click on the **Share Dashboard** icon  .
3. Click on **Export**.
4. Click on **Save to file** and select the location to store it on your client system (not PACEdge).

7.2.2 Import Flow

While in the Grafana application:

1. Navigate to **Dashboards** and select the **+Import** option

Figure 144: Dashboards



-
2. Click on Upload JSON File.
 3. Select your desired Grafana view file xxx.json and click on **open**.

4. Select the **InfluxDB** database from the drop-down menu

Note: If there is no valid InfluxDB database there, then most likely Grafana was not configured yet to connect with InfluxDB. Please follow the steps in configuring Grafana in this manual first.

5. Click on the **Import** button.

7.3 Importing Projects in Connex/WebHMI

Connex and WebHMI projects are first created on engineering workstations using Movicon.NExT software and then loaded into the PACEEdge device. Therefore there is no need to export projects from PACEEdge. Projects can be loaded from the engineering workstation onto the PACEEdge device when recovery is needed.

7.4 Backing Up and Restoring InfluxDB and MySQL Databases

InfluxDB and MySQL databases are stored in the main Linux file system at the following path:

- /home/admin/docker/volumes/emersonedgestack_influxdb/_data/
- /home/admin/docker/volumes/emersonedgestack_mysql/_data/

As an example, see the figure below:

Figure 145: Example paths

```
admin@paceredge:~$ sudo ls -l docker/volumes/emersonedgestack_influxdb/_data/data
total 12
drwx----- 4 root root 4096 Aug 12 11:48 _internal
drwx----- 4 root root 4096 Aug 12 11:48 telegraf_metrics
drwx----- 4 root root 4096 Aug 23 18:03 test_1_db

admin@paceredge:~$ sudo ls -l docker/volumes/emersonedgestack_mysql/_data/data
total 104
-rw-rw---- 1 systemd-coredump systemd-coredump    67 Aug 12 11:48 db.opt
-rw-rw---- 1 systemd-coredump systemd-coredump 1556 Aug 23 18:03 gj_test_data.frm
-rw-rw---- 1 systemd-coredump systemd-coredump 98304 Aug 23 18:40 gj_test_data.ibd
```

When a backup is desired, the user can copy all folders from/_data/ folder and restore them to the same location later.

Note: Access to these folders requires admin privileges.

7.5 Saving License Files

PACEEdge devices come with pre-installed software and valid factory license files. Under normal circumstances, the user does not have to worry about keeping a copy of the valid license file. Still, if restoring to PACEEdge Factory Default needs to be done on RXi2-BP and RXi2-LP Industrial PCs, then a valid license file will be required afterward to fully activate the PACEEdge software. In such a situation, the user can always request a new license file from the Customer Care representative, as described in section 3.7, *PACEEdge License File*. Still, an alternative would be to make a copy of the license file before performing a restore operation. The easiest way to copy files is using the Navigator application in Cockpit. To do so:

1. Open Navigator
2. Navigate to: /home/admin/pacedge/emerson-software/license/ folder
3. Right-click on the **license.json** file and choose the Download option, specify the location to store the file on your local PC

NOTE: If the download fails with the error: **Failed – Network error**, log out from Cockpit and log in again as the admin user.

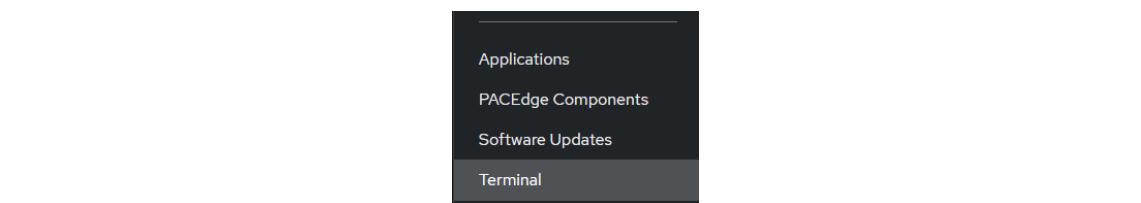
4. Repeat the steps for **license.sig** file in the same folder.

Alternatively, one can store license files on a USB stick directly attached to the PACEEdge device by following the steps below:

Mount the USB stick by following the procedure in **10.1 Mounting USB Stick**.

1. Within the Cockpit, go to the Terminal screen and enter the following commands:

Figure 146: Open Cockpit's Terminal



- a. sudo cp pacedge/emerson-software/license/license.json /mnt/usb
 - b. Enter the admin password (if requested).
 - c. sudo cp pacedge/emerson-software/license/license.sig /mnt/usb
2. Next, unmount the USB stick by following: 10.2 Un-Mounting USB Stick.
 3. You may remove the USB stick and save the files: 1) license.json and 2) license.sig on your Windows workstation.

Section 8: PACEEdge Software

Backup/Recovery

PACEEdge software allows users to install additional software packages and gives full flexibility to make configuration changes. It is expected that during the development and validation phases, the user will want to either back up the current setup or restore the original PACEEdge state.

The backup procedure allows the user to create a copy of PACEEdge software with your customizations, flows, views, databases, Connex, and WebHMI projects, including Linux operating system. This step might be useful for creating an operational backup or duplicating the existing setup onto several additional systems.

The restore procedure allows users to restore PACEEdge either to the Factory Default state or the previously made backup. Note that the backup/restore procedure can also be used to create a golden setup image and then restore it on multiple other PACEEdge devices, even migrating from RXi2-LP to RXi2-BP or vice versa.

Note: be aware that when performing a restore operation, all current data on the device, including databases and current license files, will be erased. Please backup critical data, if any, and license files using procedures documented in Section 7: *Saving and Restoring User Data*.

Note: This procedure shows how to recover PACEEdge software installation back to Factory Default; however, if the user has entered UEFI setting menus and modified UEFI settings, these will remain as is. To restore UEFI settings to Factory Defaults, enter the UEFI Settings menu and choose the recovery option on the Save & Exit page.

Given that CPL410 and CPE400 have integrated automation controllers and PACEEdge functionality, backup/restore procedures are slightly different than those for IPCs; please refer to the applicable section below.

8.1 PACEEdge Software Backup on RXi2-BP, RXi2-LP IPCs

From the [Emerson Software Downloads](#) site, download **PACEEdge v2.2.0 Backup/Restore RXi2-BP/RXi2-LP** (file name: **PEv220BPRestore.zip**). Then follow the steps below to perform a backup:

1. Get an empty, min 32 GB size USB stick. Make sure it is empty, as conflicts might arise.

Note: if using old USB sticks, one might experience a problem in which a device may not boot from the USB stick. This is typically due to the file system changes on the USB stick. The workaround, in this case, is either: 1) use a brand new USB stick, 2) download from the Internet one of Linux installation ISO images, use a Windows utility such as Rufus, to make a bootable USB stick, then delete all the files from the USB stick and proceed with next steps.

2. Copy all files to the root directory of the USB stick. The directory structure should look like this:

Note: Double-check if the automatic PACEEdge Factory Image install is disabled; otherwise, it will automatically overwrite all your data and install a Factory Default PACEEdge image. To check:

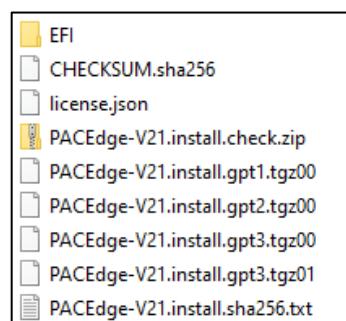
- a. In Notepad or a similar editor, open file: EFI\BOOT\unattendedinstall
- b. Look at the top of the file for a line **action=" install"** or **action=" backup"**.

Make sure this line is commented out with # in front of it. This way utility will stop and ask you what operation should be executed.

```
# action: action to perform
# if set no user input is required
# valid values: "backup", "restore", "install"
# default: not set (== user needs to select action in gui)
# action="backup"
```

- c. Make changes as necessary and save the modified file.

Figure 147: Copy All Files to the Root Directory of the USB Stick



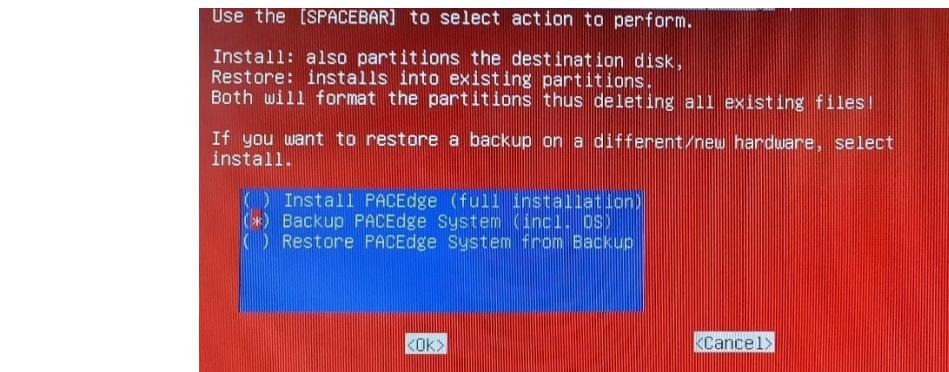
3. Plug the USB stick into any of the USB ports, power up the IPC, and keep pressing the F7 button to get into the boot selection menu; select the USB Stick to boot from (in the figure below, this would be the second line item: UEFI: SanDisk Extreme Pro 0.)

