

# PACEdge™

## USER MANUAL



# Contents

<b>Section 1: Introduction.....</b>	<b>1</b>
1.1 Revision History.....	1
<b>Section 2: Physical Connections and Configuration Settings..</b>	<b>2</b>
2.1 Network, Keyboard, Display.....	2
2.2 Power.....	2
<b>Section 3: PACEdge Getting Started.....</b>	<b>3</b>
3.1 PACEdge Usage Models.....	3
3.2 PACEdge in a Direct-Use Configuration.....	3
3.2.1 Getting Started.....	3
3.3 PACEdge in Headless Configuration.....	4
3.3.1 Getting Started.....	4
3.4 Initial Login.....	5
3.5 PACEdge Users, Rights, and Passwords.....	7
3.5.1 Changing Passwords via Automated Utility.....	9
3.5.2 Changing Passwords Manually (Portainer, Cockpit).....	10
3.5.3 Changing Connexxt Password.....	12
3.5.4 Changing Telegraf Database Password.....	14
3.6 PACEdge System Level Settings.....	16
3.6.1 System Configuration Changes via Cockpit.....	16
3.6.2 Physical – Logical Ethernet Port Mapping.....	17
3.7 PACEdge License File.....	18
3.7.1 Licensed Connexxt and WebHMI Features.....	19
<b>Section 4: PACEdge Architecture Details.....</b>	<b>21</b>
4.1 PACEdge Software Services.....	21
4.1.1 Application Development Components.....	23
4.1.2 Database Components.....	24
4.1.3 Administration Components.....	24
4.2 PACEdge Communications and Data Flow.....	26
4.2.1 PACEdge Internal Communication Parameters.....	27
4.2.2 PACEdge External Communication Parameters.....	28
4.2.3 Accessing Connexxt OPC UA Server.....	29
4.2.4 PACEdge Suggested Data Flows.....	31
4.2.5 Southbound Communication Capabilities.....	32
4.2.6 North Bound Communication Capabilities.....	32

4.2.7 PACEdge Internal Communications and Data Flow ..... 33

**Section 5: PACEdge Usage Examples.....34**

5.1 Node-RED Getting Started ..... 34  
5.1.1 Example Description ..... 34  
5.1.2 Prerequisites ..... 34  
5.1.3 Example Walkthrough ..... 34  
5.2 Using the InfluxDB Database ..... 39  
5.2.1 Example Description ..... 39  
5.2.2 Prerequisites ..... 39  
5.2.3 Example Walkthrough ..... 39  
5.3 Using MySQL Database ..... 42  
5.3.1 Example Description ..... 42  
5.3.2 Prerequisites ..... 42  
5.3.3 Example Walkthrough ..... 42  
5.4 Using MQTT Communication ..... 44  
5.4.1 Example Description ..... 44  
5.4.2 Prerequisites ..... 44  
5.4.3 Example Walk Through ..... 44  
5.5 Using Telegraf and Grafana ..... 47  
5.5.1 Example Description ..... 47  
5.5.2 Prerequisites ..... 47  
5.5.3 Example Walk Through ..... 47  
5.6 Using Connex and WebHMI ..... 53  
5.6.1 Prerequisites ..... 53  
5.6.2 Example Walk Through ..... 53  
5.7 Configuring Telegraf ..... 61  
5.7.1 Example Description ..... 61  
5.7.2 Prerequisites ..... 61  
5.7.3 Example Walk Through ..... 61

**Section 6: PACEdge Performance Indication .....63**

6.1 OPC UA Performance Indication ..... 63  
6.2 Node-RED Dashboard Performance Indication ..... 64  
6.3 Node-RED Modbus TCP Performance Indication ..... 66  
6.4 Node-RED Connex Historian Performance Indication ..... 66

**Section 7: Saving and Restoring User Data .....67**

7.1 Exporting and Importing Flows in Node-RED ..... 67  
7.1.1 Export Flow ..... 67

7.1.2	Import Flow.....	67
7.2	Exporting and Importing Views in Grafana .....	68
7.2.1	Export Flow .....	68
7.2.2	Import Flow.....	68
7.3	Importing Projects in Connex/WebHMI .....	68
7.4	Backing Up and Restoring InfluxDB and MySQL Databases .....	69
7.5	Saving License Files.....	69
<b>Section 8: PACEdge Software Backup/Recovery.....</b>		<b>72</b>
8.1	PACEdge Software Backup on RXi2-BP, RXi2-LP IPCs .....	72
8.2	PACEdge Software Restore on RXi2-BP, RXi2-LP IPCs.....	74
8.3	PACEdge Software Backup on CPL410, CPE400 Controllers.....	76
8.4	PACEdge Software Restore on CPL410, CPE400 Controllers .....	79
8.4.1	CPL410, CPE400 PACEdge Recovery to Factory Default .....	79
8.4.2	CPL410, CPE400 Restore of PACEdge Backup Image .....	80
<b>Section 9: Upgrade from Earlier PACEdge Versions .....</b>		<b>84</b>
9.1	Upgrading RXi2-BP, RXi2-LP from PACEdge 2.0 to 2.1 .....	84
9.2	Upgrading CPL410, CPE400 from Older Versions to PACEdge 2.1 .....	88
<b>Section 10: Troubleshooting .....</b>		<b>89</b>
10.1	Difficulties Accessing PACEdge Components, Erratic Behavior .....	89
10.2	Error in node-red-contrib-emerson Examples.....	89
10.3	PACEdge Files .....	90
10.3.1	Installation Test.....	90
10.4	Docker Commands.....	93
10.4.1	docker-compose up -d .....	93
10.4.2	docker-compose down.....	93
10.5	./config-compose.sh .....	94
General Contact Information .....		95
Technical Support.....		95

## 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 as to 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 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

PACEdge processing maximizes the value of your data by improving operational reliability, safety, and energy consumption. PACEdge provides all aspects of edge processing and simplifies your IIoT application development, deployment, and administration. All components necessary in the IIoT application lifecycle are brought together 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
D	Dec 2021	Added details on Accessing the Connex OPC UA Server (Section 4.2.3)
C	Sep 2021	PACEdge 2.1 Initial release
B	Nov 2020	Added Section 7.2 Node RED Dashboard Performance Indication
A	Sep 2020	PACEdge 2.0 Initial release

# Section 2: Physical Connections and Configuration Settings

## 2.1 Network, Keyboard, Display

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

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

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

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

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

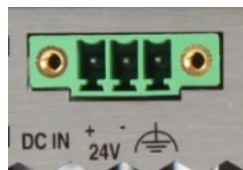
Optionally (RXi2-LP, RXi2-BP), attach a display to the DP++ port.

**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

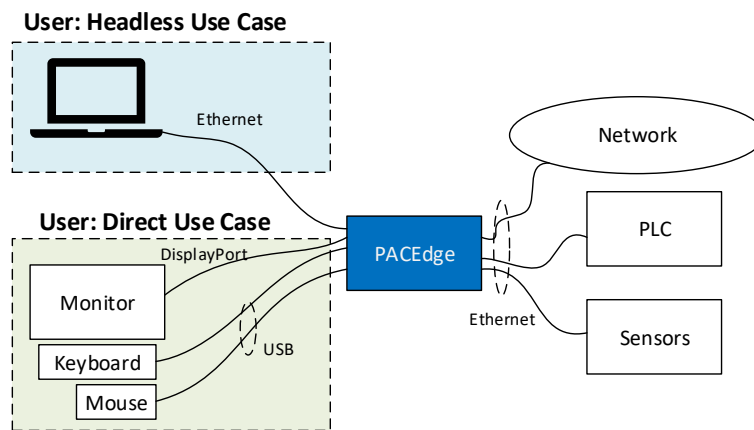
**Note:** PACEdge software comes pre-installed on Emerson Industrial PCs and Controllers.

## 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 in a headless mode, with the user accessing 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. 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 by 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**.
5. **Note:** The user will be asked to change the default password to a unique password at the first login.

6. Most interactions with PACEdge are done via a browser-based interface. To get started, click on **Activities**->**Show Applications** and start the Firefox browser.
7. Within the Firefox browser, go to <https://localhost>
8. Proceed to the section entitled 3.4Initial 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 also configured to get IP addresses assigned by the DHCP server. This dynamically assigned IP address can also be used to access PACEdge. In case the DHCP server also supports DNS service, PACEdge unit dynamically assigned IP address can be obtained by using command shell and **ping pagedge** command.

5. Proceed to Section 3.4, Initial Login.

## 3.4 Initial Login

When the user connects to PACEdge for the first time (using the instructions outlined in *PACEdge in a Direct-Use Configuration* or *PACEdge in Headless Configuration*), a warning message will display due to the self-signed certificate on the unit. The message will state that the identity of the device could not 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 Services screen. PACEdge software consists of multiple software tools and applications, each having its user management and password settings. To simplify user and password management tasks, PACEdge comes with pre-configured users: *admin*, *developer*, and *operators*, each having its password. Use the Password Services screen to change passwords for the whole group of services. This will be displayed on the left side of the screen. (Figure 3).

**Note:** When you click on Apply Changes button and then following on the Restart button, web services that are 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, please manually enter [https://ip\\_address/pw/](https://ip_address/pw/) in the browser address field to return to the Password Services page. The restart will take up to 60 seconds.

Additionally, change the Portainer and Cockpit/Linux passwords individually (

Figure 4). In Portainer, by default, the only *admin* user is created. You can create additional users as required.

In Cockpit please change passwords for *admin*, *developer*, and *operators* users individually.

**Note:** In Cockpit, do not set any passwords for user *Root*. For security purposes this user is disabled and setting a password for it would enable it, which is not recommended.

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

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

**Figure 3: Password Services Screen (The Left Side will Display the User Levels and change password shortcuts for each level)**

EMERSON | Emerson PACEdge

Services

**Attention:** For security purposes all default passwords shall be changed!

Group Managed Profile Services  
Node-RED, InfluxDB, MySQL, Chronograf, Grafana Password Management

User	Node-Red	InfluxDB	MySQL	Chronograf	Grafana	Update Password
admin	R/W	R/W	R/W	R/W	admin rights	Update
developer	R/W	R/W	R/W	R/W	editor rights	Update
operators	R	-	-	-	viewer rights	Update

**Figure 4: Password Services Screen (The Right Side will Display the shortcuts to change Portainer and Cockpit/Linux Passwords)**

EMERSON | Emerson PACEdge

Individually Managed Profile Services  
Portainer, Cockpit/Linux Password Management

User	Portainer	Cockpit/Linux
admin	R/W	sudo
developer	R/W	sudo
operators	-	limited

Update Psw. Update Psw.

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

**Figure 5: PACEdge Landing Page**

EMERSON | Emerson PACEdge

Settings

PACEdge Applications

- Node-RED: Data Processing
- Grafana: Data Visualization
- Movicon WEBHMI: Process Visualization
- COCKPIT: Manage Linux
- portainer.io: Manage Docker
- chronograf: Manage InfluxDB

PACEdge | Version: R12.1.0 | Username: EMERSON | 2021-10-14

To explore PACEdge, click on Node-RED and log in with:

- **user:** admin
- **password:** edgestack (or the password that was set on the Password Services page)

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

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

## 3.5 PACEdge Users, Rights, and Passwords

The passwords of PACEdge shall be changed at the first login. Emerson strongly recommends the use of 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 PACEdge each of the following applications has its users and user credential management interfaces:

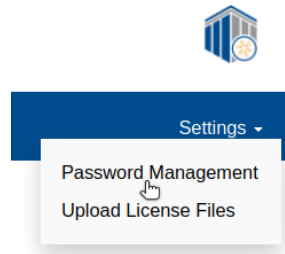
**Table 1: Default Passwords**

Functionality	Authenticated Subjects (user)	Default Passwords
SSH remote login	admin	Auto-prompted upon the first login to change the password, the default password is "edgestack"
Cockpit	admin	Cockpit uses the Linux password, the default password is "edgestack", auto-prompted upon the first login to change the password.
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"
Connex/WebHMI	admin@edgestack.com	Default password: "Edgestack123!"

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

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

Figure 6: Password Management



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

User Name: **admin**

Password: **edgestack**

Once on the password services page (Figure 7) on the left are the options to change passwords for each user role, automatically across all listed services. On the right, are the shortcuts to change passwords for the remaining services, namely Portainer and Cockpit.

Figure 7: Password Services Page

Services ?

**Attention:** For security purposes all default passwords shall be changed!

Group Managed Profile Services  
Node-RED, InfluxDB, MySQL, Chronograf,  
Grafana Password Management

User	Node-Red	InfluxDB	MySQL	Chronograf	Grafana	Update Password
admin	R/W	R/W	R/W	R/W	admin rights	Update
developer	R/W	R/W	R/W	R/W	editor rights	Update
operators	R	-	-	-	viewer rights	Update

Individually Managed Profile Services  
Portainer, Cockpit/Linux Password  
Management

User	Portainer	Cockpit/Linux
admin	R/W	sudo
developer	R/W	sudo
operators	-	limited
Update Psw		Update Psw

Color Codes:

- Factory default Password In Use. For Security Purposes Please Change.
- Password Updated. For some Services a restart is required to apply changes.
- No Changes Needed or Pending.

The table indicates what rights each user role has in each of the services, R indicating Read, W indicating Write, and symbol (-) indicating no access rights. For Grafana 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/](https://grafana.com/docs/grafana/latest/permissions/organization_roles/).

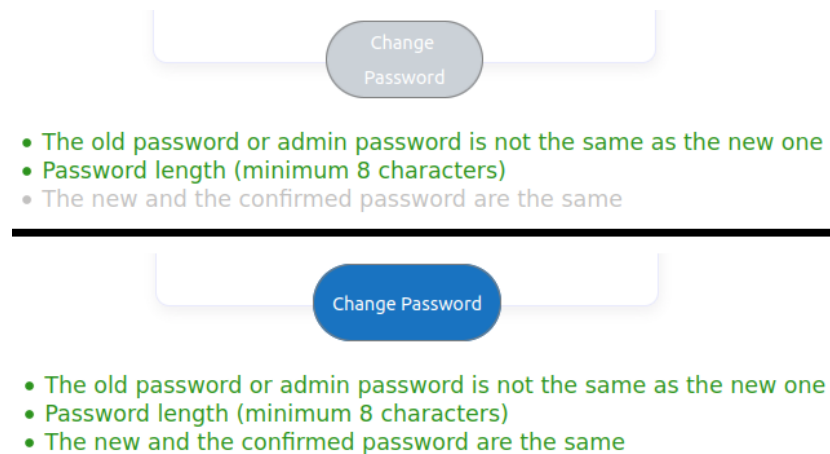
Initially, all buttons on the Password Services page will be colored red, indicating that the user shall take an action and change the password. Once the password change process is in progress button color will turn yellow and once complete it will turn grey.

**Note:** it is highly recommended to change passwords by using provided automated utility as opposed to 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 change utility will fail.

### 3.5.1 Changing Passwords via Automated Utility

Using functionality on the left side of the page, after clicking on the **Update** button, the current **admin** password has to be provided. Then the new password can be defined. The Hints dialog will turn green when the stated requirement is met. When all requirements are met the **Change Password** button color turns from grey to blue and is clickable, see (Figure 8).

**Figure 8: Hints Dialog**



Once the **Change Password** button is clicked, some additional checks will be performed, and if no error, the main password management page will be displayed. The most common reason for the occurrence of an error is that the given **admin** password was not correct. This happens most often when the **admin** password has already been changed, then from this point on the new password must be used to manage the other users.

**Note:** After changing the admin password the **NEW** password must be used for managing the other users.

**Figure 9: Admin Password Change in Progress**

User	Node-Red	InfluxDB	MySQL	Chronograf	Grafana	Update Password
admin	R/W	R/W	R/W	R/W	admin rights	Pending Restart
developer	R/W	R/W	R/W	R/W	editor rights	Update
operators	R	-	-	-	viewer rights	Update
						Apply changes

Note that once one of the password changes has been started, the button color turns to yellow and a new button **Apply Changes** appears. To complete the password change process, the user must click on **Apply Changes**, but it is more efficient to first change passwords for all three user roles and then click on **Apply Changes**.

Notice that clicking on **Apply Changes** will restart the number of containers, which will result in communication loss as indicated by the warning page that shows up. As soon as the **Restart** button is clicked, the restart of the containers will be triggered.

**Note:** If using Chrome, the connection will be lost. Please wait 60 seconds, then manually enter in the browser *ip\_address/pw*, such as <https://192.168.3.100/pw> if the default IP address is being used.

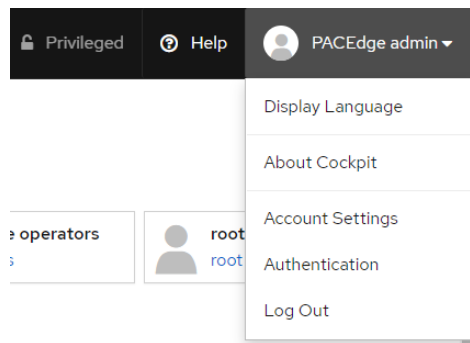
### 3.5.2 Changing Passwords Manually (Portainer, Cockpit)

Please note that the Linux and Portainer users must be managed separately.

**Note:** The automated password change utility, described in the previous chapter, does not change Cockpit/Linux and Portainer user passwords. The default passwords are **edgestack**.

The easiest way to manage the Cockpit/Linux users is to click on the Cockpit/Linux **Update Password** button and log in with the admin credentials. If this is the first login, the user will be prompted to change the default **admin** password. After changing the admin password, log out and log in into 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 upper-right corner of the screen.