Figure 148: Select the Boot Device



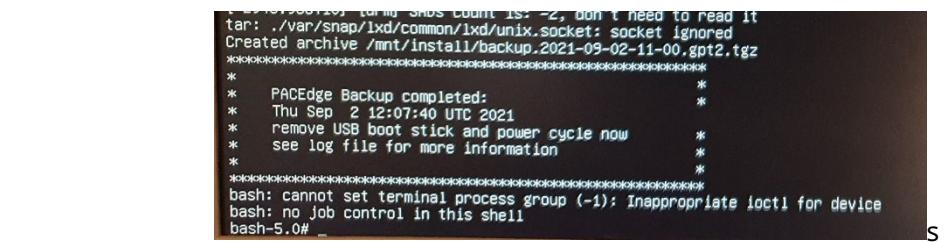
4. Wait until the following dialog appears. Use arrow keys to move the cursor and the space key to select the **Backup** option. Then use the Tab key to move the cursor to the **OK** button and hit enter.

Figure 149: Select the Backup Option



5. Wait for the completion message as shown below. Depending on the hardware type, creating a backup might take five to 20 minutes.

Figure 150: Backup Completed Screen



Note: After the backup, on the USB stick, you will see the second set of installation files with word **backup** in them.

8.2 PACEEdge Software Restore on RXi2-BP, RXi2-LP IPCs

To restore PACEEdge software, either to factory default or to your previously made backup, please perform the following steps:

1. Perform steps 1-5 document in Section 8.1 *PACEEdge Software Backup on RXi2-BP, RXi2-LP IPCs*.
2. Wait until the following dialog appears, and then use the arrow keys to move the cursor and the space key to select the **Restore PACEEdge** option. Then use the Tab key to move the cursor to the **OK** button and hit enter.
Note: if restoring on a different hardware than where the backup was made, please make sure to use **Install PACEEdge (full installation)** option, as opposed to **Restore PACEEdge**.

Figure 151: Select the Restore from Backup Option (when restoring on the same PACEEdge hardware unit)



3. If asked for a disk partition, use the arrows, space, and Tab keys to select the first partition listed and click **OK** to proceed.

Figure 152: Select the Disk Partition (If Applicable)



4. Use arrows, Space, and Tab keys to select either a factory default image (**PACEEdge-V220.install**) or one of your backups, then choose **OK** and hit enter.

Figure 153: Select the Backup Image



5. Wait until the restore procedure completes, as indicated by a similar message.
Note: during the installation process, please ignore any questions that might come up on the screen, letting them time out and continue. No user intervention is required.

Note: Depending on the HW type, this step may take anywhere from a couple of minutes to 10-20 minutes.

Figure 154: PACEEdge Restore Completed

```
Proceed anyway (or wait 5 seconds to proceed) ? (y,N) <proceeding>
Mounting /dev/sda1 on /mnt/gpt1
Installing partition SW for gpt1 (taropt: --no-same-owner)
Mounting /dev/sda2 on /mnt/gpt2
[ 65.537267] EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts: (null)
Installing partition SW for gpt2 (taropt: --xattrs)
Mounting /dev/sda3 on /mnt/gpt3
[ 117.847601] EXT4-fs (sda3): mounted filesystem with ordered data mode. Opts: (null)
Installing partition SH for gpt3 (taropt: --xattrs)

*****
*          PACEEdge Restore completed: *
* Thu Sep 2 11:26:13 UTC 2021      *
* remove USB boot stick and power cycle now   *
* see log file for more information        *
*                                              *
*****
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.0#
```

Note: If you have backed up a PACEEdge device with a valid license and then restored it on the same physical unit, the license will also be restored and remain valid. If restore is done on a different physical unit, the license will be invalid and require a new license file to be manually installed, as described in 3.7, *PACEEdge License File*.

NOTE: After PACEEdge restore, you might have to Upload again Movicon project (if used) by using Movicon.NExT Editor.

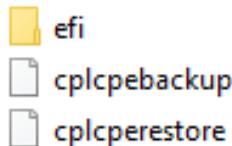
8.3 PACEEdge Software Backup on CPL410, CPE400 Controllers

To perform PACEEdge software backup, from the [Emerson Software Downloads](#) site, download **PACEEdge v2.2.0 Backup/Restore CPL410/CPE400** (file name: **PEv220CPLRestore.zip**).

Please use the following procedure to perform a backup:

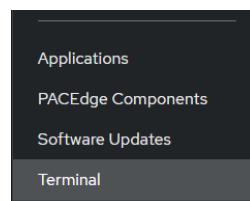
1. Copy the files to an empty USB Stick. The following are the required files:

Figure 155: Copy Files to USB Stick



2. Insert USB-Stick into CPL/CPE, USB1 port.
3. Mount USB stick in Linux OS as described in 10.1. Mounting USB Stick
4. Now, within the Cockpit, go to the Terminal screen and enter the following commands:

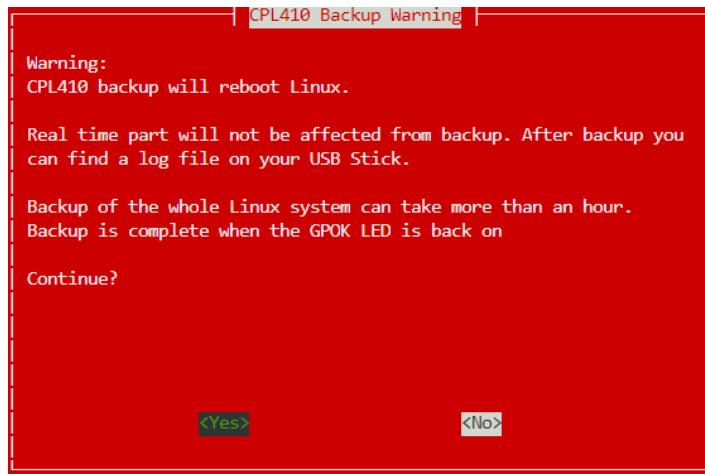
Figure 156: Cockpit's Terminal



- a. **cd /mnt/usb/**
- b. **sudo ./cplcpebackup**
- c. Enter the admin password if asked for.

5. Using the arrow keys, select **Yes** and press **enter**.

Figure 157: Select the Backup



6. When the backup starts, GPOK LED will turn off. It will remain off until the backup is complete, the PACEEdge side of the controller is rebooted, and the GPOK LED is ON. It might take close to 2h to perform a backup. You can observe SSD LED, and as long as it periodically flashes backup process is in progress.

Before removing the USB stick, please properly un-mount it by the following procedure in 10.2 Un-Mounting USB Stick

7. At this point, you can remove the USB stick and inspect the content on Windows PC. It will have several large archives with names starting with backup.

8.4 PACEEdge Software Restore on CPL410, CPE400 Controllers

8.4.1 CPL410, CPE400 PACEEdge Recovery to Factory Default

CPL410 and CPE400 have a pre-installed Factory Default image. If Factory Default recovery is needed, please follow the steps below. Switching to the Factory Default image will delete all existing data, including Node-Red flows, Grafana views, and database content.

1. Using CPL410/CPE400 built-in display trigger Reset to Factory Default process:
 - a. Press the DISP button until the arrow points at the menu entry: **Edge Settings**, then press SEL.
 - b. Press the DISP button until the **Commands** option is selected, then press SEL.
 - c. Press the DISP button until the **Factory Reset** option is selected, then press SEL.
 - d. Press the DISP button until the **OK** option is selected, then press SEL.
2. At this point, GPOK LED will go OFF, and you need to reboot a complete unit by removing and applying power again. After the unit is up again, check the update status via display:
 - a. Press the DISP button until the **Edge Settings** option is selected, then press SEL.
 - b. Check that the message **Resetting GP** is shown. Also, a blinking SSD LED will indicate the recovery is in progress.
3. Wait until GPOK LED turns ON. This can take up to 20 minutes.
4. At this point, Factory Default restore is done. PACEEdge is functional, and the License file is valid.

8.4.2 CPL410, CPE400 Restore of PACEEdge Backup Image

If a PACEEdge backup image was created, it could be later restored on the same HW family unit (meaning CPL410 to CPL410 or CPE400 to CPE400). If restoring to a physically different unit, new valid licenses will be required afterward to activate the PACEEdge.

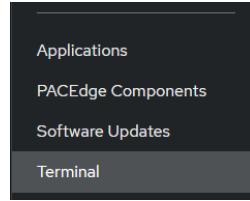
Note that the backup restore operation will delete all existing data, including Node-Red flows, Grafana views, and database content

To perform PACEEdge software restore from the backup, use the same USB stick to create a backup. Please use the following procedure to perform a backup

1. Insert USB-Stick into CPL410/CPE400, USB1 port.
2. Mount USB stick in Linux OS as described in **10.1. Mounting USB Stick**

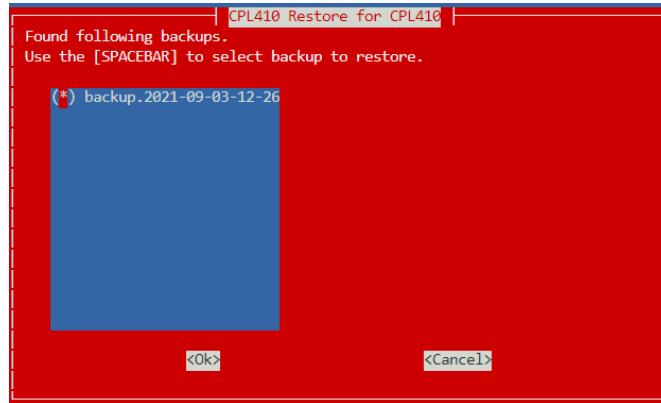
Now, within the Cockpit, go to the Terminal screen and enter the following commands:

Figure 158: Cockpit's Terminal



- a. `cd /mnt/usb/`
- b. `sudo ./cplcperestore`
- c. Enter the admin password (if applicable).
3. Using the arrow keys, select **Yes** and press **enter**.
4. Using arrow keys and space keys, mark the backup image you would like to restore, then with the Tab key, select **OK** and **enter**.

Figure 159: Select the Backup File



5. When restore starts, GPOK LED will turn off. It will remain off until the backup is complete, the PACEEdge side of the controller is rebooted, and the GPOK LED is ON. Backup can take up to 1h to complete. Observe SSD LED on the front panel, and restores are still being performed as long as it periodically flashes.

Before removing the USB stick, please properly un-mount it by the following procedure in **10.2 Un-Mounting USB Stick.**

6. Remove the USB stick.

NOTE: After PACEdge restore, you might have to Upload again Movicon project (if used) by using Movicon.NExT Editor.

Section 9: Upgrade from Earlier PACEEdge Versions

Beginning with PACEEdge version 2.2.0, software updates will be split into Linux OS updates and PACEEdge Application (Node-RED, Grafana, InfluxDB, MySQL, Connex, etc.) updates. This will enable the user to keep PACEEdge Applications without upgrading but continuously upgrade the host Linux OS with the latest security patches. All updates are self-contained and do not require an Internet connection when installing.

Given the differences in the hardware platform, the following sections will be split into RXi2-BP/LP and CPL410/CPE400 categories.

9.1 Upgrading RXi2-BP, RXi2-LP from PACEEdge 2.1.1 to 2.2.0

RXi2-BP and RXi2-LP units with PACEEdge 2.1.1 or PACEEdge 2.1 can be upgraded to PACEEdge 2.2.0. Since PACEEdge v2.1 does not have Navigator, the upgrade process is more tedious; therefore, it is highly recommended to start with version 2.1.1

Note: If upgrading from an older version PACEEdge you will need to first upgrade to version 2.1, preferably 2.1.1, before being able to upgrade to version 2.2.0. Upgrade procedures to version 2.1 and patch installation to v2.1.1 are documented in the older version of the PACEEdge User Manual and the older IPI documents, respectively. Still, these procedures are included below in this document for easier reference.

Alternatively, the user can use PACEEdge new version 2.2.0 Restore to factory default procedure, as described in section 8.2 PACEEdge Software Restore on RXi2-BP, RXi2-LP IPCs, and convert older systems directly to the v2.2.0, new factory installation. Important to note that this will delete any user data (Node-RED flows, Grafana dashboards, InfluxDB, and MySQL data), and it will require the user to install a license file after the upgrade procedure. When using this upgrade path, please first back up any user data. A valid license file can be obtained from Customer Care.

There are two methods to upgrade from PACEEdge version 2.1.1 to new version 2.2.0:

- Using Navigator to transfer the update files to the PACEEdge unit and execute the update. It is simpler and can be executed remotely, but it requires at least 7 GBytes of free space on the disc **/home** partition, which might not be available on the 32GB RXi2-LP version.

- Using a USB stick. This upgrade is also suitable for PACEEdge v2.1 systems, which do not have Navigator. It requires less disc space and physical access to the PACEEdge unit.

Note: latest PACEEdge upgrade utility was designed to upgrade PACEEdge applications while preserving standard user data, license files, and passwords. Automatically preserved data includes Node-RED flows and dashboards, Grafana dashboards, InfluxDB databases, MySQL databases, user passwords, PACEEdge license files, Linux OS libraries, and settings. Files that cannot be automatically preserved require data merging operation. This includes the files the user might have manually changed, such as the telegraf.conf file or docker-compose.yml file. If the original file cannot be upgraded to the new version without data loss, the original version of the file will be saved at: **/home/admin/pacedge22-update-backup/** location. The user will have to manually re-apply changes he made to the old file to the new file.

9.1.1 PACEEdge Upgrade using Navigator

Linux OS Upgrade (Navigator)

1. From the [Emerson Software Downloads](#) site, download **PACEEdge v2.2.0 OS Upgrade** (file name: **PEv220UBUpdate.zip**) file and unzip it on your local computer.
2. Login to Cockpit as the admin user, open Navigator, navigate to folder **/home/admin/**, and drag and drop a completely unzipped PEv220UBUpdate file folder from your computer to the Navigator. When asked for confirmation, please press the **Yes** button (at the bottom of the window).

Copy whole directory: /PEv220UBUpdate?

Note: This folder is roughly 600Mbyte and will take minutes to copy it fully. Wait until this folder shows up in your Navigator window.

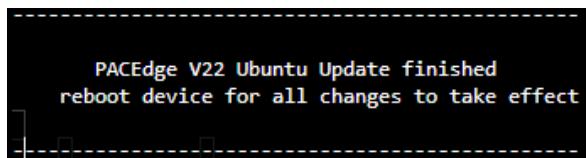
3. Change copied folder ownership by highlighting it in Navigator, then clicking on the settings icon on the right side  and change **Owner** and **Group** names from **root** to **admin** and click on the save button in the bottom right corner 

Figure 160: Permissions



4. Open the Terminal window and issue the following commands:
 - a. `cd /home/admin/PEv220UBUpdate`
 - b. `./installPEv220UBUpdate.sh`
 - c. When asked, enter the admin password
 - d. Depending on the HW, the OS update will take 10-40 min. When finished, you will see the following message.

Figure 161: Update Complete



5. If you are short on disc space for the next step, you can delete the original file folder with updated files. To do so, use Navigator, browse to `/home/admin/` folder, right-click folder **PEv220UBUpdate**, and choose the **Delete** option.

Note: At the screen's bottom, a message will pop up asking if you are sure. Please click on the **Yes** button.

Figure 162: Delete Confirmation



Note: After confirming folder deletion, a second question, asking if deletion shall be done recursively, will be asked. Please click on the Yes button.

Figure 163: Delete Recursively



6. Please reboot the system by issuing the command:

`sudo reboot`

Note: CPL410 and CPE400 need to be restarted by power cycling so that Hypervisor recognizes the OS changes

PACEEdge Application Upgrade (Navigator)

1. Before starting the upgrade, if you are running applications that are actively storing data in InfluxDB or MySQL, such as Node-RED flows, please stop these flows before starting the update.

IMPORTANT: Before starting an upgrade, please ensure that all PACEEdge Application passwords for the **admin** user role are the same, which means that the **admin** role has the same password in Node-RED, Grafana, including **Portainer**. Note that in PACEEdge version v2.1 and 2.1.1, to change the Portainer password user must execute a separate step within the Password Management utility. Cockpit/Linux passwords can be different.

2. From the [Emerson Software Downloads](#) site, download **PACEEdge v2.2.0 Application Upgrade** (file name: **PEv220AppUpdate.zip**) file and unzip it on your local computer.
3. Login to Cockpit as the admin user, open Navigator, navigate to folder **/home/admin/**, and from your computer, drag and drop a complete **PEv220AppUpdate** file folder to the Navigator. When asked for confirmation, please press the **Yes** button (at the bottom of the window).

Figure 164: Copy Whole Directory?



Note: This folder is very large (~4GBytes), and it will take minutes to copy it fully. Wait until this folder shows up in your Navigator window.