Figure 10: PACEdge Admin Menu

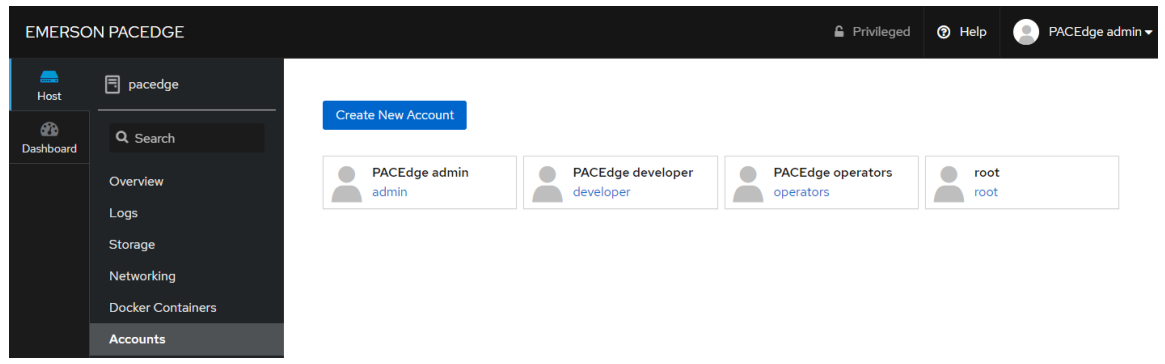


1. Once logged in as admin, set the passwords for the developer and operators roles.

#### **⚠ CAUTION**

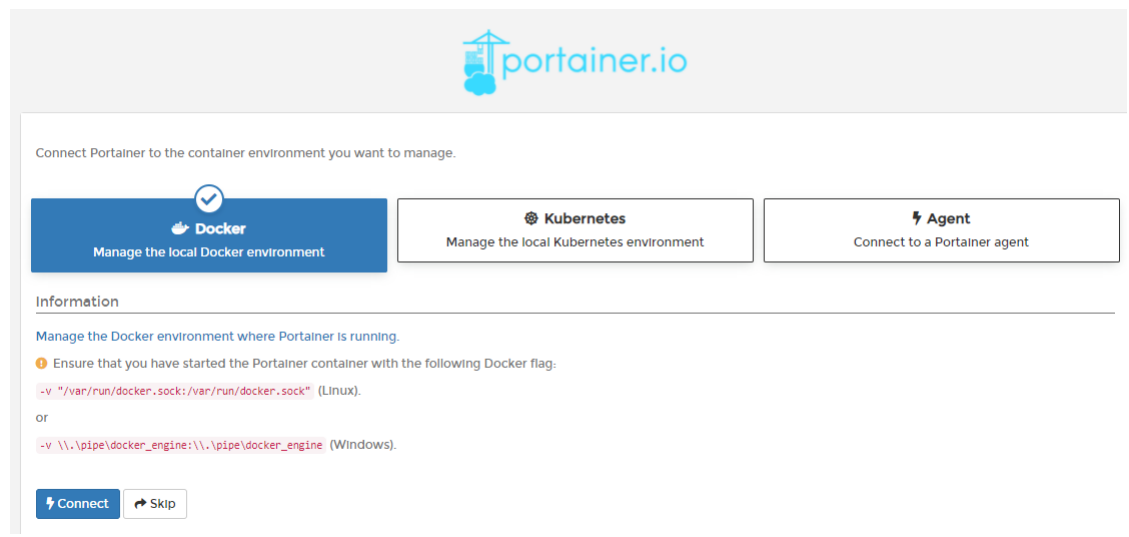
Do not set any password for the root user, as this would enable the root account and create a cybersecurity vulnerability.

Figure 11: User Profiles



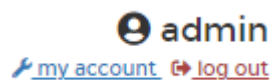
- To change the Portainer password, click on the **Update Password** button under Portainer and log in with admin credentials. Once logged in, click on the **Docker** option and then on **Connect** button.

Figure 12: Portainer



- To change passwords, click on the **my account** link in the upper-right corner.

Figure 13: My Account



Then enter the current admin password and new password.

**Figure 14: Enter Credentials**

Change user password

Current password

New password

✖ Your new password must be at least 8 characters long

Confirm password

Update password

For Portainer, only the admin user is preconfigured. If needed, a new user can be created and the rights can be granted to this user. It is not recommended to create a user for the user role **operators**. Changing the password for Portainer is outlined in the Section [Portainer](#).

### 3.5.3 Changing ConnexT Password

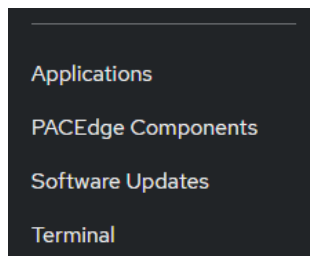
The ConnexT and WebHMI applications must first be developed using the Movicon.NExT software on the user's workstation and then uploaded to a remote unit that is running PACEdge software. Connectivity between Movicon.NExT and remote unit is protected by username and password, which should be changed to user-specific values.

#### Changing ConnexT Password on PACEdge Device

To change the ConnexT and WebHMI password on PACEdge:

1. Log in to Cockpit as the admin.
2. Go to the Terminal option.


**Figure 15: Cockpit Settings**



3. Type: `nano pacedge/emerson-progea/DeployServer-Files/appsettings.json`
4. Edit the username and password as required. Use the arrow keys to move the cursor around while editing. Keep the values enclosed in quotes as shown in Figure 16.

---

**Figure 16: Terminal Example**



```
"Login": {  
  "User": "admin@edgestack.com",  
  "Password": "Edgestack123!"  
},
```

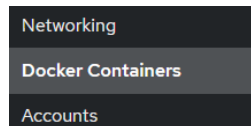
5. Once finished editing, save changes as follows:
  - a. CTRL + x
  - b. Y (to Save modified buffer?)
  - c. Enter (keeping same file name)
6. Type: `sudo rm pacedge/emerson-progea/DeployServer-Files/DeployServerDB.db`
7. Confirm the admin password when prompted.
8. Restart the hardware device.

Alternatively, restart just the Progea container as follows:

- a. In Cockpit, go to the Docker Containers option

---

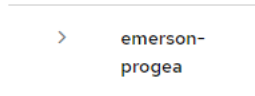
**Figure 17: Docker Containers**



- b. Click on the **emerson-progea** container:

---

**Figure 18: Emerson-Progea Container**

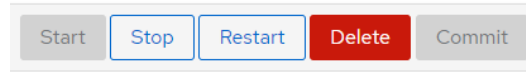


A screenshot showing a list of containers. The container named "emerson-progea" is selected, indicated by a chevron symbol (>) to its left.

- c. In the upper right corner, click on the **Restart** button:

---

**Figure 19: Restart Button**



9. After the container restarts, start using the new username and password in Movicon.NEXt application when connecting to the device with PACEdge.

### 3.5.4 Changing Telegraf Database Password

Telegraf, besides other data, can also be used to store system statistics and logs into an InfluxDB database. PACEdge comes with Telegraf pre-installed and pre-configured to store local IPC statistics into the InfluxDB database. Access to the InfluxDB is granted by using a username and a password, which needs to match the username and password that was configured when changing passwords for InfluxDB. The following procedure is used to change the InfluxDB username and password within Telegraf:

1. From the PACEdge landing page open Cockpit and login as the admin.
2. Go to the Terminal application and type in the following command (enter admin password if requested):

---

**Figure 20: Open the Telegraf Configuration File**

```
admin@pachedge:~$ sudo nano pachedge/emerson-telegraf/telegraf.conf  
[sudo] password for admin:
```

3. The command will open the Telegraf configuration file. To navigate the cursor, use the arrow keys on your keyboard. The database name in InfluxDB (telegraf\_metrics) will have all the saved data. Note the username and password for that user. The password in this file is stored in clear text; hence, do not use the admin username and password. If desired, the user can create a dedicated user inside the InfluxDB for Telegraf and use that username and password. In this example, however, we use the developer user and the password that was given to the developer user.

4. Once edits are done,
  - a. CTRL + x
  - b. Y (to Save modified buffer?)
  - c. Enter (keeping same file name)

---

**Figure 21: Select the User Profile to Change**

```
### OUTPUT

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

## Retention policy to write to. Empty string writes to the default rp.
retention_policy = ""
## Write consistency (clusters only), can be: "any", "one", "quorum", "all"
write_consistency = "any"

## Write timeout (for the InfluxDB client), formatted as a string.
## If not provided, will default to 5s. 0s means no timeout (not recommended).
timeout = "5s"
username = "developer"
password = "edgestackd"
```

5. Restart the Telegraf container by using Cockpit or Portainer or restart the whole controller.

## 3.6 PACEdge System Level Settings

PACEdge 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 PACEdge. This will bring you to the PACEdge 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: 3.5.2 Changing Passwords Manually (Portainer, Cockpit)
- Adding /Removing users accounts
- Changing Hostname  
Refer to: Section *Changing Host Name*
- Modifying IP addresses, VLANs, enabling Firewall
- Managing Storage
- Managing Docker Containers
- Monitoring CPU, Memory, Network, Storage usage, and performance
- Analyzing system logs
- Monitoring system services and performing system updates
- Shortcuts to other PACEdge applications
- Linux Terminal (for users that desire further customization)

#### Changing Host Name

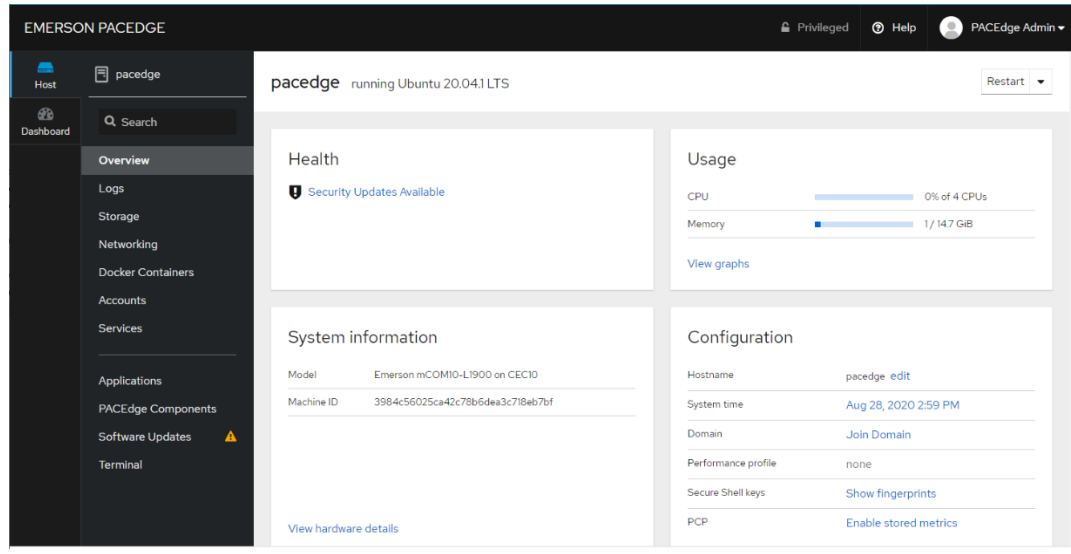
PACEdge supports mDNS/DNS-SD protocols, which allows devices to be discovered on the network by using Hostname. By default, PACEdge systems are delivered with Hostname set to **pacedge**. In the networks that do support mDNS/DNS-SD users can, for instance, ping PACEdge devices by using **pacedge** instead of the IP address. If a user has more than one PACEdge device in the same network, Hostname conflicts will arise and therefore the user should consider changing the Hostname of the PACEdge to a unique value. To change the Hostname:

1. From the PACEdge 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 in the Overview tab, in the upper-right corner select the desired action. Note that when selecting an action there is also an option to select time delay.

Figure 22: Cockpit Screen



For further Cockpit details, 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

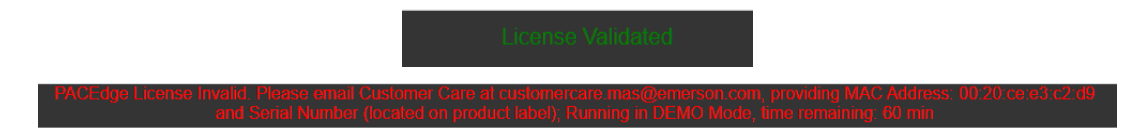
## 3.7 PACEdge License File

PACEdge is protected with the license file, which is locked to the physical device that PACEdge is running on. PACEdge 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 *PACEdge 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 PACEdge 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):

**Figure 23: License Validation**



2. If the message indicates **License Validated** then nothing is to be done; otherwise, copy the MAC Address shown in the message (for instance 00:20:ce:e3:c2:d9), get a serial number of the PACEdge device from the label on the unit, and 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 PACEdge landing page, Settings, Upload License Files option, and upload both files:

Figure 24: Upload License Files Menu

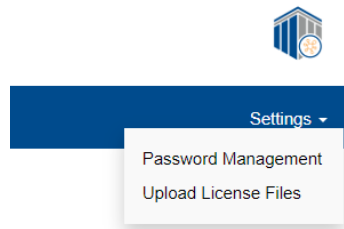


Figure 25: Browse for License Files

### Upload License Files

Select 'license.json'

No file chosen

Select 'license.sig'

No file chosen

**Note:** Only license.json and license.sig are valid filenames (max 1KB).

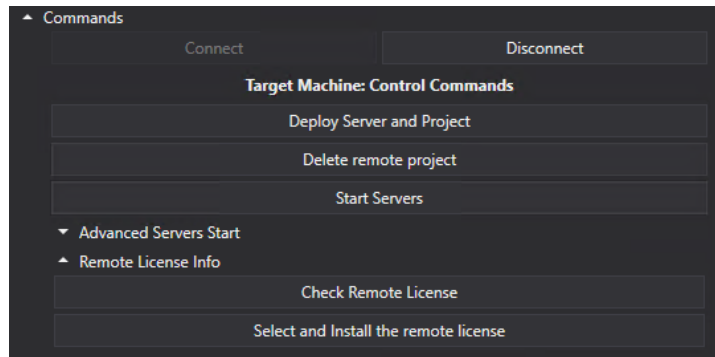
4. Once the files have been uploaded, perform a graceful reboot of the PACEdge device via Cockpit. Information on how to gracefully reboot the system can be found in Section *Gracefully Restarting, Shutting Down the* .
5. After reboot, wait a few seconds for all services to fully come up and refresh the browser. If successful, the **License Validated** message seen in step 1 should be visible.

## 3.7.1 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 unit running PACEdge from your workstation where Movicon.NExT software is installed and do following:

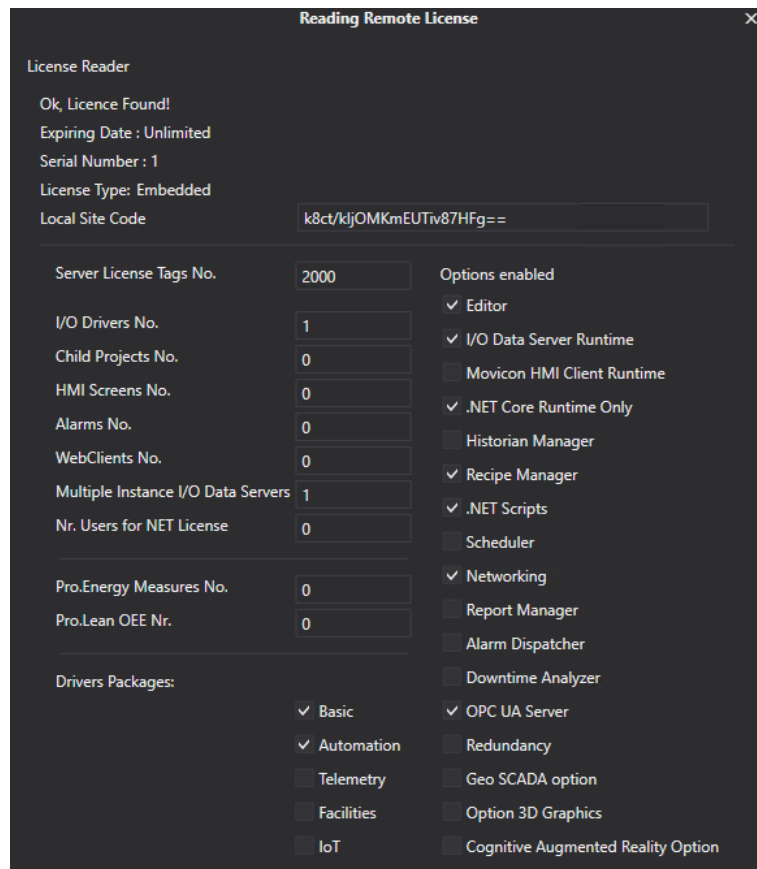
1. Follow steps described in: **Movicon.NExT to Connex/WebHMI on IPC** to establish connection to PACEdge using Deploy Project window
2. Go to **Remote License Info** section and click on **Check Remote License:**