4. Change copied folder ownership by highlighting it in Navigator, then clicking on the settings icon on the right side and change **Owner** and **Group** names from **root** to **admin** and click on the save button in the bottom right corner

Figure 165: Permissions

Owner	admin
Group	admin
Mode	unchanged
	Read Write Execute
Owner	<input checked="" type="checkbox"/>
Group	<input checked="" type="checkbox"/>
Other	<input checked="" type="checkbox"/>

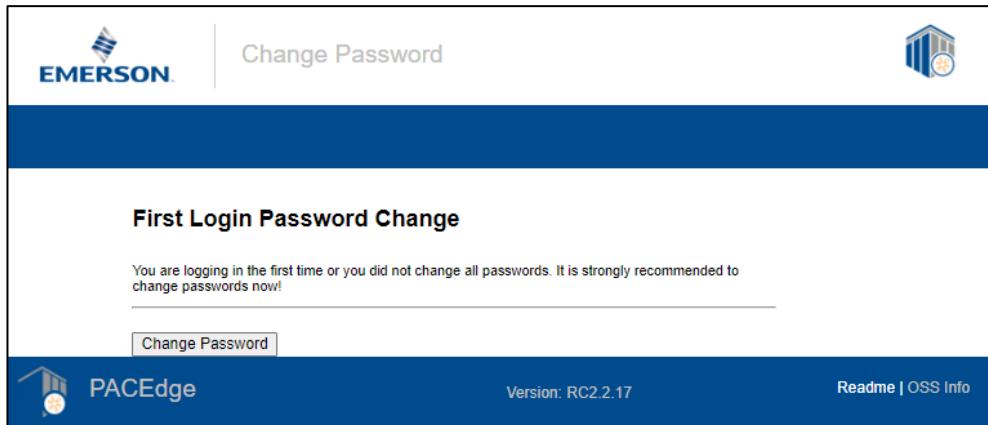
5. Open the Terminal window and issue the following commands:
 - a. **cd /home/admin/PEv220AppUpdate**
 - b. **./install.sh**
 - c. When asked, enter the admin password
 - d. Read through the warning message, then press Tab to highlight the <Yes> option and hit the enter key
 - e. Choose additional Backup options from the list, use the arrow keys to navigate, and use the space bar to select. Remember that some options require significant disc space (up to 5 GBytes). The only recommended option is to backup **Volumes**, which would create a backup of Node-RED, Grafana, InfluxDB, MySQL, and other application, user data volumes. Press the Tab key to select <Ok> and then hit enter key to continue
If asked, enter the admin password
 - f. If some configuration files have been modified from their original version, you will see the following message. Please select <Yes> to proceed.
 - g. Once the upgrade is complete, you will get the following message. Please select <Ok> and click enter.

Figure 166: Installation Successful



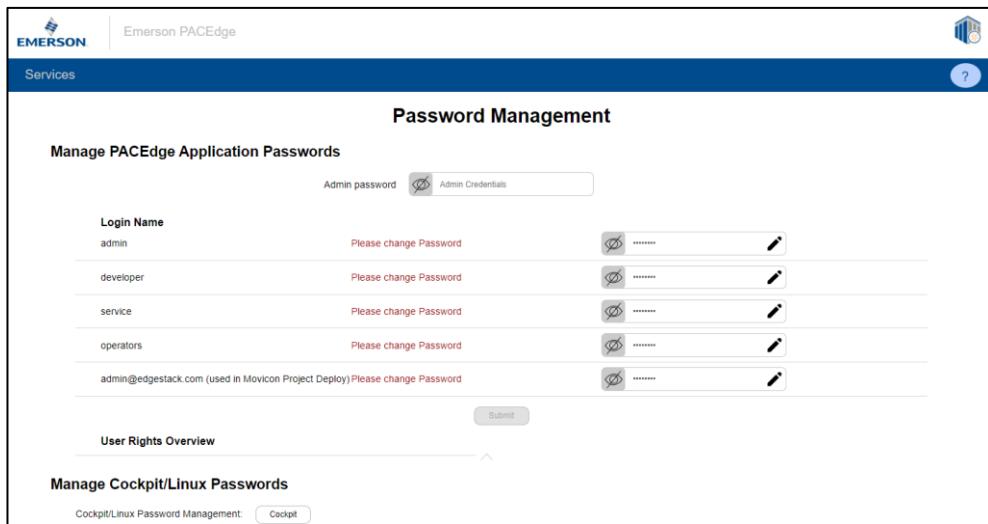
6. Next, please restart your browser and enter the IP address of your PACEEdge device in the new browser window. You will get a page asking to change passwords. This step is **MANDATORY** since we have added a new user and had to reset some existing passwords to their default values

Figure 167: Change Password



7. Login with your existing admin password, and you will get to a new, simplified Password Management page.

Figure 168: Admin Password



8. Please set all new passwords and notice that a new user role **service** has been added. This role is meant for the supplier service personnel at the customer site.

CAUTION

Do not use the dollar sign "\$" in a password. The dollar sign "\$" special character is incorrectly labeled as a supported character. Using "\$" can result in PACEEdge applications being inaccessible and unable to reach Grafana or Telegraf.

Note: newly redesigned Password Management utility will automatically change all PACEEdge application passwords (including Portainer, Telegraf, and project Deploy passwords within Movicon Editor). Linux/Cockpit passwords can change by the following link at the bottom, but for the PACEEdge upgrade, this is not necessary.

Note: if you get an Error message saying that the current password is incorrect, but you are sure it is correct, check to ensure that in the older version, you have set the Portainer admin password to be the same as the admin in other PACEEdge applications. If this is not the case, please correct it and try setting new passwords again.

To get to Portainer, bypassing the password change screen, enter PACEEdge device IP address followed by /portainer, such as *192.168.3.100/Portainer*

Note: since you have changed InfluxDB and MySQL passwords, you will have to update database access nodes in Node-RED flows, as well as in Grafana database mappings

9. Once the upgrade is complete, the following information is available:
 - a. In the file folder that the upgrade has been started from, a folder **pacedge22-update-backup** is created. It contains all the backup files that you chose when performing an upgrade
 - b. In the file folder from which the upgrade started, a file **modified files.lst** is created, which lists the files that had some custom modifications to them and could not be merged with the new version of the file. Most of these result from setting your password, which you can ignore, but if you made some manual customizations in, say, telegraf.conf file, you will have to apply the same customizations to the new telegraf.conf file.
If you have not manually modified these files, ignore them.
Note that for your reference, the original files will be stored in folder **/home/admin/pacedge22-update-backup/**, preserving the original file path.

Figure 169: Preserving original file path

```
modifiedfiles=(  
/home/admin/pacedge/.env  
/home/admin/pacedge/docker-compose.yml  
/home/admin/pacedge/emerson-grafana/config/grafana.ini  
/home/admin/pacedge/emerson-nginx/www/emerson/889e0d7343405c079195e7b8903c8c9e.php  
/home/admin/pacedge/emerson-nginx/www/emerson/version.css  
/home/admin/pacedge/emerson-telegraf/telegraf.conf  
/home/admin/pacedge/setup/docker-components/nginx.yml  
/home/admin/pacedge/setup/docker-components/progea.yml  
)
```

10. Cleanup by going to Cockpit->Navigator, browsing to folder: **/home/admin/**, and deleting both file folders with original update files:

PEv220UBUpdate

PEv220AppUpdate

To delete the folder, right-click on the folder name, and choose the **Delete** option. Confirm the delete operation by clicking the **Yes** button at the bottom right corner of the screen.

Note: At the screen's bottom, a message will pop up asking if you are sure. Please click on the **Yes** button.

Deleting `/home/admin/PEv220UBUpdate`.

This cannot be undone. Are you sure?

Note: After confirming folder deletion, a second question, asking if deletion shall be done recursively, will be asked. Please click on the Yes button.

Deleting `/home/admin/PEv220UBUpdate`.

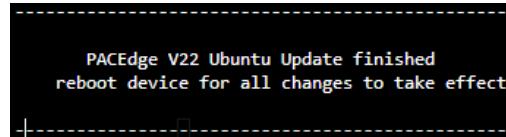
This cannot be undone. Are you sure?

9.1.2 PACEEdge Upgrade using USB Stick

If there is insufficient disk space left on the PACEEdge v2.1.1 or an upgrade is being done using the PACEEdge v2.1 device, a USB stick upgrade procedure can be used.

Linux OS Upgrade (USB)

1. From the [Emerson Software Downloads](#) site, download **PACEEdge v2.2.0 OS Upgrade** (file name: **PEv220UBUpdate.zip**) file and unzip it on your local computer
2. Copy a complete unzipped **PEv220UBUpdate** file folder to the USB stick and insert the USB stick into any USB port on the PACEEdge device
3. Login to Cockpit as the admin user, open the Terminal tab
4. If not already present, create a folder to mount the USB drive by issuing the following:
`sudo mkdir /mnt/usb`
5. Mount the USB stick so that owner is the **admin**, not the root, by issuing the following command:
`sudo mount -o uid=admin /dev/sdb1 /mnt/usb`
6. Change into the folder where update files are by issuing:
`cd /mnt/usb/PEv220UBUpdate/`
7. Start OS update by issuing:
`./installPEv220UBUpdate.sh`
8. Once finished



Reboot by issuing:

`sudo reboot`

Note: CPL410 and CPE400 units need to be restarted using a power cycle so that hypervisors do recognize OS updates.

PACEEdge Application Upgrade (USB)

1. Before starting the upgrade, if you are running applications that are actively storing data in InfluxDB or MySQL, such as Node-RED flows, please stop these flows before starting the update.
2. From the [Emerson Software Downloads](#) site, download **PACEEdge v2.2.0 Application Upgrade** (file name: **PEv220AppUpdate.zip**) file and unzip it on your local computer
3. Copy a complete unzipped **PEv220AppUpdate** file folder to the USB stick and insert the USB stick into any USB port on the PACEEdge device
4. Login to Cockpit as the **admin** user, open the Terminal tab
5. If not already present, create a folder to mount the USB drive by issuing the following:
`sudo mkdir /mnt/usb`
6. Mount the USB stick so that owner is the **admin**, not the root, by issuing the following command:
`sudo mount -o uid=admin /dev/sdb1 /mnt/usb`
7. Change into the folder where update files are by issuing:
`cd /mnt/usb/PEv220AppUpdate/`
8. Start PACEEdge Application update by issuing:
`./install.sh`
9. From here on, the upgrade procedure follows the same steps as when using Navigator. Please follow the same steps starting with
10. **PACEEdge Application Upgrade (Navigator) section d**
NOTE: When executing an update from the USB stick, all the chosen backup files will also be placed on the USB stick, further saving space on the PACEEdge device.

9.2 Upgrading Older RXi2-BP, RXi2-LP from PACEEdge 2.0 to 2.1.1

Note: this is an interim step when upgrading from the older PACEEdge 2.0 version to get to version 2.1.1, which then can be used to upgrade to version 2.2.0.

Note: if upgrading RXi2-BP or RXi2-LP from an older PACEEdge version, please consider the simpler restore to factory default option as described in section **8.2 PACEEdge Software Restore on RXi2-BP, RXi2-LP IPCs**. Note that this will result in a new version 2.2.0 installation from the factory. All your existing data, including licenses, will be lost. Please backup your data first and get new licenses from Customer Care afterward.

9.2.1 Upgrade PACEEdge v2.0 to PACEEdge 2.1 (old version)

Note: This upgrade procedure will delete all existing user data. PACEEdge 2.1 is substantially different from PACEEdge 2.0, which became prohibitive to automatically back up and restore user data (Node-RED flows, Grafana dashboards, InfluxDB, and MySQL databases). Therefore user needs to manually backup this data, as described in PACEEdge 2.0 User Manual, and then restore it in the new PACEEdge 2.1, as described in Section 7: Saving and Restoring User Data of (PACEEdge v2.1 User Manual).

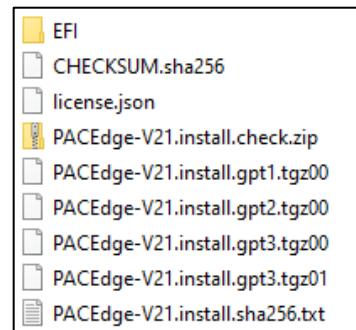
Please download the **PACEEdge 2.1 RXi2-BP RXi2-LP Software Image** from the [Emerson Software Downloads](#) site. Then follow the steps below to perform an upgrade:

1. Backup all user data (Node-RED flows, Grafana dashboards, MySQL, and InfluxDB databases) to a USB stick or network. The upgrade procedure will completely wipe clean internal storage
2. Get an empty, min 32GB size USB stick. Make sure it is empty, as conflicts might arise.

Note: If using old USB sticks, one might experience a problem in which the device may not boot from the USB stick. This is typically due to the file system changes on the USB stick. The workaround, in this case, is either: 1) use a brand new USB stick, 2) download from the Internet one of Linux installation ISO images, use a Windows utility such as Rufus, to make a bootable USB stick, then delete all the files from the USB stick and proceed with next steps.

3. Copy all files to the root directory of the USB stick; the directory structure should look like this:

Figure 170: Copy Files to the Root Directory of the USB Device



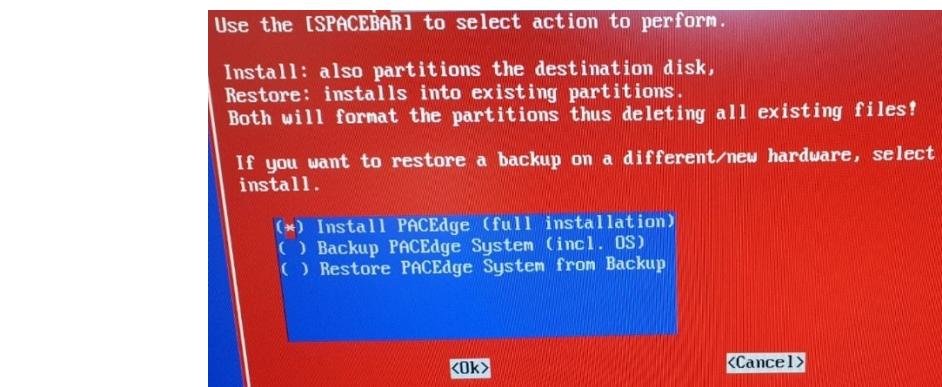
4. Plug the USB stick into any of the USB ports, power up the IPC, and keep pressing the F7 button to get into the boot selection menu; select the USB Stick to boot from (in the figure below, this would be the second line item: UEFI: SanDisk Extreme ...)

Figure 171: Select the Boot Device



5. Wait until the following dialog appears; use the arrow keys to move the cursor and the space key to select the Install PACEEdge option. Then use the Tab key to move the cursor to the **OK** button and hit enter.

Figure 172: Select the Backup



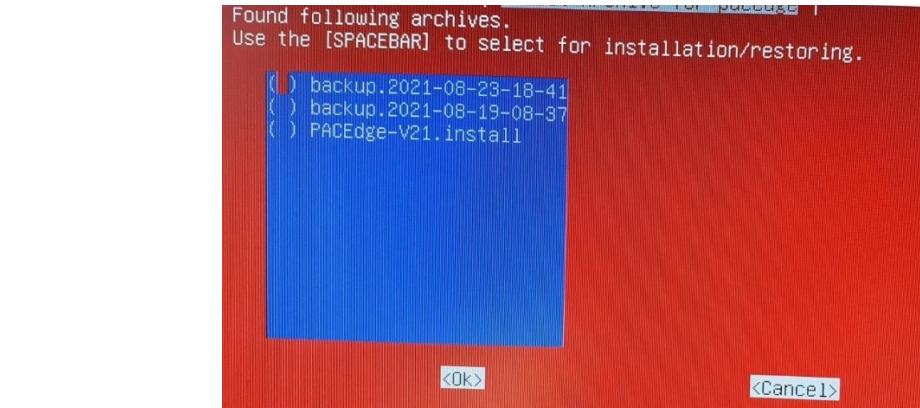
6. If asked for the disc partition, using arrow keys, space and Tab, select the first partition listed and click on **OK** to proceed.

Figure 173: Select the Boot Partition



7. Using arrow keys, space, Tab selects a factory default image (PACEEdge-V21.install), then select **OK** and hit enter.

Figure 174: Select a Factory Image



8. Wait until the installation procedure completes, as indicated by a similar message. Depending on the HW type, this step will take between a couple of minutes to 10-20min.
9. Restart the PACEEdge unit.
10. Once connected, follow the steps to change passwords, including the Portainer password. This step is mandatory for the upgrade to PACEEdge version 2.2.0 to succeed.