Figure 26: Connex and WebHMI License Check Dialogue



3. A picture below shows the enabled features for the Connex SKU:

Figure 27: License Enabled Features in Connex SKU

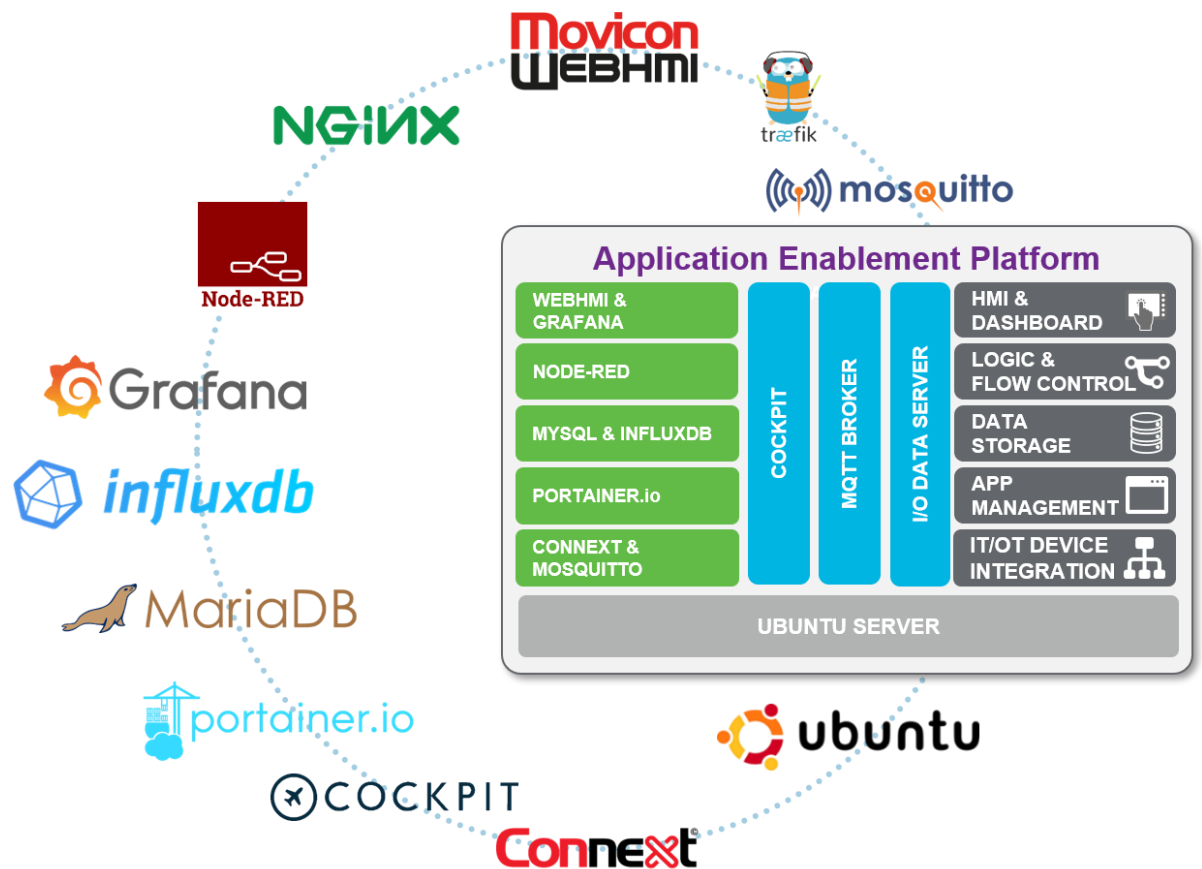


# Section 4: PACEdge Architecture Details

## 4.1 PACEdge Software Services

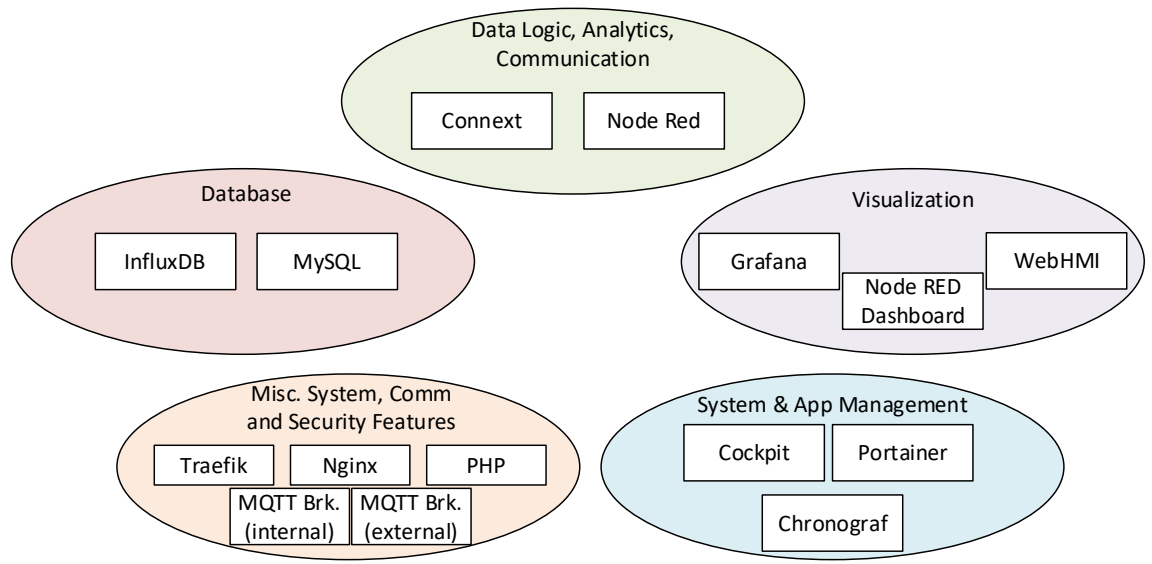
PACEdge entails several communications, data processing, data storage, and visualization applications that run on the Linux operating system. The following diagram gives an overview of components that make up PACEdge:

Figure 28: Software Overview



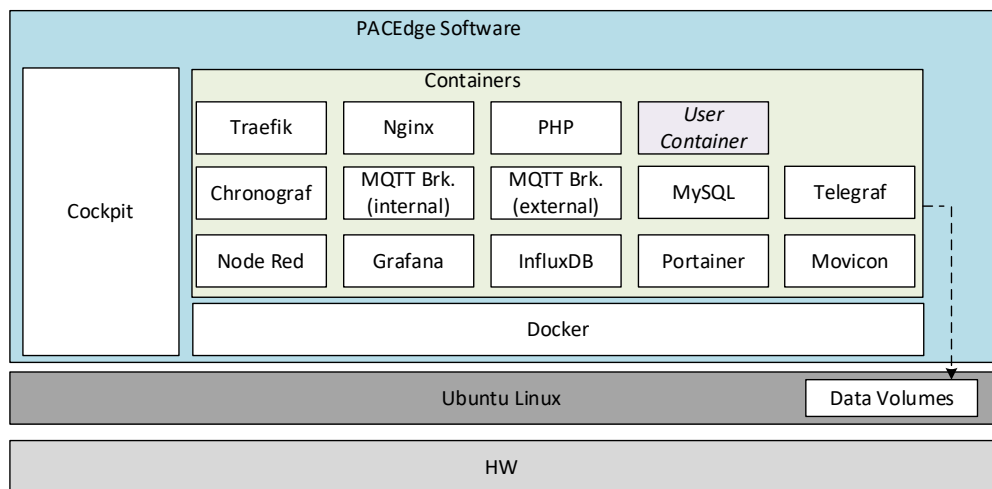
PACEdge tools and applications, based on the function that they perform, can be grouped into functional categories as follows:

Figure 29: PACEdge Application Categories



PACEdge was designed using Docker architecture, in which each application is running in its own Docker container (Figure 30). Given that containers are designed to be easily replaceable they do not retain an internal state between reboots unless specifically designed to do so. With PACEdge, 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 30: Containerized PACEdge Implementation



## 4.1.1 Application Development Components

### Node-RED

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 a large number of freely available nodes that can be easily installed. PACEdge comes with a large selection of pre-installed nodes, that allows users to easily send and receive data via MQTT, OPC-UA, ModbusTCP, ModbusRTU interfaces, process data, visualize data via the dashboard, store data in InfluxDB or MySQL databases and send alerts via emails. 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](http://www.nodered.org)

### Connex and WebHMI

Connex 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 is a visualization tool that enables users to view and analyze data as well as to 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](http://www.grafana.com)

### MQTT

PACEdge implements two MQTT brokers, one is meant to be used for PACEdge 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](http://www.mqtt.org)

### Traefik

Since most of the PACEdge 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 PACEdge IP address, as opposed to 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/](http://docs.traefik.io/)

## Nginx and PHP

NGINX and PHP are two services that are 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

PACEdge 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](https://www.influxdata.com/products/integrations/?_integrations_dropdown=telegraf-plugins)

### 4.1.2 Database Components

#### MariaDB (MySQL)

MariaDB (MySQL) is a relational database. Node-RED has a pre-installed node to access this database. I/O Data Server in Connex 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/](http://mariadb.org/)

#### InfluxDB

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](http://www.influxdata.com)

### 4.1.3 Administration Components

#### Cockpit Description

PACEdge 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 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 as well as 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_PACEdge:9090/cockpites/`

For more details about Cockpit please refer to [www.cockpit-project.org](http://www.cockpit-project.org)

## Portainer

PACEdge is heavily utilizing Docker's container-based implementation, which allows the user 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, stop containers, check the container log file, configure restart policies, open ports, etc.

For more details about Portainer please refer to [www.portainer.io/](http://www.portainer.io/)

## Chronograf

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_PACEdge/chronograf/`

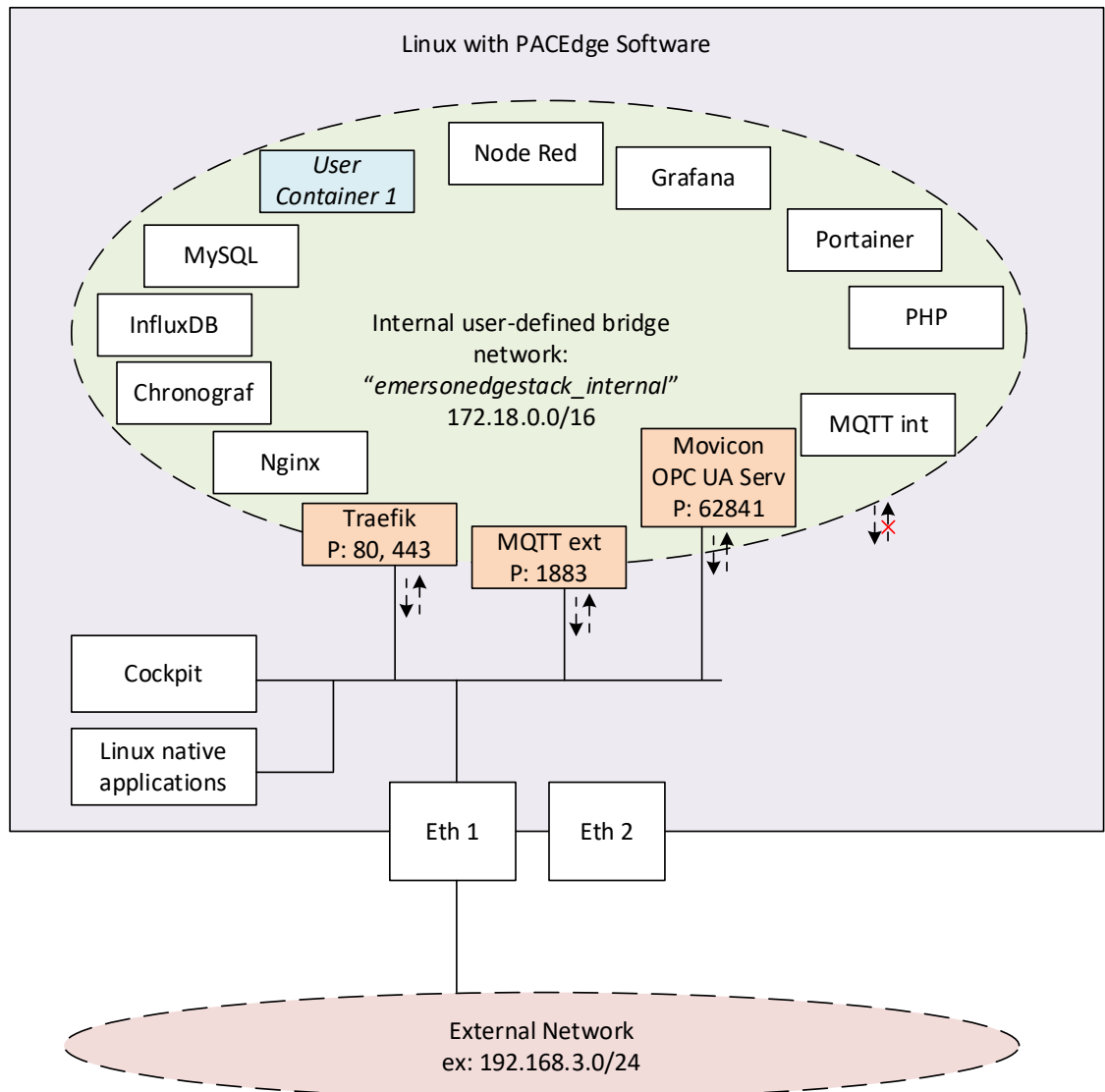
For more details about Chronograf please refer to [www.influxdata.com/time-series-platform/chronograf/](http://www.influxdata.com/time-series-platform/chronograf/)

## 4.2 PACEdge Communications and Data Flow

Considering that most of the services in PACEdge are implemented using Docker Containers main communication between them is implemented using network interfaces. For security and traffic segregation reasons PACEdge is using an internal user-defined bridge network, with IP subdomain address 172.18.0.0/16. Each of the Docker containers 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 do have open ports (80, 443, 1883, 62841) which enable these services to be reached from the outside.

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

Figure 31: PACEdge Network Communication Paths



## 4.2.1 PACEdge Internal Communication Parameters

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

### InfluxDB Access Parameters

- Host: influxdb
- Port: 8086
- Usernames: admin, developer
- Passwords: as changed by the user

### Grafana communication parameters

Grafana is configured with the following users:

- admin: Full rights, can setup users and their passwords, add databases
- developer: can create a dashboard, but can not setup databased
- operators: read-only access

To configure databases inside Grafana, login as an admin user.

To setup InfluxDB database:

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

To setup 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
- Username: admin or developer
- Password: as changed by the user

### Internal MQTT communication parameters

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

## 4.2.2 PACEdge External Communication Parameters

### PACEdge Landing page

To access PACEdge landing page via browser:

- [https://ip\\_address\\_of\\_PACEdge](https://ip_address_of_PACEdge)

### Node-RED

To access Node-RED via browser:

- [https:// ip\\_address\\_of\\_PACEdge /nodered/](https://ip_address_of_PACEdge/nodered/)
- shortcut on the landing page

### Grafana

To access Grafana via browser:

- [https:// ip\\_address\\_of\\_PACEdge /grafana/](https://ip_address_of_PACEdge/grafana/)
- shortcut on the landing page

### 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

### Portainer

To access Portainer via browser:

- [https:// ip\\_address\\_of\\_PACEdge /portainer/](https://ip_address_of_PACEdge/portainer/)
- shortcut on the landing page

### WebHMI

To access WebHMI via browser:

- [https:// ip\\_address\\_of\\_PACEdge /progea/](https://ip_address_of_PACEdge/progea/)
- shortcut on the landing page

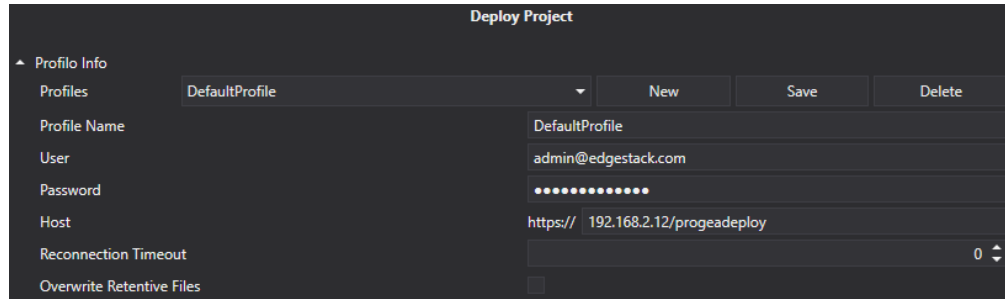
### Movicon.NExT to ConnexT/WebHMI on IPC

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

- User: [admin@edgestack.com](mailto:admin@edgestack.com)

- Password: Edgestack123!
- Host: xx.xx.xx.xx/progeadeploy  
(where xx.xx.xx.xx is an IP address of the remote industrial PC, such as 192.168.3.100)

Figure 32: Profilo Info



Note: User name and password used need to match what is specified in appsettings.json file, as described in section 3.5.3 Changing Connex Password

## External MQTT Communication

PACEdge 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 PACEdge unit:
  - Server: MQTT
  - Port: 1883
- When accessing MQTT broker in the other PACEdge 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

Note that by default MQTT security (username, password, certificates, encryption) is disabled, but a user can choose to enable it via MQTT Broker configuration settings. For details please refer to: [mosquitto.org](https://mosquitto.org)

### 4.2.3 Accessing Connex OPC UA Server

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

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

`opc.tcp://192.168.3.100:62841`

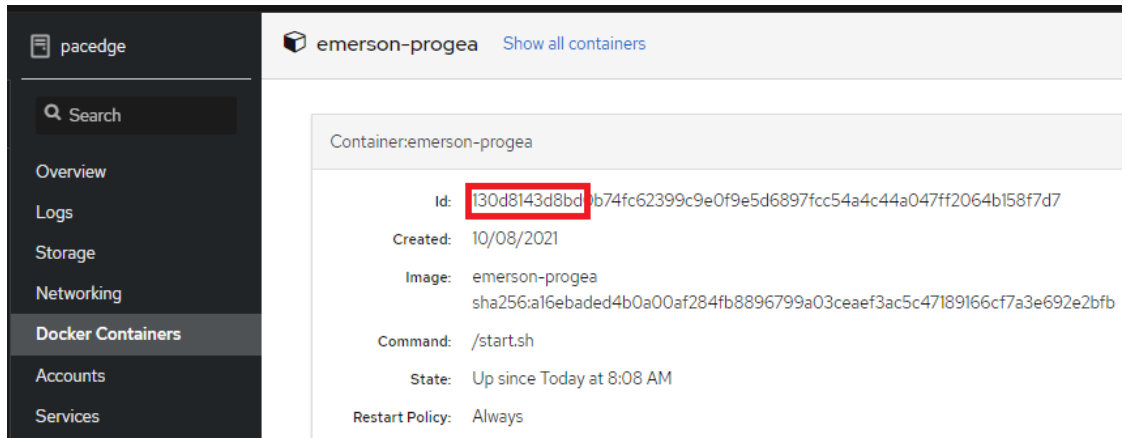
## Accessing Connex OPC UA Server from Movicon.NExT Browser

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

To find out host name of Connex docker container, please follow the steps below:

1. Open Cockpit, go to Docker Containers tab.
2. Find container named **emerson-progea** and click on the name to open details.
3. The container Hostname is **first 12 characters of container Id**:

**Figure 33: Finding the Hostname with Cockpit**



**Note:** An alternative method to get the container Hostname is open Cockpit through Terminal. Tap and issue the following command. Hostname is set of characters following **root@**.

**Figure 34: Finding the Hostname with Terminal**

```
admin@pacedge:~$ docker exec -it emerson-progea /bin/bash
root@130d8143d8bd:/#
```

4. If Movicon.NExT is running on a Windows computer, the Hostname has to be associated with PACEdge device IP address. This association is established in the hosts file:  
C:\Windows\System32\drivers\etc\hosts  
This file is protected and cannot be edited directly, therefore use following procedure:
  - a. Copy hosts file to Desktop folder.
  - b. Open hosts file with Notepad or other editor.
  - c. Add a line with PACEdge IP address, space, Hostname of container, such as:

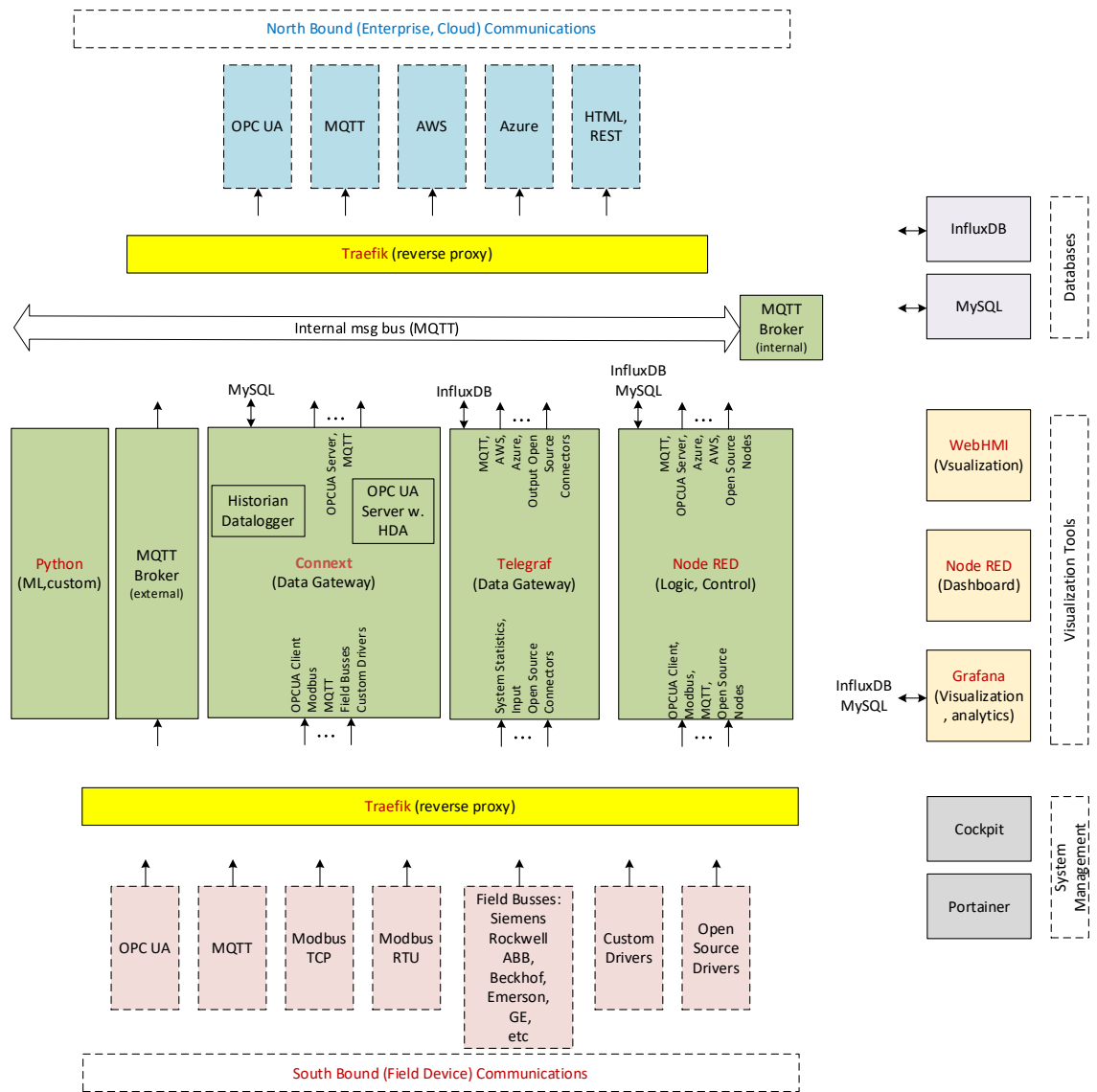
```
192.168.3.100 130d8143d8bd
```

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

## 4.2.4 PACEdge Suggested Data Flows

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

**Figure 35: PACEdge Communication Capabilities**



## 4.2.5 Southbound Communication Capabilities

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

### Connex I/O Drivers

Connex data gateway component implements 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, Beckhof, GE, Hilsher, Mitsubishi, Omron, Phoenix and many more. For the more 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 efficiently add new and custom drivers supporting efficient customization. Data that is being received by the driver 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 OPC UA Server.

### Node-RED Nodes

Node-RED has a large open-source community developing and constantly adding new communication nodes. Node-RED has good support for OPC UA client and server, 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

PACEdge 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](https://www.influxdata.com/products/integrations/?_integrations_dropdown=telegraf-plugins)

### Custom Driver

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

## 4.2.6 North Bound Communication Capabilities

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

### OPC UA Server

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

Connex comes with an integrated high-performance OPC UA Server and is a recommended option. All data tags within the Connex 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 includes OPC UA server.

## MQTT

PACEdge comes with MQTT Broker, which can be accessed from outside of IPC. MQTT Broker used is based on Mosquitto open-source implementation and does support Sparkplug B specification, as well as security and encryption features. For access details please refer to Section *External MQTT Communication*.

## Cloud Connectivity

Cloud connectivity can be established by 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

### 4.2.7

## PACEdge Internal Communications and Data Flow

PACEdge can be seen as a Swiss army knife when it comes to 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

PACEdge 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 PACEdge. MQTT is fast, simple to set up, and has low overhead, saving CPU resources. Most agents and applications within PACEdge, such as Connex, Node-RED, Telegraf, Python readily support MQTT communication. As an 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 be publishing MQTT data and Connex can receive it via the MQTT driver. Telegraf can publish received data, such as CPU utilization statistics, via MQTT as well.

## OPC UA – Alternative for Internal Communication

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

## Sharing data via Databases

For not-so-real-time-critical data sharing a good way to consider is storing data in the database and letting other applications extract data directly from it. As an example, Connex has a Historian and Datalogger functionality, which can be configured to store 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.

# Section 5: PACEdge 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 shows them via Node-RED Dashboard

### 5.1.2 Prerequisites

None

### 5.1.3 Example Walkthrough

#### Getting Started with Node-RED

1. Open 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 36: Inject Node and Debug Node




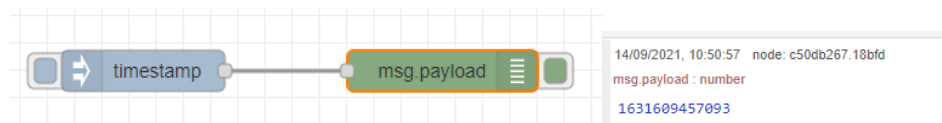
3. Connect **inject** node to **debug** node by dragging the line between grey connection points.
4. Open the debug window by clicking on the bug symbol  in the upper-right corner.
5. Click on the red **deploy** button in the upper-right corner
6. At this point, the flow has been deployed and is active. By 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 37: 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, instead of sending the time stamp, you can send a string or JSON structure. For our example, let us configure a repeat mode, by selecting **interval** and setting it to be 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. Now the message the **inject** node sends will first go to function node and then to **debug** node.

Figure 38: Place the Function Node Between the Timestamp and Debug Nodes



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

Figure 39: JavaScript Code

```
Setup | On Start | On Message | On Stop
1 var a = msg.payload;
2 msg.payload = Math.sin(a/1000/10);
3 return msg;
```

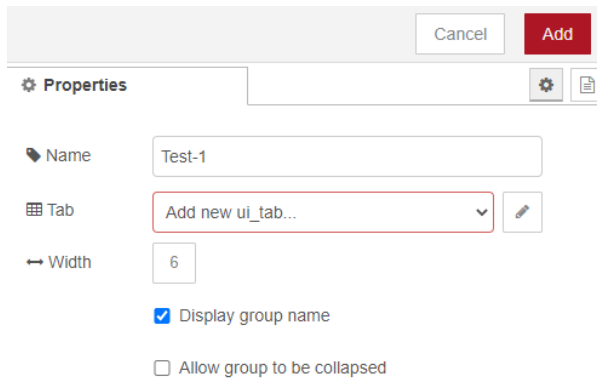
This code will first assign the received timestamp to **variable a**, then scale it down a bit and calculate the Sine value of it. The result will be sent out to the next node.

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

## Adding Node-RED Dashboard

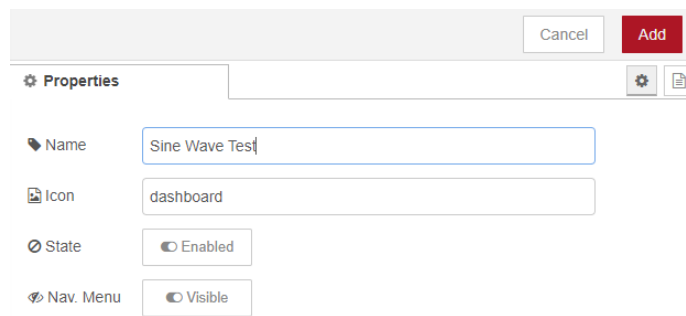
1. Drag and drop a **chart** node and connect it to the function node.
2. Double-click the chart node and make the following changes:
  - a. In the Group, select **Add new ui\_group** and click on the pen symbol to edit.
  - b. In the Properties Window, provide a name (such as “Test-1”) in the Name field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

Figure 40: Properties



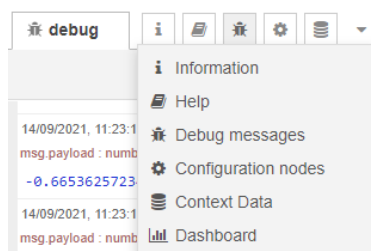
- c. In the new dialogue, give a name, such as **Sine Wave Test** and click on the red **Add** button.

Figure 41: Sine Wave Test



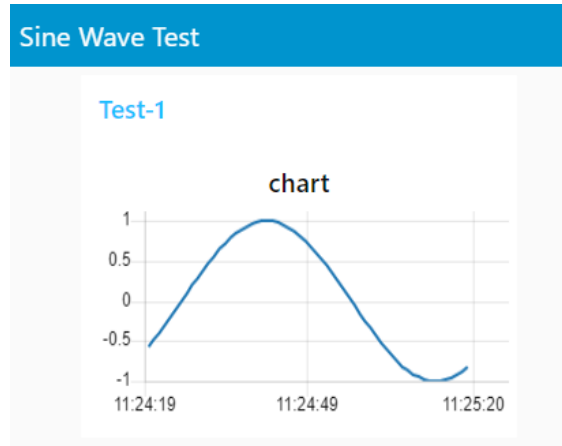
- d. Click on **Add** button again
3. Back in the Edit chart dialogue, in X-axis, select **last 1 minute**, so that the data for the last 1 minute is visible.
4. Give the chart a name and click **Done**.
5. Click on **Deploy** to save the changes.
6. Open the Dashboard window by clicking on the drop-down symbol in the upper-right corner and select **Dashboard**.

Figure 42: Dashboard Menu



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

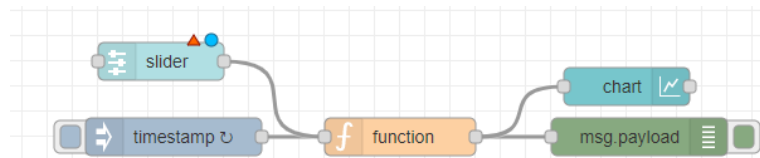
Figure 43: Sine Wave Test



## Adding Control to the Node-RED Dashboard

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

Figure 44: 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 that two different messages are coming into the **function** node. We need to change the code to check for the message topic and take an action accordingly. Change the code inside the function node as follows:

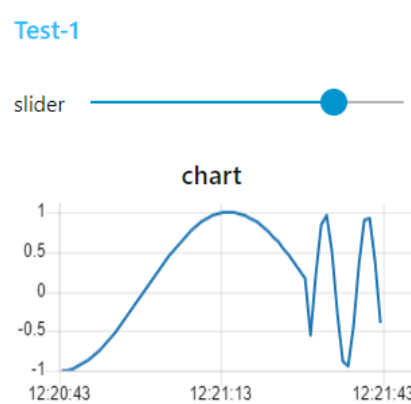
Figure 45: Function Node's Code Change

```

  Setup | On Start | On Message | On Stop
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;
```

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 46: Changed Sinus Value



## 5.2 Using the InfluxDB Database

### 5.2.1 Example Description

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

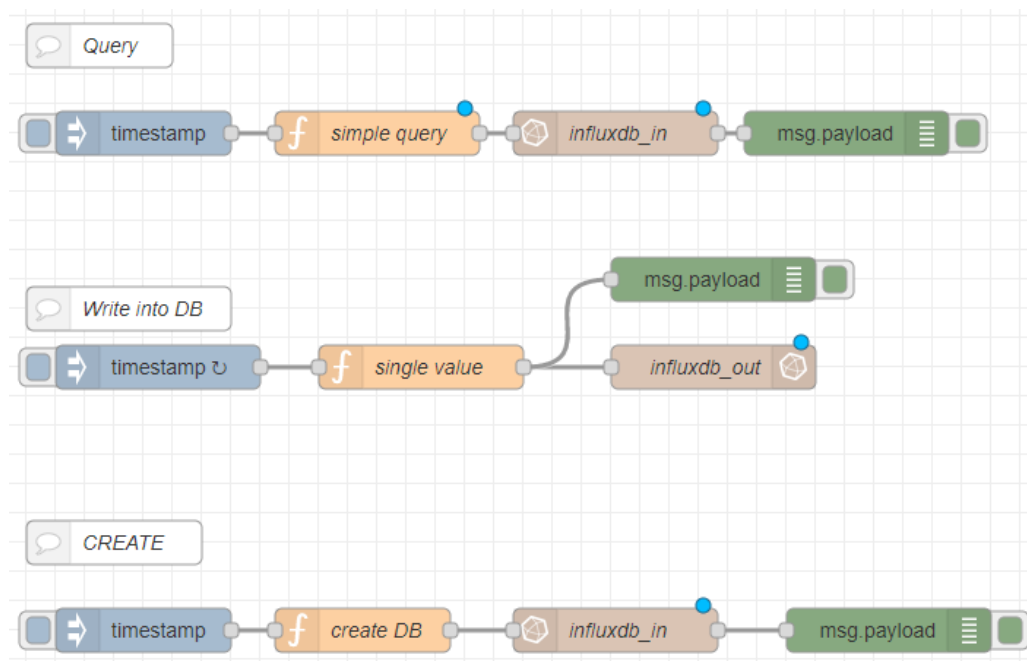
### 5.2.2 Prerequisites

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

### 5.2.3 Example Walkthrough

In the Node-RED flow, window place **inject**, **function**, **influxdb\_in**, **influxdb\_out**, and **debug** nodes and interconnect them as shown. Note, for example, is also using comment nodes to make it more readable.

Figure 47: Example Walkthrough



1. In the Query section, double-click on the **function** node and enter the following code. It will be used to query the database:

**Figure 48: Query Section: Function Node's Code**



```
1 msg.query="select * from demo_1;";
2 return msg;
```

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

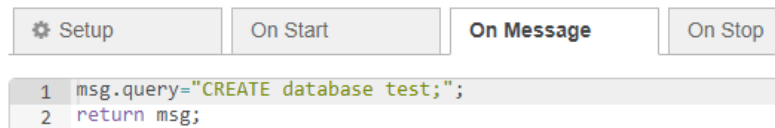
**Figure 49: DB Section: Function Node's Code**



```
1 msg.payload = Math.random()*10;
2 return msg;
```

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

**Figure 50: Create Section: Function Node's Code**



```
1 msg.query="CREATE database test;";
2 return msg;
```

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

Figure 51: 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, which will write data into the database we need to also provide the name for the Measurement. Please enter the desired name, such as `demo_1`. Click **Done** afterward.

Figure 52: 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, then you can write to and read from the database.
9. Note that the database is referenced by name, which is in our flow tests. This name is entered in the InfluxDB nodes Database field, but also needs to match in function node create DB.

Therefore, if you would like to change the database name, you need to change it in the create database flow function node as well.

10. If you want to modify query parameters, edit the function node.

## 5.3 Using MySQL Database

### 5.3.1 Example Description

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

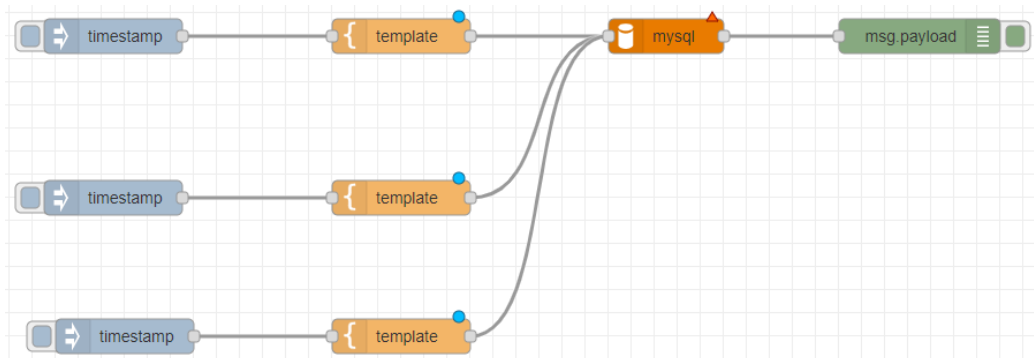
### 5.3.2 Prerequisites

If new to Node-RED, completion of the 5.1 *Getting Started with Node-RED* highly recommended, as it shows some Node-RED usage basics and tips.

### 5.3.3 Example Walkthrough

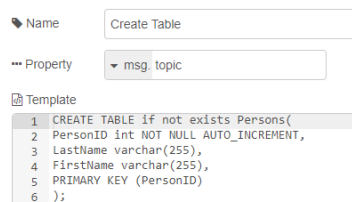
In this example, we will use a template note, instead of a function node. This is just to show that there are alternative ways to implement the same functionality. Drag and drop inject, template, MySQL, and debug nodes and interconnect them as shown. Note that in contrast to InfluxDB nodes, for MySQL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

Figure 53: Walkthrough Example



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

Figure 54: Template Node



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

**Figure 55: Write to Database Node**

Name: Write to Database

Property: msg.topic

Template: `1 INSERT into Persons (`LastName`,`FirstName`) VALUES ('Smith', 'John');`

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

**Figure 56: Write to Database Node**

Name: Query Database

Property: msg.topic

Template: `1 SELECT * FROM Persons;`

4. Double-click MySQL node in the Database and 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 that you choose and Database field value data. Click on **Update** and then on **Done**.