Figure 175: Restore Completed

```
Proceed anyway (or wait 5 seconds to proceed) ? (y,N) <proceeding>
Mounting /dev/sda1 on /mnt/gpt1
-----
Installing partition SW for gpt1 (taropt: --no-same-owner)
Mounting /dev/sda2 on /mnt/gpt2
[ 65.537267] EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts: (null)
-----
Installing partition SW for gpt2 (taropt: --xattrs)
Mounting /dev/sda3 on /mnt/gpt3
[ 117.847601] EXT4-fs (sda3): mounted filesystem with ordered data mode. Opts: (null)
-----
Installing partition SW for gpt3 (taropt: --xattrs)

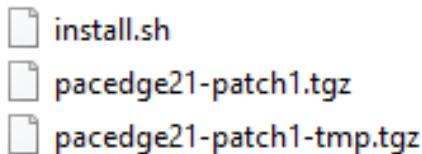
*****
*          PACEEdge Restore completed:      *
*    Thu Sep 2 11:26:13 UTC 2021           *
*    remove USB boot stick and power cycle now   *
*    see log file for more information        *
*                                              *
*****
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.0#
```

Note: The user will need to obtain and install a valid license as described in Section 3.7, *PACEEdge License File*

9.2.2 Apply Patch to PACEEdge v2.1 to get v2.1.1 (old version)

1. Download the patch version v2.1 Patch1 from the [Emerson Software Downloads](#) site (file: **PACEEdge-V21-Patch-1.zip**).
2. Unzip the files and copy the **PACEEdge-V21-Patch-1** folder to an empty USB stick (at least 8 GB in size).
 - a. File folder PACEEdge-V21-Patch-1 will have the following files:

Figure 176: Extract Files



- b. Insert the USB stick into any USB port on the PACEEdge device.
Note: On CPL410 and CPE400, use USB1 port
- c. Mount USB stick as described in: 10.1 Mounting USB Stick
- d. In Cockpit, go to the Terminal tab and execute the following commands:
`cp -r /mnt/usb/PACEEdge-V21-Patch-1/ /home/admin/
cd PACEEdge-V21-Patch-1/
.install.sh`
- e. In the open dialogue, use the Tab key to highlight <Yes> and press the enter key.
- f. When asked, enter the admin password.
- g. The installation will take a few minutes. Once complete, the user will see a dialogue box that states it was completed. Once the user highlights **OK** and presses the enter key, the user will be back in the Cockpit terminal window.
Note: The user will likely see many printout messages in the terminal window. Ignore the messages; this is an output from the script performing the installation and cleanup.
- h. Restart PACEEdge by going to **Overview** Tab and selecting **Restart** or by issuing the following command in the terminal window: **reboot**

9.3 Upgrading CPL410, CPE400 from PACEEdge 2.1.1 to 2.2.0

PACEEdge upgrade to version 2.2.0 can start with version 2.1.1 (preferred) or version 2.1.

PACEEdge version 2.1 does not have Navigator, which makes the upgrade process a bit more tedious and requires to use USB Stick upgrade method documented below.

If starting with a PACEEdge version older than 2.1, please first install PACEEdge 2.1 and then upgrade to 2.2.0

9.3.1 PACEEdge Upgrade using Navigator

Upgrade procedure to PACEEdge version 2.2.0 from version 2.1.1 for CPL410 and CPE400 is, in fact, identical to RXi2-BP and RXi2-LP, hence please use the procedure as documented in:

9.1.1 PACEEdge Upgrade using Navigator

9.3.2 PACEEdge Upgrade using USB Stick

Upgrade procedure to PACEEdge version 2.2.0 from version 2.1.1 for CPL410 and CPE400 is, in fact, identical to RXi2-BP and RXi2-LP, hence please use the procedure as documented in:

0

PACEEdge Upgrade using USB Stick

9.4 Upgrading CPL410, and CPE400 with Older PACEEdge to PACEEdge 2.1.1

This is an interim step required to upgrade CPL410, and CPE400, with older PACEEdge versions, to recent version 2.1 (2.1.1), which then can be used to upgrade to the latest version, 2.2.0

9.4.1 Upgrade PACEEdge v1 to PACEEdge 2.1

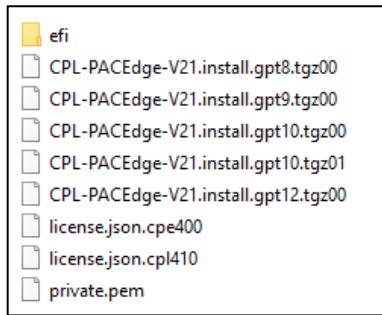
Note: Upgrading CPL410 and CPE400 from older Field Agent and Edge software versions will delete all data on the Linux side of the controller. The real-time PACs side will not be affected by the upgrade. Make sure to save user data to a USB stick or network storage.

Note: Before performing the PACEEdge update, please update PACs runtime firmware to the latest version. The Minimum PACs version required is 10.15

Download the **PACEEdge 2.1 CPL410 CPE400 Software Image** from the [Emerson Software Downloads](#) site. Then follow the steps below to perform an upgrade:

1. Get an empty, min 32GB size USB stick. Make sure it is empty, as conflicts might arise.
2. Note: if using old USB sticks, one might experience a problem in which the device may not boot from the USB stick. This is typically due to the file system changes on the USB stick. The workaround, in this case, is either: 1) use a brand new USB stick, 2) download from the Internet one of the Linux installation ISO images, use a Windows utility such as Rufus, make a bootable USB stick, then delete all the files from the USB stick and proceed with next steps.
3. Copy all files to the root directory of the USB stick. The directory structure should look like this:

Figure 177: Copy Files to Empty USB Device



4. Power down the unit, plug the USB stick into the USB1 port, and power up the unit
5. The unit will automatically perform an upgrade. Once the upgrade is finished, LEDs will show a pattern of running lights in the middle. At this point, please remove the USB stick and cycle the power to the unit
6. Once connected, follow the steps to change passwords, including the Portainer password. This step is mandatory for the upgrade to PACEEdge version 2.2.0 to succeed.

NOTE: The upgrade will not make any changes to the PACs real-time controller side; only the Linux side will be updated. After the update unit will have a valid license.

Apply Patch to PACEEdge v2.1 to get v2.1.1

1. Download the **PACEEdge 2.1 Patch 1** from the [Emerson Software Downloads](#) site (file: **PACEEdge-V21-Patch-1.zip**).
2. Unzip the files and copy the complete **PACEEdge-V21-Patch-1** folder to an empty USB stick (at least 8 GB in size).

3. File folder PACEEdge-V21-Patch-1 will have the following files:

Figure 178: Content of the PACEEdge-V21-Patch-1 folder



4. Insert the USB stick into any USB port on the PACEEdge device.

Note: On CPL410 and CPE400, use USB1 port

5. Mount USB stick as described in **10.1 Mounting USB Stick**

6. In Cockpit, go to the Terminal tab and execute the following commands:

```
cp -r /mnt/usb/PACEEdge-V21-Patch-1/ /home/admin/  
cd PACEEdge-V21-Patch-1/  
.install.sh
```

7. In the open dialogue, use the Tab key to highlight <Yes> and press the enter key.

8. When asked, enter the admin password.

9. The installation will take a few minutes. Once complete, the user will see a dialogue box that states it was completed. Once the user highlights **OK** and presses the enter key, the user will be back in the Cockpit terminal window.

Note: The user will likely see many printout messages in the terminal window. Ignore the messages; this is an output from the script performing the installation and cleanup.

10. Restart PACEEdge by going to **Overview** Tab and selecting **Restart** or by issuing the following command in the terminal window: **reboot**

Section 10: Software Utilities and Troubleshooting

10.1 Mounting USB Stick

PACEEdge can read and write data from/to a USB stick. However, the USB stick must first be properly mounted in Linux OS. The procedure to mount a USB stick is the same for RXi2-BP, RXi2-LP, CPL410, and CPE400; the only difference between systems will be the disc partitioning scheme that you see in Navigator; ignore inconsistencies with the images shown below.

To mount the USB stick, please follow the steps below:

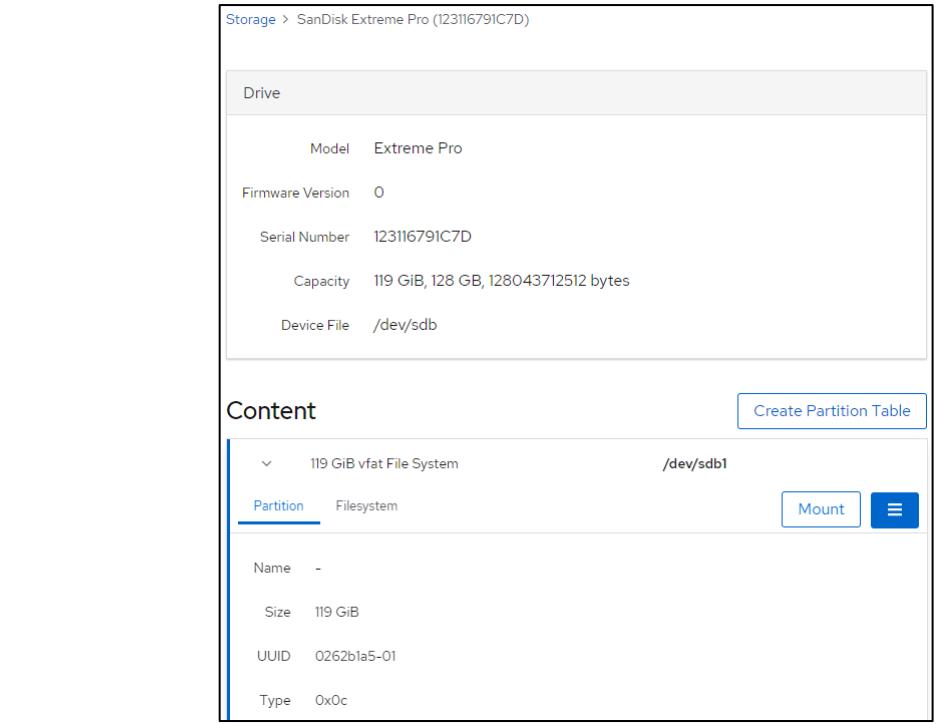
1. Insert USB-Stick into any USB port.
2. Login into Cockpit as admin.
3. Go to the Storage tab on the left side of the screen, look for the USB stick in the list of Filesystems (typically at the very bottom of the list, UBUNTU-SERV in the example below), and click on it.

Figure 179: List of Filesystems (CPL410, CPE400)

Filesystems		
Name ↑	Mount Point ↓	Size
/dev/loop0	-	55.4 MiB
/dev/loop1	-	70.3 MiB
/dev/loop2	-	32.3 MiB
/dev/loop3	-	32.3 MiB
/dev/loop4	-	55.4 MiB
/dev/loop5	-	69.9 MiB
/dev/sda1	/boot/efi	5.25 / 1022 MiB
/dev/sda2	/	6.29 / 38.9 GiB
/dev/sda3	/home	8.32 / 76.8 GiB
UBUNTU-SERV	/media/usb	13.2 / 119 GiB

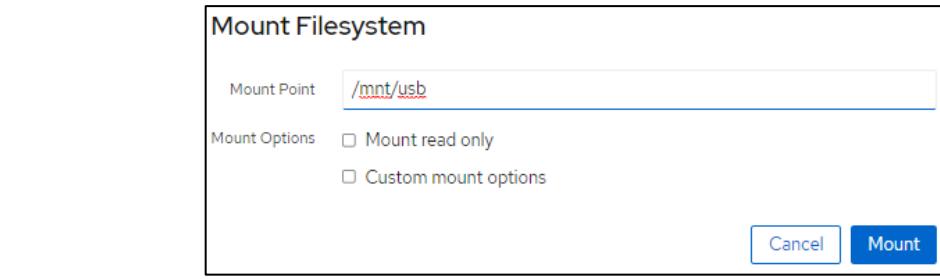
4. On the next page, click to expand the view of the Content and then click on the **Mount** button on the right side of the screen.

Figure 180: Mount the Device



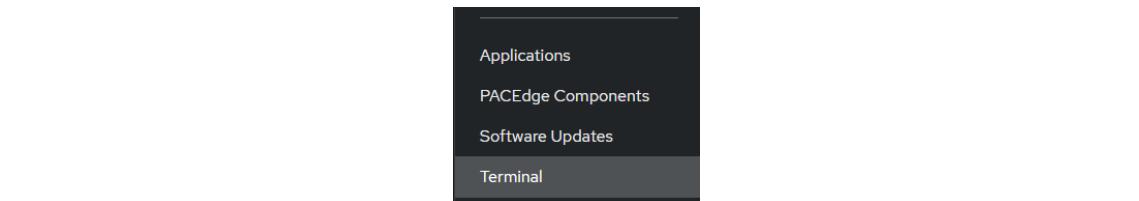
5. In the open dialogue, enter Mounting Point: **/media/usb** or **/mnt/usb** and click on the **Mount** button.

Figure 181: Mount the Filesystem



6. At this point, the USB stick is mounted and accessible. To see the files, you can use Navigator or go to the Cockpit->Terminal screen and enter the following commands:

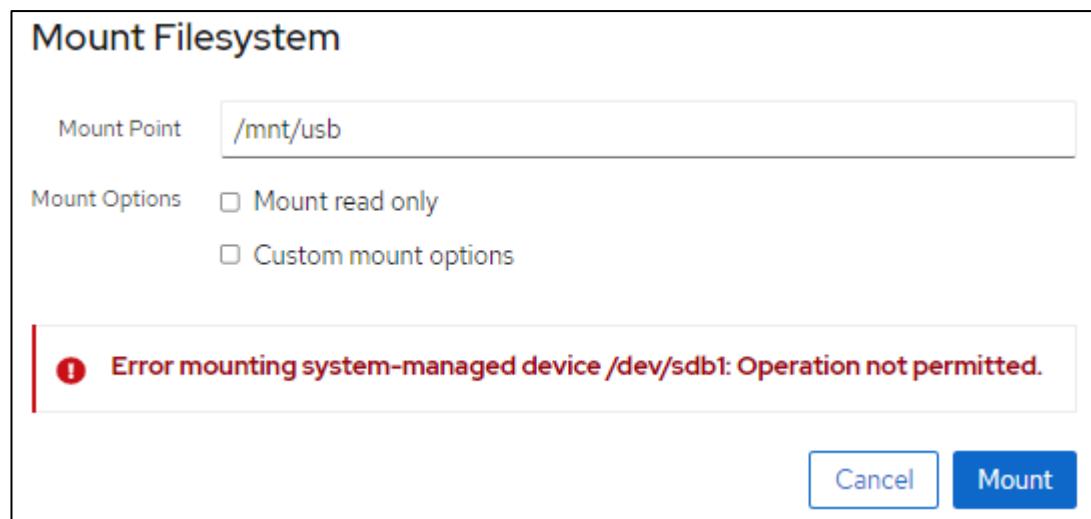
Figure 182: Cockpit's Terminal



-
- a. `cd /mnt/usb/`
 - b. `ll`

Note: If the USB stick has not been properly un-mounted in the past, it might lead to an error

Figure 183: Mount File System



In such case, entries in the/etc/fstab file need to be cleaned up as follows:

- a. Log in as **admin** into Cockpit
- b. Go to Terminal
- c. Type: **sudo nano /etc/fstab**
- d. Using arrow keys move your cursor behind the last character and delete all lines that start with UUID at the end of the file (last two lines in the given an example):

Figure 184: Example

```
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/disk/by-uuid/68a8b494-e44e-45ad-ab34-276eba199690 / ext4 defaults 0 0
/dev/disk/by-uuid/cb3c4787-ad33-4323-803e-e504d0dd3c35 /home ext4 defaults 0 0
/dev/disk/by-uuid/C564-B065 /boot/efi vfat defaults 0 0
/swap.img none swap sw 0 0
UUID=14F5-1B5F /mnt/usb auto defaults 0 0
UUID=1AF2-1D18 /mnt/usb auto defaults 0 0
```

- e. Press **CTRL+X** key combination, then the **y** key to confirm changes
- f. Try to mount the USB stick again.

10.2 Un-Mounting USB Stick

In Linux operating system, it is important to properly un-mount the USB stick before its removal.

To un-mount USB stick:

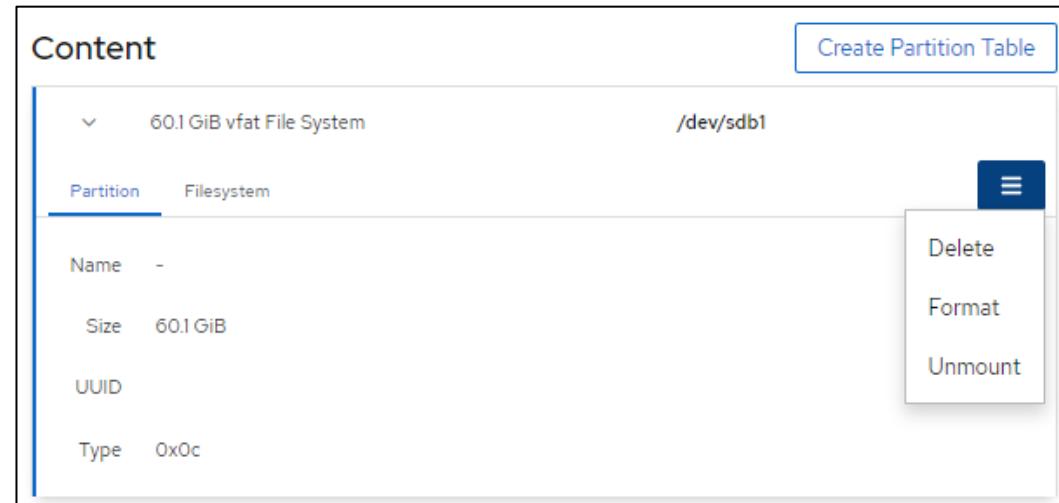
1. In Cockpit, go to the Storage tab
2. Click on the line which represents the USB stick. One way to identify it is by Mount Point being **/mnt/usb** or **/media/usb**