**Figure 57: My SQLDatabase Node**

Edit mysql node > Edit MySQLdatabase node

Delete Cancel Update

Properties

Host: mysql

Port: 3306

User: admin

Password: .....

Database: data

Timezone:

Charset: UTF8

Name: Name

6. Once all the edits are done, click on the **Deploy** button. If the MySQL node got properly configured you will see a green dot appear at the bottom of it with the label OK.
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

### 5.4.1 Example Description

This example shows how to use internal and external MQTT brokers and communication.

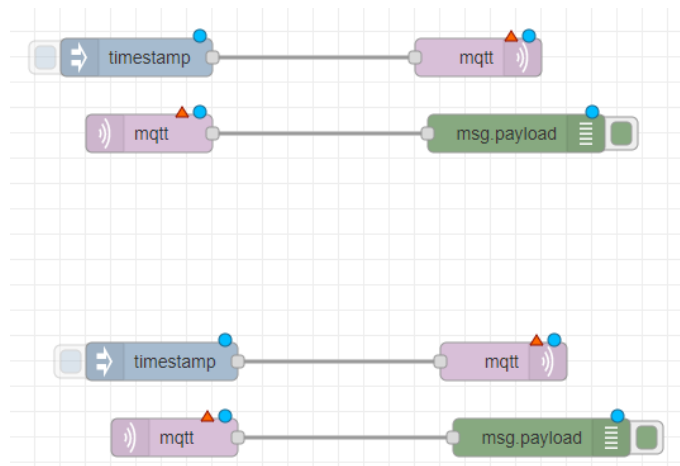
### 5.4.2 Prerequisites

If new to Node-RED, completion of the 5.1, *Getting Started with Node-RED* example is highly recommended, as it shows some Node-RED usage basics and tips.

### 5.4.3 Example Walk Through

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

Figure 58: 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 Sever field enter MQTT-internal-IPC, then click on **Add**.

The screenshot shows the 'Properties' dialog box for an MQTT broker. The 'Name' field contains 'Internal\_MQTT'. Below it are tabs for 'Connection', 'Security', and 'Messages'. Under the 'Connection' tab, the 'Server' field is 'mqtt-internal-ipc' and the 'Port' is '1883'. There is an unchecked checkbox for 'Use TLS'. The 'Protocol' dropdown is set to 'MQTT V3.1.1'. The 'Client ID' field contains the text 'Leave blank for auto generated'. The 'Keep Alive' field is '60'. At the bottom, the 'Session' section has a checked checkbox for 'Use clean session'. 'Cancel' and 'Add' buttons are at the top right.

4. Enter MQTT Topic name, such as **Internal\_Test**, and click **Done** to close the window.
5. Then, double-click on the first **MQTT input** node and select the same Server that was created above. Also, enter the same Topic as above, 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 Sever field enter MQTT, then click on **Add**.

**Figure 59: External MQTT Properties**

The screenshot shows the 'Properties' dialog box for an external MQTT broker. The 'Name' field contains 'External\_MQTT'. Below it are tabs for 'Connection', 'Security', and 'Messages'. Under the 'Connection' tab, the 'Server' field is 'mqtt' and the 'Port' is '1883'. There is an unchecked checkbox for 'Use TLS'. The 'Protocol' dropdown is set to 'MQTT V3.1.1'. The 'Client ID' field contains the text 'Leave blank for auto generated'. The 'Keep Alive' field is '60'. At the bottom, the 'Session' section has a checked checkbox for 'Use clean session'.

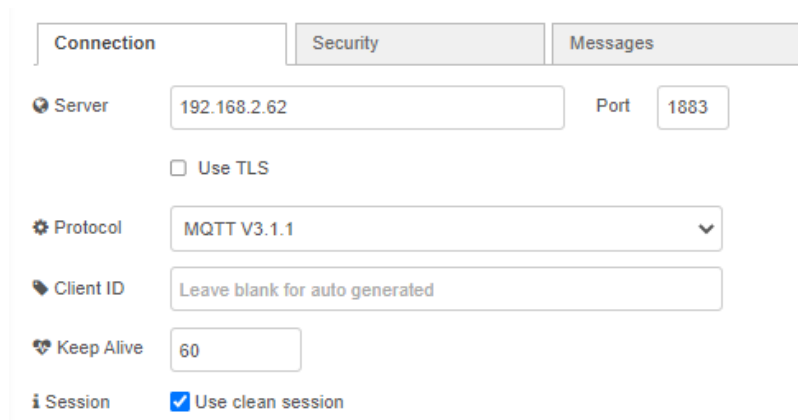
8. Enter MQTT Topic name, such as **External\_Test**, and click **Done** to close the window.
9. Double-click on the second **MQTT input** node and select the same external Server that was created above. Also, enter the same Topic as above, click **Done**.

10. After all the edits are done, click on the **Deploy** button
11. When triggering each flow, by clicking on the **inject** node you will see that a time stamp gets generated, published to the respective MQTT Broker, and specified 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 PACEdge system, then you can connect it to this setup by configuring the MQTT broker as follows, where 192.168.2.62 is an IP address of the first PACEdge system.

---

**Figure 60: Connection Settings**



The screenshot shows the 'Connection' settings for an MQTT broker. The interface includes three tabs: 'Connection', 'Security', and 'Messages'. The 'Connection' tab is selected. The settings are as follows:

- Server:** 192.168.2.62
- Port:** 1883
- Use TLS:**
- Protocol:** MQTT V3.1.1
- Client ID:** Leave blank for auto generated
- Keep Alive:** 60
- Session:**  Use clean session

## 5.5 Using Telegraf and Grafana

### 5.5.1 Example Description

Grafana is a graphical application that enables users to quickly visualize the data and perform different analytics. Grafana supports a large number of databases, including InfluxDB and MySQL, which come pre-installed with PACEdge. This example will walk through configuring Grafana to use both InfluxDB and MySQL databases and how to setup dashboards to visualize system performance

### 5.5.2 Prerequisites

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

### 5.5.3 Example Walk Through

#### Configuring Grafana with InfluxDB database

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

**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 into Grafana.

Figure 61: Grafana Configuration Screen for InfluxDB

**Data Sources / InfluxDB**  
Type: InfluxDB

Settings

Name: telegraf\_metrics Default

Query Language: InfluxQL

HTTP

URL: http://influxdb:8086

Access: Server (default) Help >

Whitelisted Cookies: New tag (enter key to add) Add

Auth

Basic auth  With Credentials

TLS Client Auth  With CA Cert

Skip TLS Verify

Forward OAuth Identity

Custom HTTP Headers

+ Add header

InfluxDB Details

Database Access

Setting the database for this datasource does not deny access to other databases. The InfluxDB query syntax allows switching the database in the query. For example: `SHOW MEASUREMENTS ON _internal` or `SELECT * FROM "_internal"."database" LIMIT 10`

To support data isolation and security, make sure appropriate permissions are configured in InfluxDB.

Database: telegraf\_metrics

User: admin

Password: .....

HTTP Method: GET

Min time interval: 10s

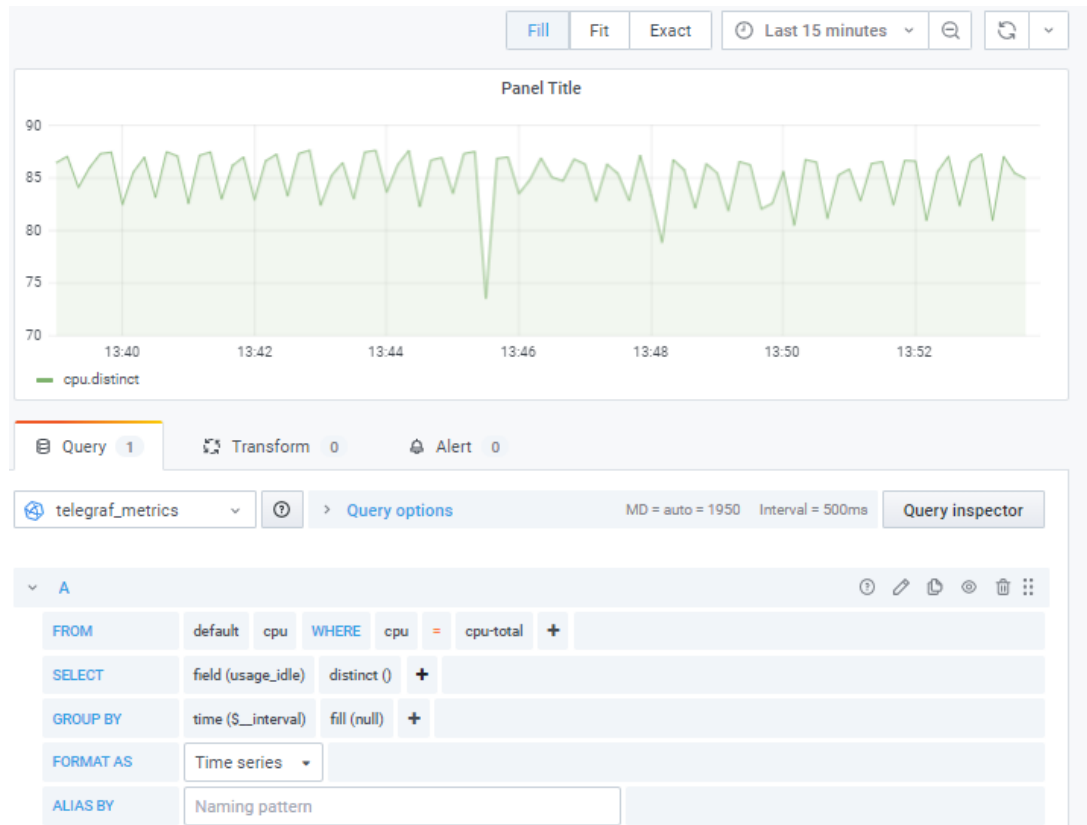
Max series: 1000

Save & Test Delete Back

1. To create a new dashboard in Grafana, hover your mouse over Create symbol and click on **Dashboard**, then select the **Add an Empty Panel** option.
2. Select your configured InfluxDB database, telegraf\_metrics in this case.

3. In line item FROM from the drop-down list pick **CPU**, then in WHERE pick **CPU** and then **CPU-total** on the other side of the equal sign
4. In SELECT line item pick field value **usage\_idle** and then change the default **mean** to **distinct**
5. At this point, you will see the graph and can further manipulate the views and save it.

Figure 62: Telegraf Metrics



**Note:** By default there is a relatively large number of system parameters being stored and is available to view, things like memory, disc, interrupts swap, etc. What is being stored in InfluxDB you can change by editing the Telegraf configuration file.

6. Alternatively, you can download an already prepared Grafana dashboard from the community. To do so go to Dashboards, then Manage. Click on **Import** and in the Import via Grafana.com field type in: **928** and click on **Load**. In the dialogue that opens up point to the database, Telegraf is using: telegraf\_metrics and click Import. The resulting Grafana dashboard will look as follows:

Figure 63: Grafana System Dashboard



## Configuring Grafana with MySQL database

1. From the PACEdge landing page click on the **Grafana** link and make sure to log in as admin.
2. On the left side, move the cursor over the Configuration icon 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 into Grafana. The following steps assume that the database Movicon has been created in MySQL.

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

Figure 64: MySQL Connection

Name  Default

### MySQL Connection

Host

Database

User  Password

TLS Client Auth  With CA Cert

Skip TLS Verify

### Connection limits

Max open

Max idle

Max lifetime

### MySQL details

Min time interval

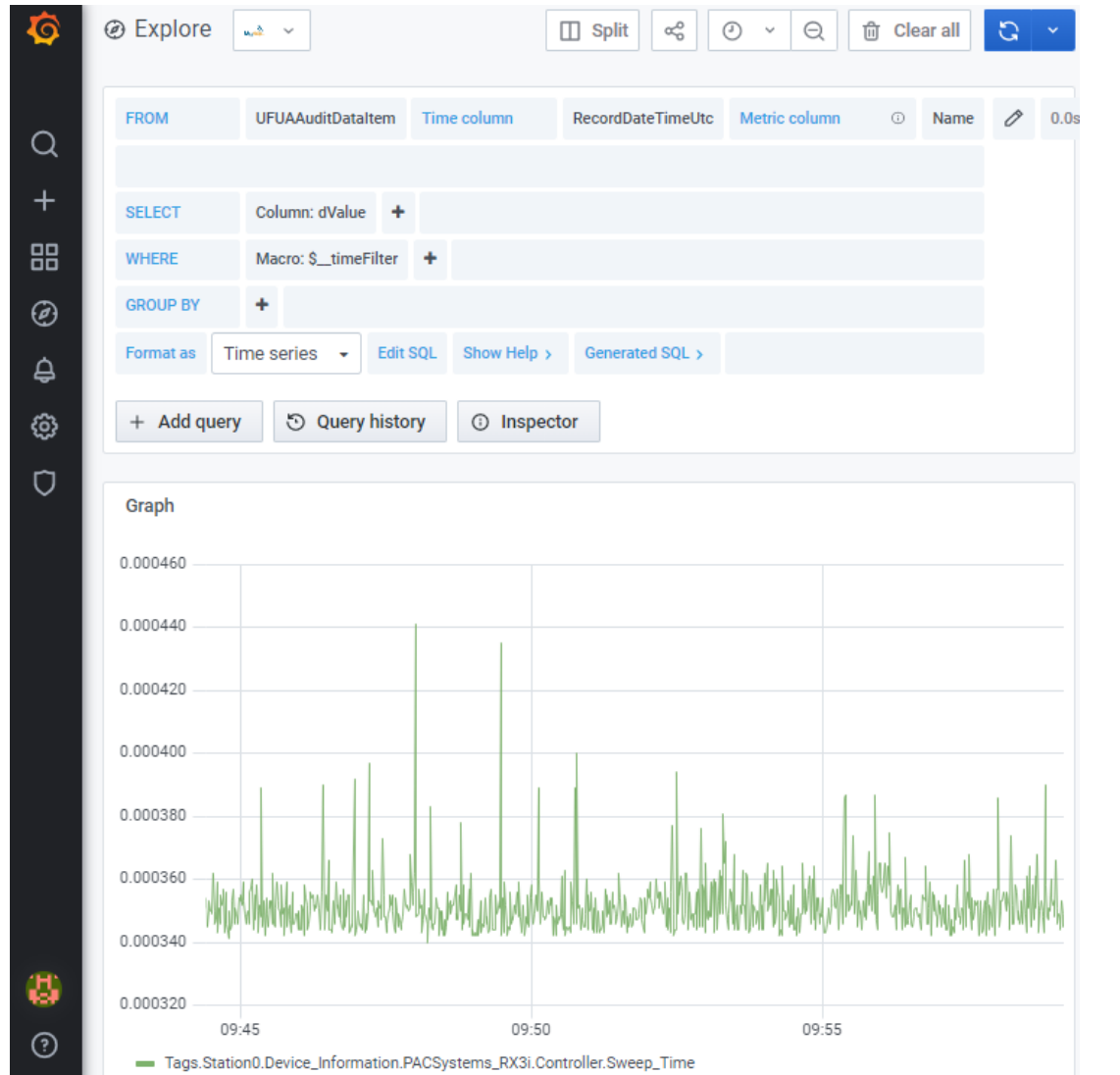
#### User Permission

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

5. On the left side, bar click either on Explore icon or hover your mouse over Create symbol and click on Dashboard, then select Add an Empty Panel option

6. Make sure the newly configured **MySQL database** is selected, then in the Metric column select Name, and the **Select** field click on the **Column: OID** and select **dValue**. You should see the historical values of the Sweep Time data tag.

Figure 65: Sweep Time Data



## 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, historize this data and visualize it via both WebHMI as well as via Grafana.

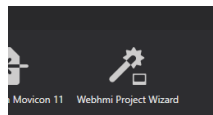
### 5.6.1 Prerequisites

Basic knowledge of Movicon.NEXt is required.

### 5.6.2 Example Walk Through

Movicon Connex and WebHMI functionality first need to be created on the user workstation using Movicon.NEXt software, typically running under Windows. Once required data connectivity, internal data tags, historian settings as well as 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.

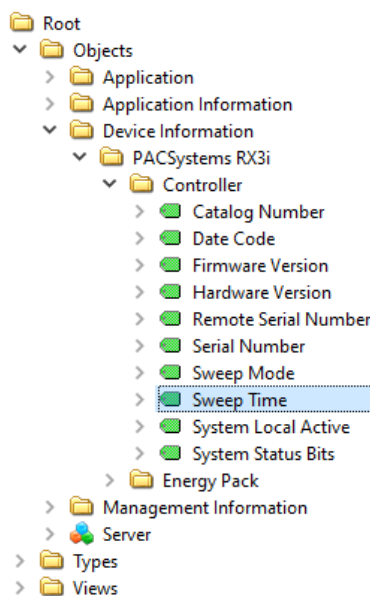
**Figure 66: WebHMI Project**



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

As an example, let us assume we have a CPL410 and want to use OPC UA to get some 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:

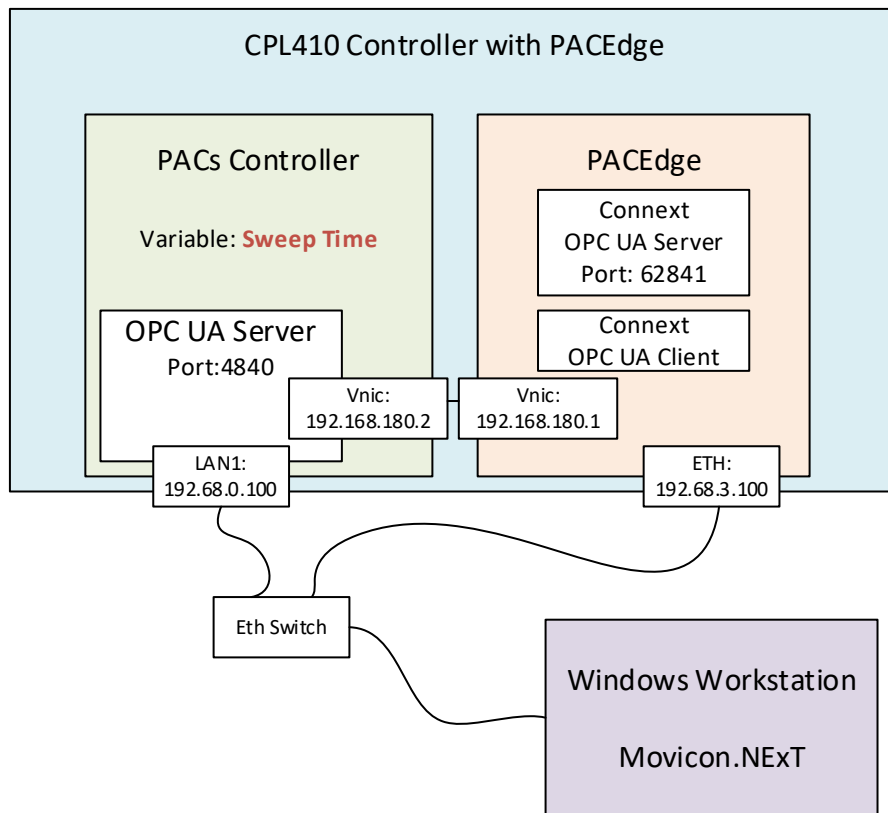
**Figure 67: UA Expert view of CPL410 data tags**



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

CPL410 is a hybrid device, containing both PACs automation controller as well as 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) as well as an external Ethernet LAN1 port.

**Figure 68: CPL410 Internal Block Diagram**



There are different ways to configure OPC UA Client, but for this example, we will use the following strategy:

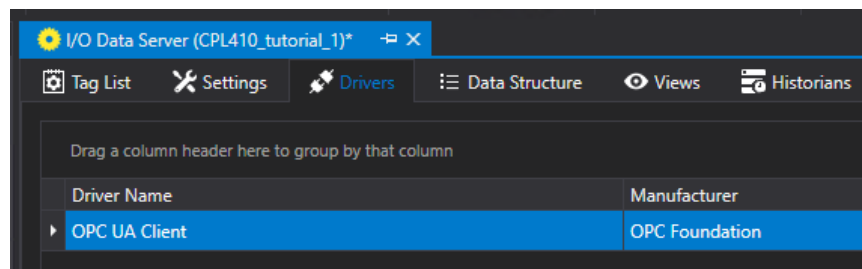
- Use Movicon.NExT running on the workstation to browse through OPC UA data tags on the PACs OPC UA Server, by using the LAN1 ethernet port.
- Configure WebHMI project
- Test WebHMI project locally on the workstation
- Transfer WebHMI project to target CPL410
- Test WebHMI project running on CPL410

## Configuring I/O Data Server

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

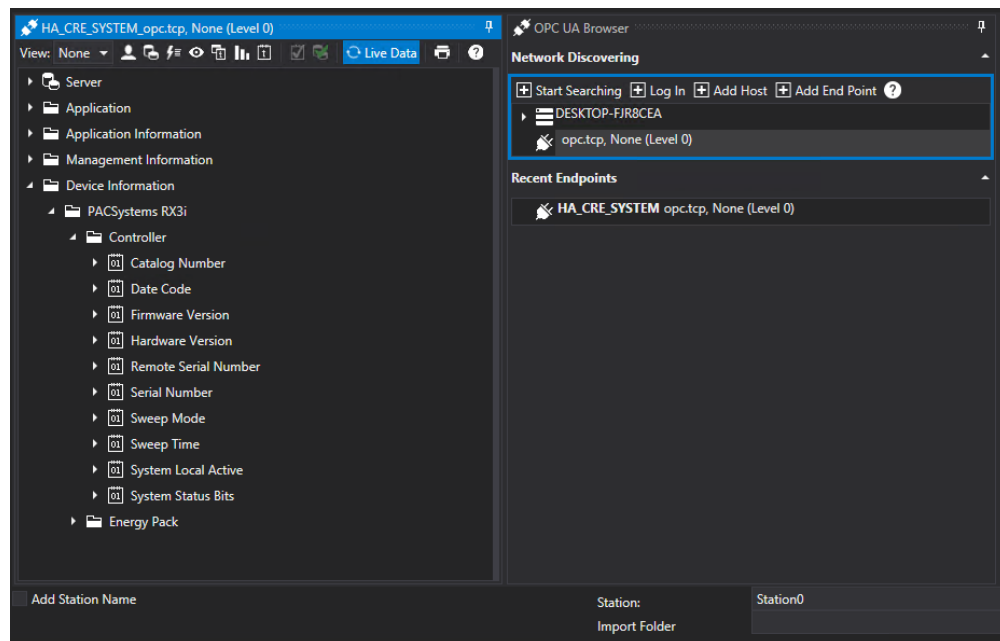
1. In Movicon.NExT use wizard to create new Webhmi project
2. In Movicon.NExT, in the I/O Data Server section, double-click on **Drivers**.
3. In the main window right-click **select Add New Driver** and select **OPC UA Client** from the list, click on the > button.
4. Keep the default settings in step 3 and click again on the > button. In the Host Name enter: 192.168.0.100 (only IP address, no port number, no opc. TCP prefix) and click > button
5. Keep the default settings in step 4 and click again on >, click on **Finish**. You will see an OPC UA Client driver listed in the driver window.

Figure 69: OPC UA Client



6. Right-click on the OPC UA Client entry and select **Import Tags**.
7. Select **Add End Point: opc.tcp://192.168.0.100:4840**, double-click on the newly created server entry to connect to OPC UA Server, and browse for available tags.

Figure 70: Browse for Available Tags



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

## Configuring WebHMI Screen

1. In Movicon.NExT, right-click on **Screens** and select **New** to configure the HMI Screen.
2. Choose the size of the screen, 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.

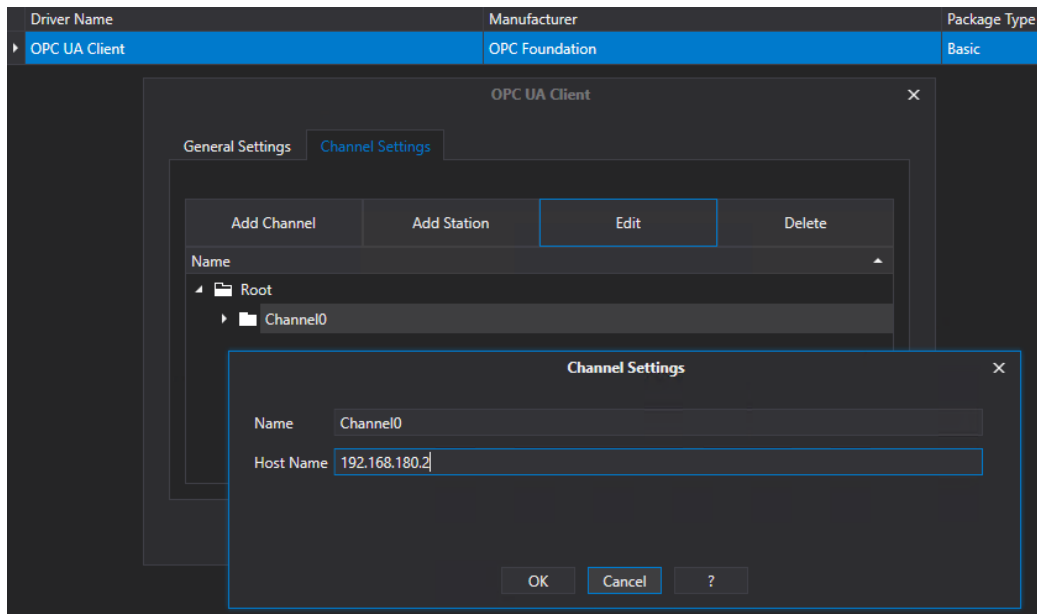
**Note:** To stop the HMI on the workstation press the Alt+F4 keys.

## Downloading Connex/ WebHMI Project to CPL410 Controller

To transfer the WebHMI project from Movicon.NExT development tool to CPL410 controller, we need to change 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**, then double-click on **OPC UA Client**, select **Channel Settings**, click on **Channel0**, and select **Edit**. Change the Host Name to CPL410 vnic IP address: 192.168.180.2 and click **OK** to close all dialogues.

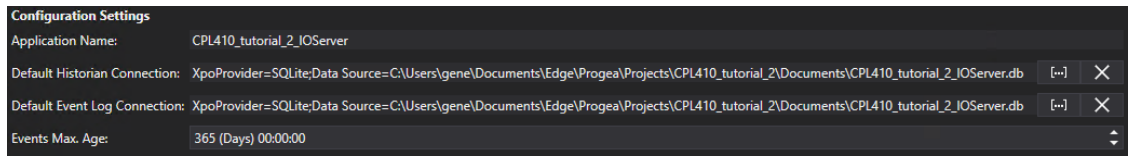
**Figure 71: Host Name**



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

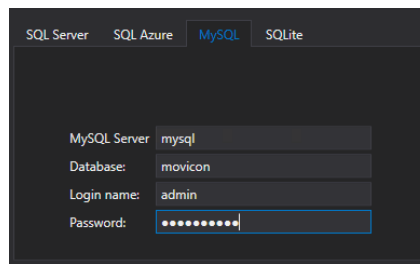
1. In Movicon.NEXt I/O Data Server section double-click on **Settings**.
2. In Default Historian Connection click on the three dots on the right side to access settings.

**Figure 72: Configuration Settings**



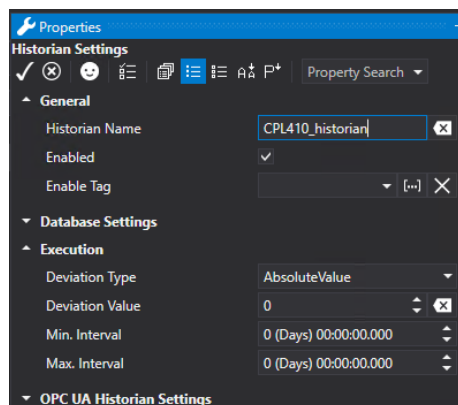
3. Select MySQL tab, enter MySQL as server name, choose a name for the database, login name as admin, and admin password that you have configured for the PACEdge. Click **OK**.

**Figure 73: Password**



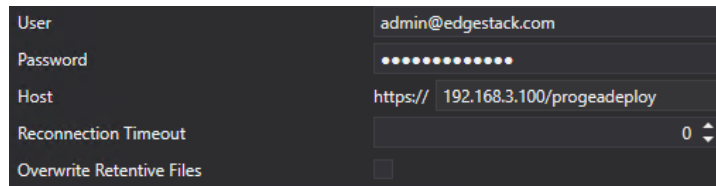
4. At this point, you will get a message saying Unable to connect to the database, which is expected since you cannot connect to MySQL inside the PACEdge from outside. Click **OK** to choose this dialogue.
5. Double-click on **Historians in the I/O Data Server** section, then in the main window right-click 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 74: Properties**



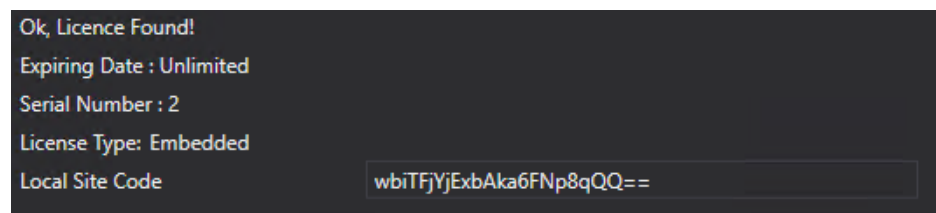
- Right-click on the newly created historian and select **Assign Tag**, navigate to Sweep Time variable, and click **OK**.
- Save the Movicon.NExT project and click on the button **Create WebHMI**. Click on **Yes** to run WebHMI Server from remote.
- In the Deploy Project window that opens enter username and password, which match the settings in the appsettings.json file as described in section 3.5.3, *Changing Connex Password*. (Default password is: Edgestack123!)

**Figure 75: Connex Login**



- Click on the **Connex** button, respond with Yes to Deploy the Project, and the first time you run, to upload I/O Data Server and WebHMI questions.
- 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.
- 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 one, please make sure to first 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.
- From the same Deploy Project window, you can also check License status. Expand Remote License Info and then click on Check Remote License. The CPL410 license will look like the following:

**Figure 76: License Found**



- At this point, go to the PACEdge landing page and refresh the browser. WebHMI Icon should become active, which will bring 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 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, make sure Ethernet cable is plugged into LAN1 port on the controller and workstation is configured to communicate on 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, make sure Ethernet cable is plugged into ETH port on the controller and 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:

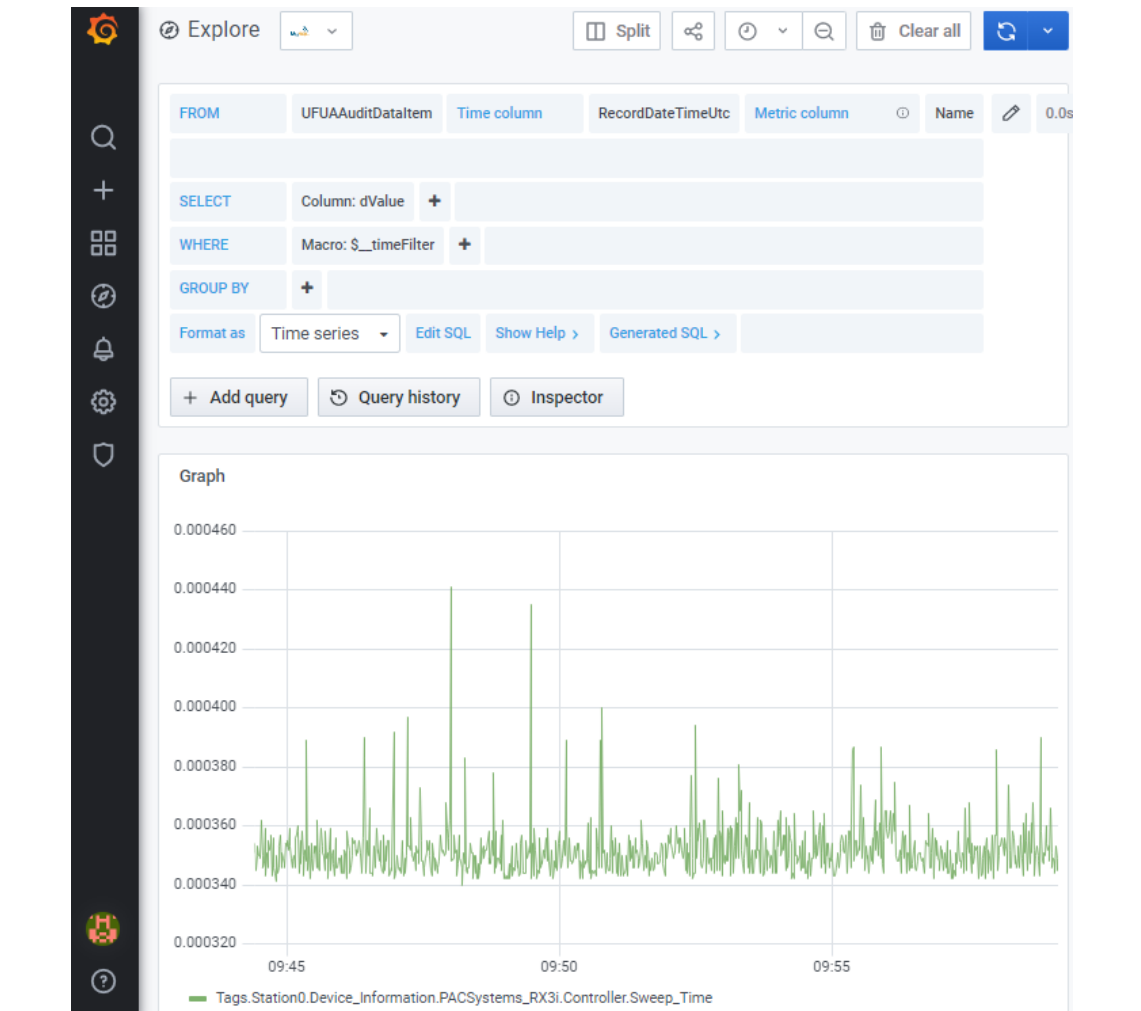
7. From the PACEdge landing page click on the **Grafana** link and make sure to login as an admin.
8. On the left side, move the cursor over the **Configuration** icon and select **Data Sources**.
9. Click on **Add Data Source** and select **MySQL**.
10. Enter values as shown and admin password, click on Save & Test on the bottom. You should get a message **Database Connection OK**.

Figure 77: MySQL

The screenshot shows the configuration page for a MySQL data source in Grafana. At the top, there is a 'Name' field containing 'MySQL' and a 'Default' toggle switch which is turned on. Below this is the 'MySQL Connection' section with fields for 'Host' (mysql), 'Database' (movicon), 'User' (admin), and 'Password' (masked with dots). There are also checkboxes for 'TLS Client Auth', 'Skip TLS Verify', and 'With CA Cert'. The 'Connection limits' section includes 'Max open' (unlimited), 'Max idle' (2), and 'Max lifetime' (14400). The 'MySQL details' section has a 'Min time interval' field set to '1m'. At the bottom, there is a 'User Permission' section with a warning message: 'The database user should only be granted SELECT permissions on the specified database & table to validate that queries are safe so queries can contain any SQL statement. For example, statement DROP TABLE user; would be executed. To protect against this we highly recommend you only grant SELECT permissions. Checkout the MySQL Data Source Docs for more information.' At the very bottom, there are three buttons: 'Save & Test' (blue), 'Delete' (red), and 'Back' (grey).

11. On the left sidebar, click on the **Explore** icon.
12. Make sure the newly configured MySQL database is selected, then in the Metric column select Name, and the **SELECT** field click on **the Column: OID** and select **dValue**. You should see the historical values of the Sweep Time data tag.

Figure 78: Sweep Time Data



## 5.7 Configuring Telegraf

### 5.7.1 Example Description

PACEdge 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](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.

### 5.7.2 Prerequisites

If new to Node-RED, completion of the 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.

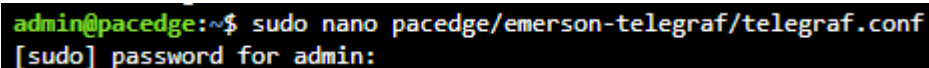
### 5.7.3 Example Walk Through

Steps to access Telegraf configuration file:

1. From the PACEdge landing page open Cockpit, login as an admin user.
2. Go to Terminal application and type in the following command, enter admin password (if requested).