Figure 185: Select Storage to Unmount

Filesystems		
Name	Mount Point	Size
/dev/loop0	-	55.4 MiB
/dev/loop1	-	55.6 MiB
/dev/loop2	-	63.2 MiB
/dev/loop3	-	67.8 MiB
/dev/loop4	-	70.3 MiB
/dev/loop5	-	47.0 MiB
/dev/sda1	/boot/efi	5.25 / 1022 MiB
/dev/sda2	/	6.50 / 38.8 GiB
/dev/sda3	/home	10.2 / 767 GiB
SP UFD U3	/mnt/usb	3.04 / 60.0 GiB

3. Next, click on the disc within the Content window to open more details
Then, click on the three lines to open options and then click on Unmount

Figure 186: Unmount Storage



10.3 Difficulties Accessing PACEEdge Components, Erratic Behavior

- It is always a good first step to reboot the hardware.
- If you can log in, go to Cockpit and analyze the Logs. Note that logs contain several warnings and notices about different services as part of normal operation, so search for the messages related to the specific problem.
- If you have a problem accessing a specific PACEEdge application, say Node-RED, Grafana, or InfluxDB, either login into Cockpit, go to Docker Containers, or log into Portainer and check the status of the container that hosts the specific application.
Clues to look for:
 - Check if the container is not continuously restarting. You can see the last start time in the statement, such as: "Up since Today xxx."
 - In Portainer, look into the log file specific to each container for further clues.

For technical support, it is helpful to provide an exported Syslog file. To do that, please log in via a terminal window and execute **journalctl > log**.

10.4 PACEEdge Files

On Emerson IPCs with preinstalled PACEEdge, you will find the PACEEdge-Files in the Admin's home directory (**/home/admin/pacedge**). Expert users can use these files to adapt and modify the PACEEdge Docker environment to their needs and stop and start the PACEEdge system via the docker-compose command (see below). Normally there is no need to use these files.

10.4.1 Installation Test

PACEEdge comes with a script to test if PACEEdge installation is working as expected: `tst_inst.sh`. To execute this script, please change into folder: `/home/admin/pacedge/` and run the following command:

```
./tst_inst.sh -u admin -p your-admin-password
```

This script should produce an output similar to the following:

```
admin@pachedge:~/pachedge$ ./tst_inst.sh -u admin -p edgestacka
The Parameter IP address was set to: 127.0.0.1
The parameter user was set to: admin
The Parameter passwd was set to: edgestacka
start: OK
```

Wait 30 sec to finish the nodered start process

```
Check if all packages are installed
apache2-utils : OK
```

Check if all images are available

Tag	Container	Image Id	Size	Repository
<hr/>				
<hr/>				
Emerson-aes-daemon				iiot.aventics.com/images/aes-daemon
1.0.9-edgestack	eb2946d2bf57		182.7 MB	
emerson-chronograf				chronograf
1.8.8	2f07c735db5e		194.8 MB	
emerson-grafana				grafana/grafana
7.5.7	bc8c9ea5532e		201.8 MB	
emerson-influxdb				influxdb
1.8.6	66f3325c5416		307.8 MB	
emerson-mqtt				eclipse-mosquitto
2.0.11	c3a8baae5c3f		9.918 MB	

emerson-mqtt-internal-ipc	eclipse-mosquitto	
2.0.11	c3a8baae5c3f	9.918 MB
emerson-mysql	mariadb	
10.5.11	8bd09b203b08	407.5 MB
emerson-nginx		1.21.0
87560fc16fed	536.4 MB	
emerson-nodered	emerson-node-red	
1.3.5	2066beab39b7	612.5 MB
emerson-php	php	fpm
f8ceec02a25f	406.8 MB	
emerson-portainer	portainer/portainer-ce	
2.5.1	45be17a5903a	209.1 MB
emerson-progea		latest
a16ebaded4b0	3.713 GB	
emerson-telegraf	telegraf	
1.18.3	34535e6964f5	314 MB
emerson-traefik	traefik	
v2.4.8	deaf4b1027ed	91.31 MB
emerson-node-red	: OK	
portainer/portainer	: OK	
traefik	: OK	
grafana/grafana	: OK	
chronograf	: OK	
nginx	: OK	
influxdb	: OK	
eclipse-mosquitto	: OK	
mariadb	: OK	
emerson-progea	: OK	

Check if all docker-compose images are available

Check if the containers are running

emerson-nodered	: OK
emerson-portainer	: OK
emerson-traefik	: OK
emerson-grafana	: OK
emerson-nginx	: OK
emerson-chronograf	: OK
emerson-mysql	: OK
emerson-mqtt	: OK
emerson-mqtt-internal-ipc	: OK

```
emerson-influxdb : OK
emerson-progea : OK
```

Check if Cockpit is started and running
cockpit.socket : OK

Check if node-red palettes are installed correctly
To be done ...

Check if WEB-interfaces are accessible
/nodered/ 1880: OK
/grafana/ 3000: OK
/chronograf/ 8888: OK
/portainer/ 9000: OK
:9090/cockpites/ 9090: OK
/emerson/eula.html 9999: OK

Number of errors 0
OK

Note: an early version of this script has a bug and checks for the nodered/node-red container image, which is an interim product in building a final emersonnode-red image and is removed later in the build process. Please ignore this error:

emerson-node-red : OK
nodered/node-red : ERROR
portainer/portainer : OK

10.5 Docker Commands

PACEEdge is heavily based on Docker and Docker application images. Docker is a kind of lightweight virtualization allowing to group applications together in a protected self-contained environment within the Linux operating system. Setting up and configuring such a set of applications is a complex task whose description is beyond the scope of this document. Fortunately, this arrangement is already mastered by PACEEdge and you only need some commands to manage PACEEdge.

PACEEdge uses “docker -compose” to start and configure the set of PACEEdge applications. The configuration for “docker-compose” is stored in a Yaml file: **docker-compose.yml**. It comes with the installation package.

To view docker-compose.yml file and execute other docker commands please first change into directory: **/home/admin/pacedge/**

Following are the most important commands:

10.5.1 docker-compose up -d

Note that with PACEEdge version 2.2.0, there is no longer dash in between docker and compose words. With this command, the PACEEdge environment is started. All the containers configured in the **docker-compose.yml** file are created, configured, started, and connected. Normally, this command is not needed, as the PACEEdge environment is automatically started during system start. But it is used whenever the configuration has changed or the following command has shut down PACEEdge:

10.5.2 docker-compose down

Note that with PACEEdge version 2.2.0, there is no longer dash in between docker and compose words. Use this command whenever you want to shut down PACEEdge. This command will delete all containers, and data is stored within the containers, but persistent data will remain in external volumes. This command is useful if you want to exclude an application from the PACEEdge configuration (see the following command).

10.6 ./config-compose.sh

The script config-compose.sh, located in the */home/admin/paceedge/*, let you select the PACEEdge applications you want to run. By default, all applications are activated, but if you don't need some of them, you can deselect it in this semi-graphic dialogue. The dialogue will guide you through bringing down all docker containers, allowing you to add/remove docker containers, and then bringing up containers. Config-compose.sh generates a new **docker-compose.yml** configuration file (see above).

The user executing these commands must be a member of the Linux "docker" group. Preinstalled PACEEdge user "admin" is a member of this group and therefore recommended for these actions.

General Contact Information

Home link: <http://www.emerson.com/industrial-automation-controls>

Knowledge Base: <https://www.emerson.com/industrial-automation-controls/support>

Technical Support

Americas

Phone: 1-888-565-4155
1-434-214-8532 (If toll-free option is unavailable)

Customer Care (Quotes/Orders/Returns): customercare.mas@emerson.com
Technical Support: support.mas@emerson.com

Europe

Phone: +800-4444-8001
+420-225-379-328 (If toll-free option is unavailable)

Customer Care (Quotes/Orders/Returns): customercare.emea.mas@emerson.com
Technical Support: support.mas.emea@emerson.com

Asia

Phone: +86-400-842-8599
+65-6955-9413 (All other Countries)

Customer Care (Quotes/Orders/Returns): customercare.cn.mas@emerson.com
Technical Support: support.mas.apac@emerson.com

Any escalation request should be sent to mas.sfdcescalation@emerson.com

Note: If the product is purchased through an Authorized Channel Partner, please contact the seller directly for support.

Emerson reserves the right to modify or improve the designs or specifications of the products mentioned in this manual at any time without notice. Emerson does not assume responsibility for the selection, use, or maintenance of any product. Responsibility for the proper selection, use, and maintenance of any Emerson product remains solely with the purchaser.

© 2023 Emerson. All rights reserved.

Emerson Terms and Conditions of Sale are available upon request. The Emerson logo is a trademark and service mark of Emerson Electric Co. All other marks are the property of their respective owners.