---

**Figure 79: Credentials**



```
admin@pacedge:~$ sudo nano pagedge/emerson-telegraf/telegraf.conf  
[sudo] password for admin:
```

3. This will open the Telegraf configuration file. To navigate the cursor you need to use arrow keys on your keyboard. Once done editing
  - a. CTRL + x
  - b. Y (to Save modified buffer?)
  - c. Enter (keeping same file name)
4. For the configuration changes to take effect Telegraf needs to be restarted, either via Cockpit or Portainer. Alternatively, the whole IPC can be rebooted.

By default, Telegraf is configured to pull the system it is running on, statistics, and dump them into InfluxDB every 10 seconds to gain access to InfluxDB telegraf.conf needs to have a valid username and password.

Note that the password in this file is stored in clear text, hence do not use the admin username and password. If desired, you could even create a dedicated user inside InfluxDB for Telegraf and use that username and password. In this example, we use the developer user and password that was given to the developer user.

By default, Telegraf is configured to use database telegraf\_metrics where all data will be saved.

Figure 80: Telegraf Configuration File

```
### OUTPUT

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

  ## Retention policy to write to. Empty string writes to the default rp.
  retention_policy = ""
  ## Write consistency (clusters only), can be: "any", "one", "quorum", "all"
  write_consistency = "any"

  ## Write timeout (for the InfluxDB client), formatted as a string.
  ## If not provided, will default to 5s. 0s means no timeout (not recommended).
  timeout = "5s"
  username = "developer"
  password = "edgestackd"
```

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 there is a section on what inputs should be captured and stored.

Figure 81 Defining Inputs to be Captured and Stored

```
# 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
```

By default, you can see that CPU, disk usage, and disk performance statistics are enabled. User can manipulate these settings and tune what statistics she/he is interested in.

# Section 6: PACEdge Performance Indication

PACEdge software is based on flexible, open-source, and easy-to-use software components, which enables fast application development. This section is meant to provide performance expectations and guide the user in selecting a suitable PACEdge hardware option as well as 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 PACEdge via the locally (DisplayPort) attached monitor and local browser puts extra workload on the PACEdge unit as compared to the remotely accessing PACEdge 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 PACEdge is being used with locally attached Monitor and browser inside the PACEdge, where is “Rem” indicates that PACEdge is being accessed remotely via Ethernet.

## 6.1 OPC UA Performance Indication

OPC UA Performance test was done by using two PACEdge devices: one configured to act as OPC UA Server (highest performance IPC, RXi2-UP) and the other one as 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:
  - a. The client sends one read request containing 100 variables
  - b. The server responds with 10 replies, each containing 10 variable updates
  - c. Once all replies are received, the Client repeats the sequence

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

Figure 82: 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

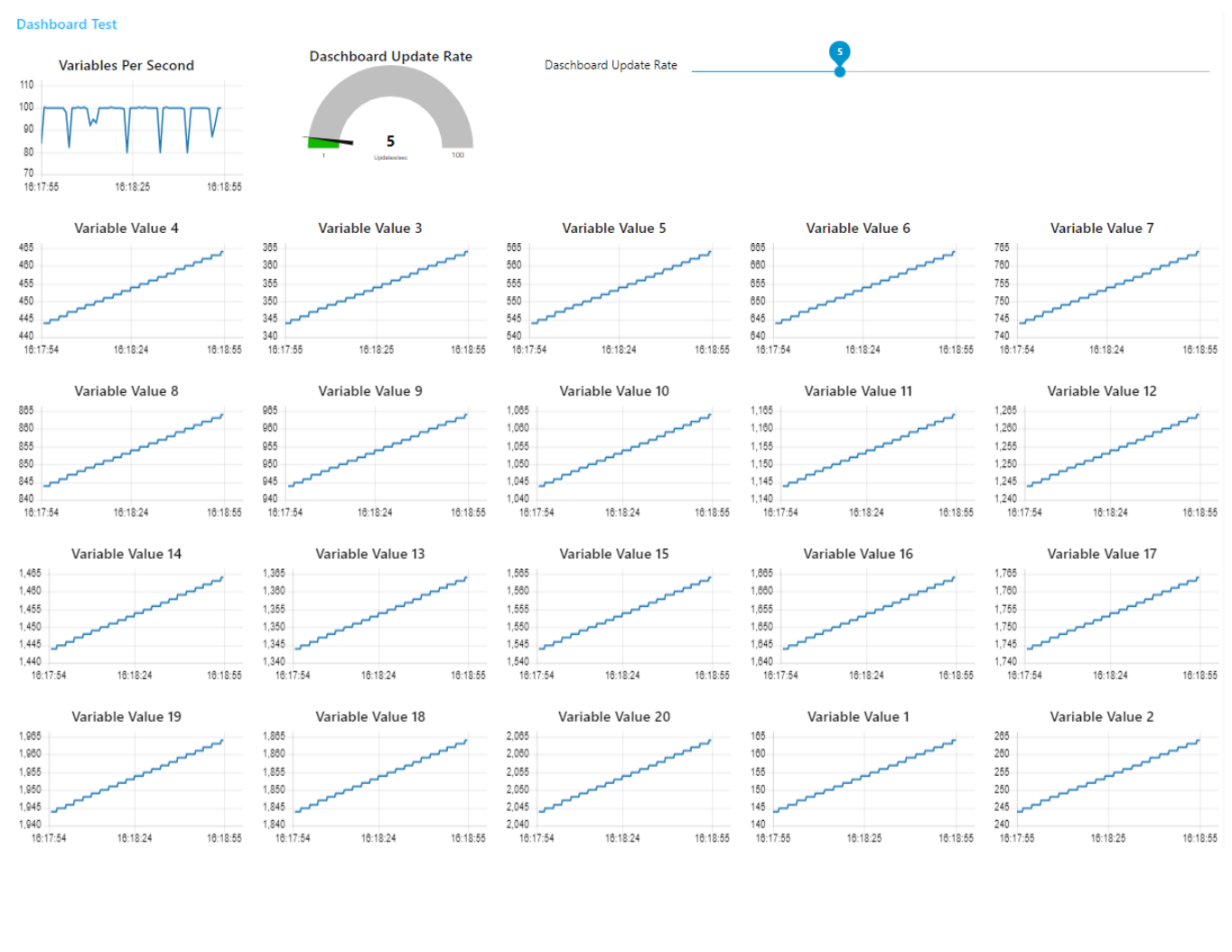
## 6.2 Node-RED Dashboard Performance Indication

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.

Table 6 Node-RED Dashboard Performance

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 83: Dashboard Update Rate



## 6.3 Node-RED Modbus TCP Performance Indication

In this test, the PACEdge device with Modbus TCP (Server) is reading 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, 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 Connexth Historian Performance Indication

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

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

## Section 7: Saving and Restoring User Data

PACEdge 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. If desired, 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/`.

Following are the most critical applications/folders:

- emersonedgestack\_grafana-storage
- emersonedgestack\_influxdb
- emersonedgestack\_mysql
- emersonedgestack\_nodered
- emersonedgestack\_portainer
- emersonedgestack\_progea


Note, that actual data to be stored/restored is in `/home/admin/docker/volumes/application-name/_data/` folder. When restoring data, copy back folders into this location. Note, access requires admin user rights.

Alternatively, Node-RED flows, 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 and can be restored by performing a new upload to the PACEdge 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 PACEdge) 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 the desired flow xxx.json file on your client system (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 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 Grafana application:

1. Navigate to **Create** icon on the left side  and select the **Import** option.
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 PACEdge device. Therefore there is no need to export projects from PACEdge. When recovery is needed, projects can be loaded once again from the engineering workstation onto the PACEdge device.

## 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 84: Example paths**

```
admin@pacedge:~$ 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@pacedge:~$ 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 at a later time.

**Note:** Access to these folders requires admin privileges.





## 7.5 Saving License Files

PACEdge devices come with pre-installed software and with valid license files from the factory. Under normal circumstances user does not have to worry about keeping a copy of the valid license file, but restore to PACEdge 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 PACEdge 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, *PACEdge License File*, but an alternative would be to make a copy of the license file before performing a restore operation. The steps below will document how to extract the current license file and store it on a USB stick for later use:

1. Plug the USB stick into one of the USB ports on RXi2-BP, RXi2-LP, or into USB1 on CPL410/CPE400.
2. Go to Cockpit, and log in as the admin.

- 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 85: USB Device List**

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

- 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 86: Mounting the Device**

Storage > SanDisk Extreme Pro (123116791C7D)

Drive

Model	Extreme Pro
Firmware Version	0
Serial Number	123116791C7D
Capacity	119 GiB, 128 GB, 128043712512 bytes
Device File	/dev/sdb

Content Create Partition Table

119 GiB vfat File System /dev/sdb1

Partition	Filesystem	Mount	⋮
Name	-		
Size	119 GiB		
UUID	0262b1a5-01		
Type	0x0c		

- In the dialogue box that opens, enter Mounting Point: `/media/USB` and click on the **Mount** button.

**Figure 87: Mount Device**

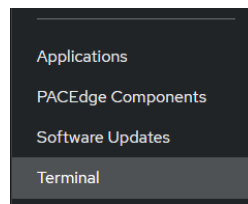
### Mount Filesystem

Mount Point

- Mount Options
- Mount read only
  - Custom mount options

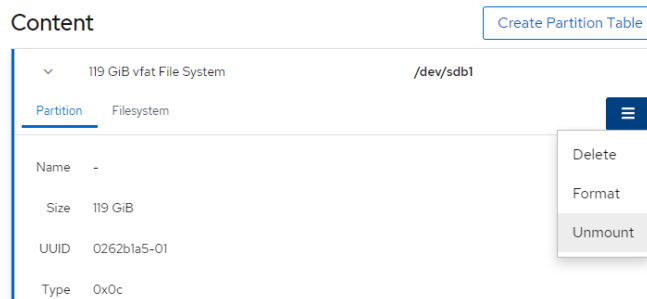
- Within the Cockpit, go to the Terminal screen and enter the following commands:

**Figure 88: Open Cockpit's Terminal**



- `sudo cp pacedge/emerson-software/license/license.json /media/usb`
  - Enter admin password (if requested).
  - `sudo cp pacedge/emerson-software/license/license.sig /media/usb`
- Go back to Storage->USB stick, click on the **More Options** button on the right side and select **Unmount**.

**Figure 89: Unmount**



- You may remove the USB stick and save the two files: 1) license.json and 2) license.sig on your Windows workstation.

# Section 8: PACEdge Software Backup/Recovery

PACEdge software allows user to install additional software packages and gives full flexibility to make configuration changes. It is expected that during the development and validation phases user will end up in the situation wanting to either backup current setup or restore the original PACEdge state.

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

The restore procedure allows users to restore PACEdge 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 PACEdge 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 PACEdge 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 Default please enter the UEFI Settings menu and choose the recovery option in the Save & Exit page.

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

## 8.1 PACEdge Software Backup on RXi2-BP, RXi2-LP IPCs

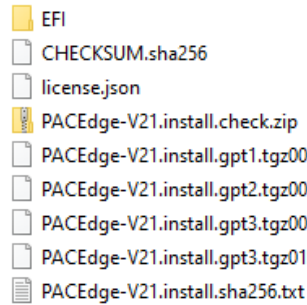
To start, please download the PACEdge Software Recovery – Emerson System Installation image from the Emerson support site. Then follow the steps below to perform a backup:

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 that have occurred on the USB stick. The workaround, in this case, is either: 1) use a brand new USB stick, 2) download from Internet one of Linux installation ISO images, use 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.

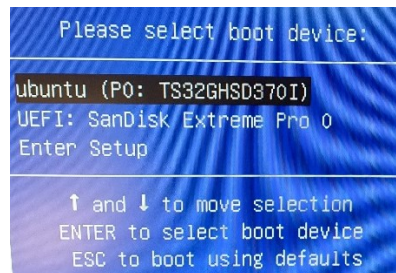
- Copy all files to the root directory of the USB stick. The directory structure should look like this:

**Figure 90: Copy All Files to the Root Directory of the USB Stick**



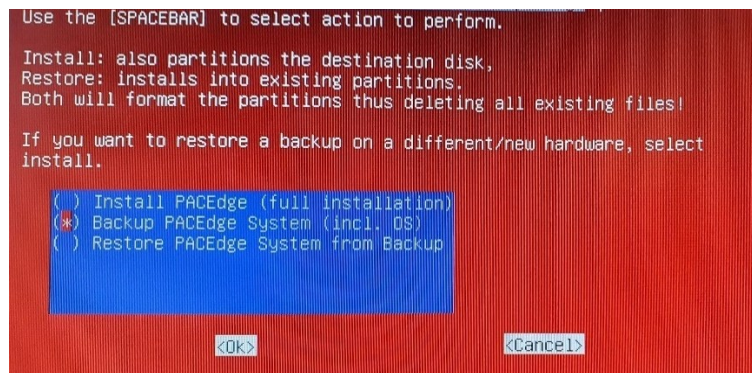
- Plug the USB stick into any of the USB ports, power up the IPC, and keep pressing the F7 button to get into the boot selection menu, select the USB Stick to boot from (in the figure below, this would be the second line item: UEFI: SanDisk Extreme Pro 0.)

**Figure 91: Select the Boot Device**



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

**Figure 92: Select the Backup Option**



5. Wait for the completion message as shown below. Depending on the hardware type it might take between five and 20 minutes to create a backup.

Figure 93: Backup Completed Screen

```
tar: ./var/snap/ixd/common/ixd/unix.socket: socket ignored
Created archive /mnt/install/backup.2021-09-02-11-00.gpt2.tgz
*****
*
*   PACEdge Backup completed:   *
*   Thu Sep  2 12:07:40 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#
```

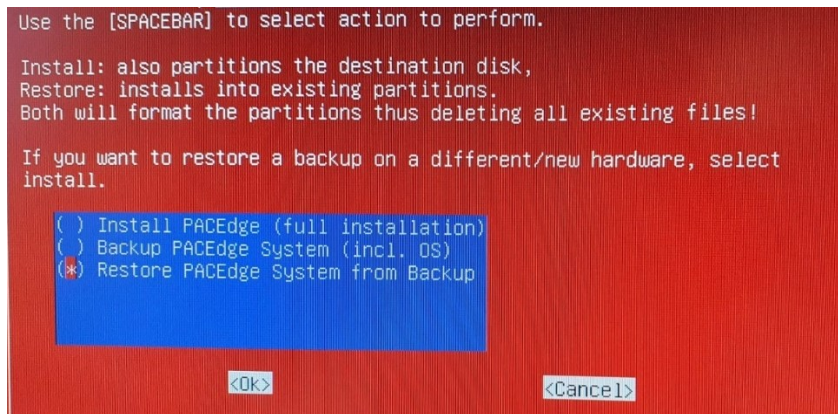
**Note:** After the backup, on the USB stick, you will see the second set of installation files, with word **backup** in them.

## 8.2 PACEdge Software Restore on RXi2-BP, RXi2-LP IPCs

To restore PACEdge software please perform the following steps:

1. Perform steps 1-5 documented in the Section 8.1 *PACEdge Software Backup on RXi2-BP, RXi2-LP IPCs*.
2. Wait until the following dialog appears and then use arrow keys to move the cursor and space key to select the **Restore PACEdge** option. Then use the Tab key to move the cursor to the **OK** button and hit enter.

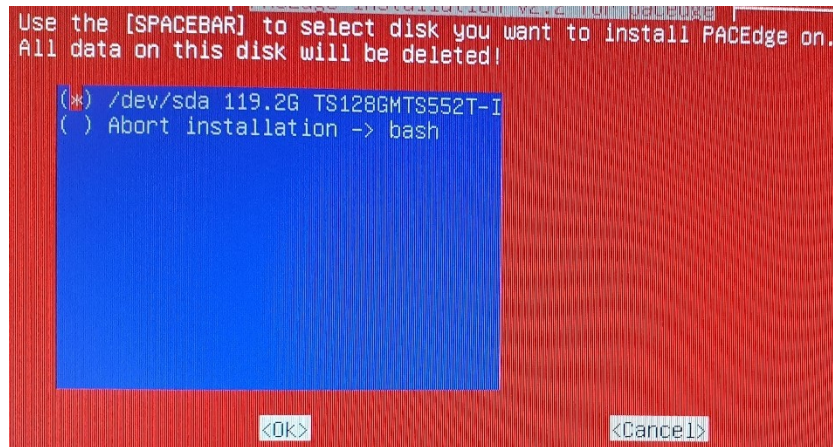
Figure 94: Select the Restore from Backup Option



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

---

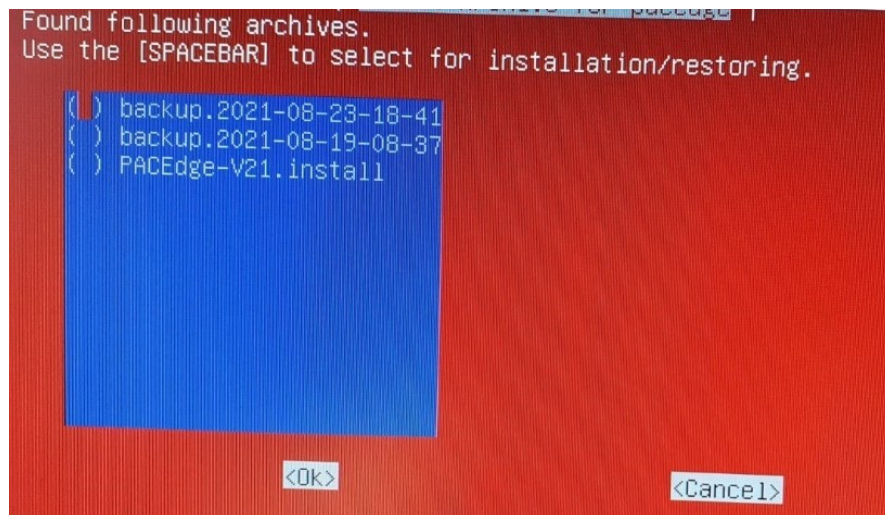
**Figure 95: Select the Disk Partition (If Applicable)**



4. Using arrows, Space, and Tab keys to select either a factory default image (PACEdge-V21.install) or one of your backups, then select **OK** and hit enter.

---

**Figure 96: Select the Backup Image**



5. Wait until the restore procedure completes, as indicated by a similar message.

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

Figure 97: PACEdge 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)
-----
*****
*                                     *
*          PACEdge 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#
```

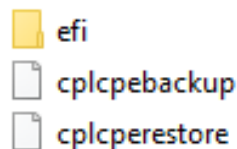
Note: If you have backed up a PACEdge device with a valid license and then restored it on the same physical unit, the license will be also restored and will remain valid. If restore is done on a different physical unit, the license will be invalid and will require a new license file to be manually installed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

## 8.3 PACEdge Software Backup on CPL410, CPE400 Controllers

To perform PACEdge software backup a PACEdgeV2.1-CPL-CPE-BackupRestoreTool needs to be downloaded from the support website. Please use the following procedure to perform a backup:

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

Figure 98: Copy Files to USB Stick



2. Insert USB-Stick into CPL/CPE, USB1 port.
3. Login into Cockpit as admin

- 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 99: List of Filesystems

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

- 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 100: Mount the Device

Storage > SanDisk Extreme Pro (123116791C7D)

Drive

Model: Extreme Pro

Firmware Version: 0

Serial Number: 123116791C7D

Capacity: 119 GiB, 128 GB, 128043712512 bytes

Device File: /dev/sdb

Content Create Partition Table

119 GiB vfat File System /dev/sdb1

Partition: Filesystem Mount ☰

Name: -

Size: 119 GiB

UUID: 0262b1a5-01

Type: 0x0c

- In the dialogue that opens, enter Mounting Point: `/media/usb` or `/mnt/usb` and click on the **Mount** button.

**Figure 101: Mount File System**

### Mount Filesystem

Mount Point

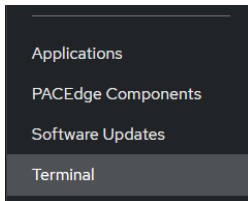
- Mount Options  Mount read only  
 Custom mount options

Cancel

Mount

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

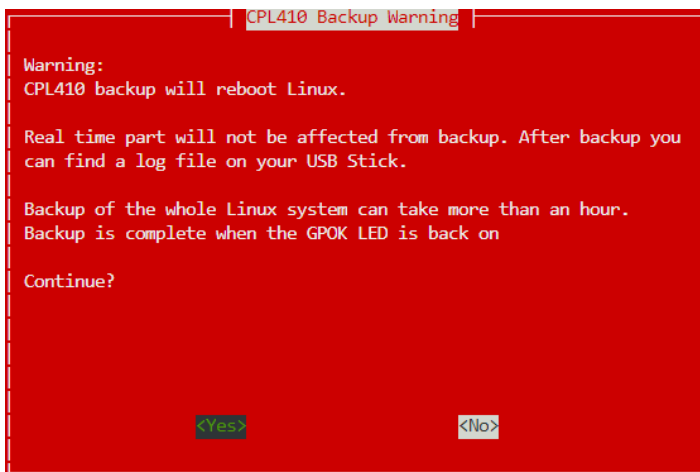
**Figure 102: Cockpit's Terminal**



- `cd /media/usb/`
- `sudo ./cplcpebackup`
- Enter the admin password if asked for.

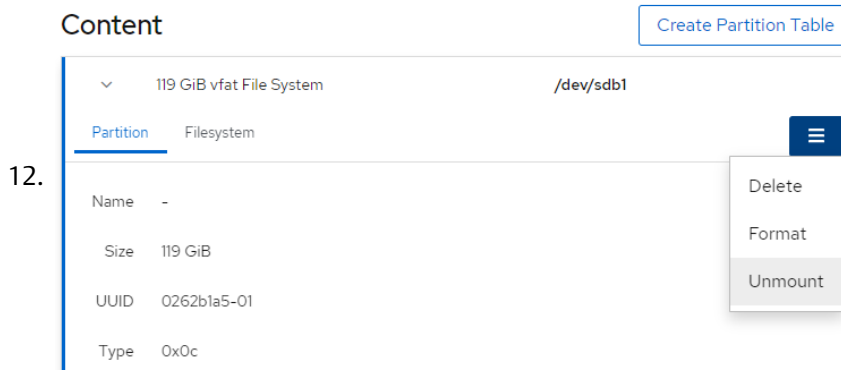
- Using arrow keys, select **Yes** and press **enter**.

**Figure 103: Select the Backup**



10. When the backup starts, GPOK LED will turn off. It will remain off until backup is complete and PACEdge side of the controller is rebooted and GPOK LED is ON. It might take close to 2h to perform a backup.
11. Before removing the USB stick, go back to Cockpit, Storage->USB stick, click on the **More Options** button on the right side and select **Unmount**.

Figure 104: Unmount the Device



13. 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 PACEdge Software Restore on CPL410, CPE400 Controllers

### 8.4.1 CPL410, CPE400 PACEdge 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. Note that switching to 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 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 update status via display:
  - a. Press the DISP button until the **Edge Settings** option is selected, then press SEL.

- b. Check that message **Resetting GP** is shown. Also, blinking SSD LED will indicate recovery being in progress.
3. Wait until GPOK LED turns ON. This can take up to 10 minutes.
4. At this point Factory Default restore is done. PACEdge is functional and the License file is valid.

## 8.4.2 CPL410, CPE400 Restore of PACEdge Backup Image



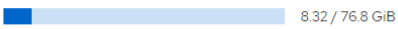

If there was a PACEdge backup image created, it can be later restored on the same HW family unit (meaning CPL410 to CPL410 or CPE400 to CPE400). Note that if restoring onto a physically different unit new valid licenses will be required afterward to activate the PACEdge.

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

To perform PACEdge software restore from backup use the same USB stick that was used to create a backup. Please use the following procedure to perform a backup

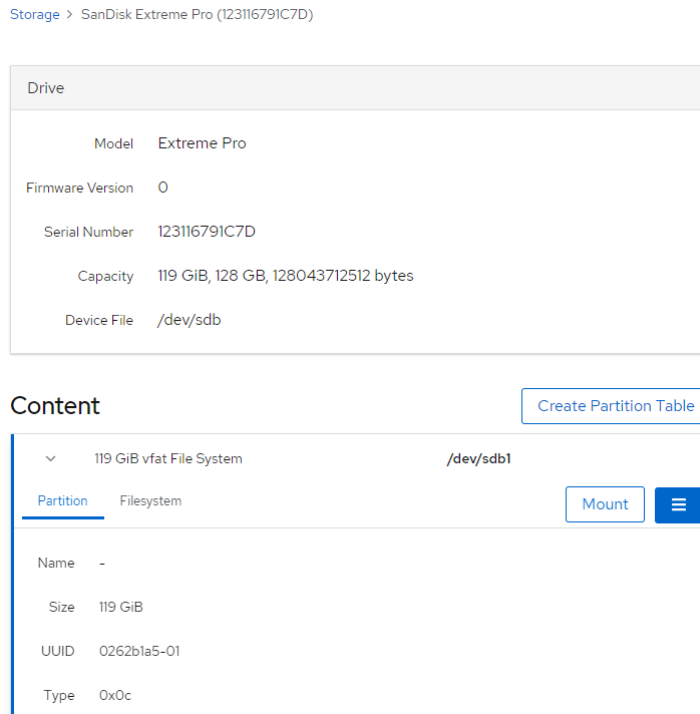
1. Insert USB-Stick into CPL410/CPE400, USB1 port.
2. Login into Cockpit as admin.
3. Go to the Storage tab on the left side of the screen, look for the USB stick in the list of Filesystems (typically at the very bottom of the list, UBUNTU-SERV in the example below), and click on it.

**Figure 105: List of Filesystems**

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

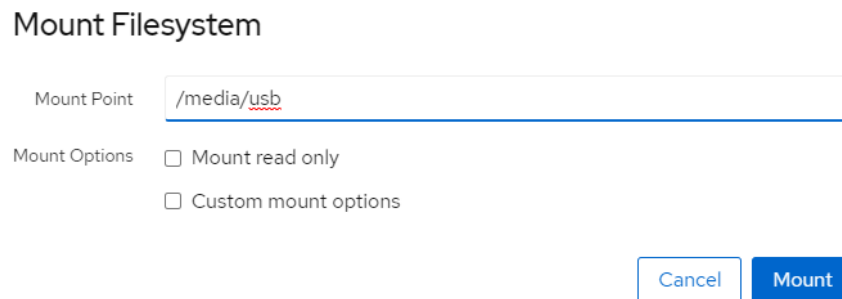
4. On the next page, click to expand the view of the Content and then click on the **Mount** button on the right side of the screen.

**Figure 106: Mount the Device**



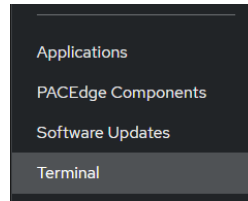
5. In the dialogue that opens, enter Mounting Point: **/media/USB** or **/mnt/USB** and click on the **Mount** button.

**Figure 107: Mount the Filesystem**



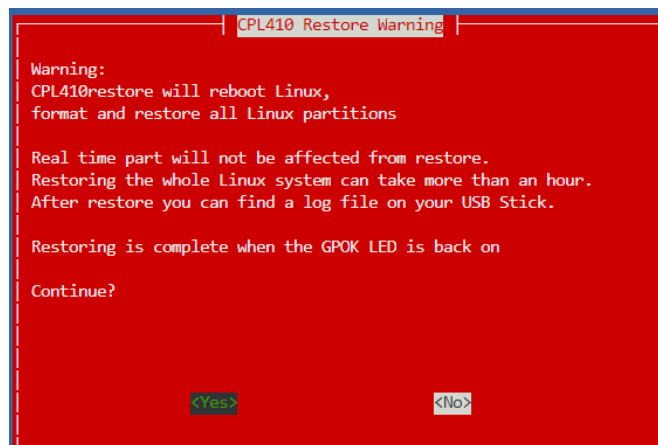
- Now, within the Cockpit, go to the Terminal screen and enter the following commands:

**Figure 108: Cockpit's Terminal**



- `cd /media/usb/`
  - `sudo ./cplcperestore`
  - Enter the admin password (if applicable).
- Using arrow keys select **Yes** and press **enter**.

**Figure 109: Continue with the Restore**



- Using arrow keys and space keys mark the backup image that you would like to restore then with the Tab key select **OK** and press **enter**.

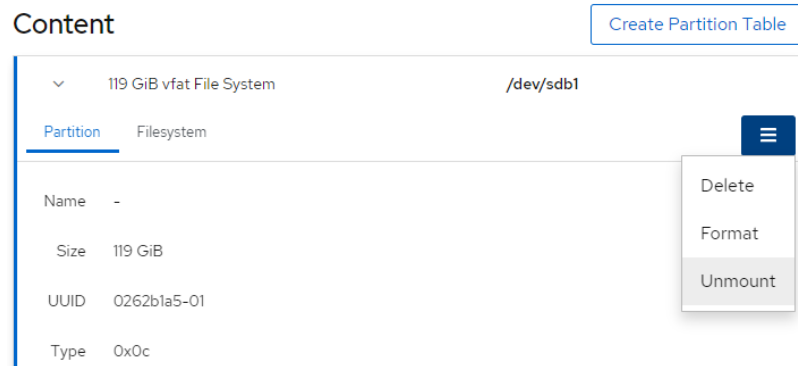
**Figure 110: Select the Backup File**



- When restore starts, GPOK LED will turn off. It will remain off until backup is complete and PACEdge side of the controller is rebooted and GPOK LED is ON.
- Before removing the USB stick, go back to Cockpit, Storage->USB stick, click on the **More Options** button on the right side and select **Unmount**.

---

**Figure 111: Unmount the Device**



- 
- Remove the USB stick.

# Section 9: Upgrade from Earlier PACEdge Versions

## 9.1 Upgrading RXi2-BP, RXi2-LP from PACEdge 2.0 to 2.1

RXi2-BP and RXi2-LP units with PACEdge 2.0 can be upgraded to PACEdge 2.1. Note that in PACEdge 2.1 an active license enforcement has been introduced and therefore a valid license will be required after the upgrade for the PACEdge features to be active. Please refer to section 3.7 PACEdge License File for how to obtain a valid license once the PACEdge software has been upgraded.

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

To start, please download the PACEdge Software Recovery – Emerson System Installation image from the Emerson support site. Then follow the steps below to perform an upgrade:

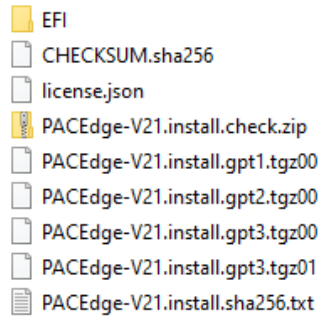
1. Backup all user data (Node-RED flows, Grafana dashboards, MySQL and InfluxDB databases) to USB stick or network. Upgrade procedure will completely wipe clean internal storage
2. Get an empty, min 32GB size USB stick. Make sure it is empty as conflicts might arise.

**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 that have occurred on the USB stick. The workaround, in this case, is either: 1) use a brand new USB stick, 2) download from Internet one of Linux installation ISO images, use 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:

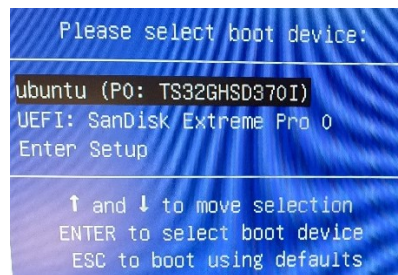
---

**Figure 112: Copy Files to Root Directory of USB Device**



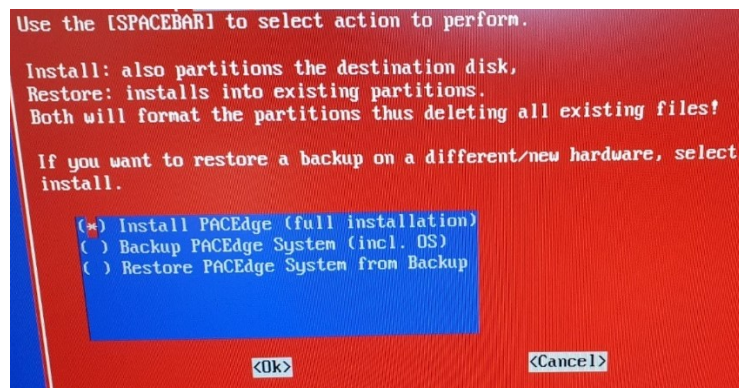
4. Plug the USB stick into any of the USB ports, power up the IPC, and keep pressing the F7 button to get into the boot selection menu, select the USB Stick to boot from (in the figure below, this would be the second line item: UEFI: SanDisk Extreme ...)

Figure 113: Select the Boot Device



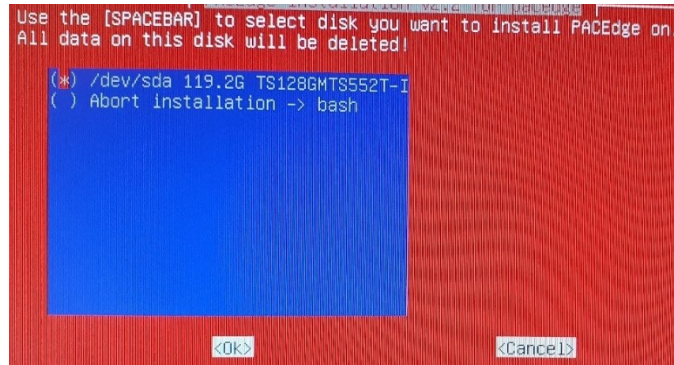
5. Wait until the following dialog appears, use arrow keys to move the cursor and space key to select the Install PACEdge option. Then use the Tab key to move the cursor to the OK button and hit enter.

Figure 114: 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.

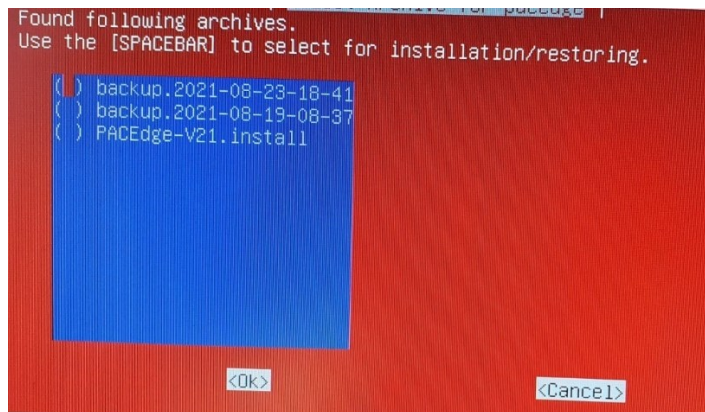
Figure 115: Select the Boot Partition



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

---

**Figure 116: Select a Factory Image**



8. Wait until the install procedure completes, as indicated by a similar message. Note, this step, depending and the HW type used, will take between a couple of minutes to 10-20min.

---

**Figure 117: 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)
-----
*****
*
*          PACEdge 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, *PACEdge License File*

## 9.2 Upgrading CPL410, CPE400 from Older Versions to PACEdge 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 any user data to a USB stick or network storage.

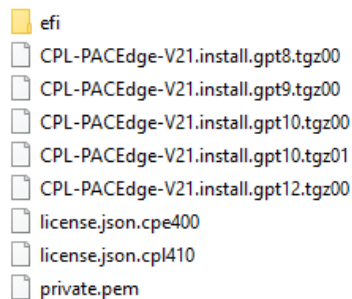
Note: Before performing PACEdge update please update PACs runtime firmware to the latest version. The Minimum PACs version required is 10.15

To start, please download the CPL\_CPE\_Production\_Upgrade\_PACEdge\_2.1 image from the Emerson support 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 that have occurred on the USB stick. The workaround, in this case, is either: 1) use a brand new USB stick, 2) download from Internet one of Linux installation ISO images, use 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, directory structure should look like this:

---

**Figure 118: 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 to the middle. At this point, please remove the USB stick and cycle the power to the unit

**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.

# Section 10: Troubleshooting

## 10.1 Difficulties Accessing PACEdge Components, Erratic Behavior

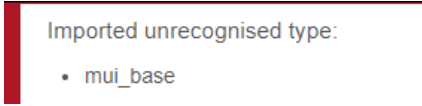
- It is always a good first step to reboot the hardware.
- If you can log in, go to Cockpit and analyze the Logs. Note that logs do contain several warnings and notices about different services as part of normal operation, so search for the messages related to the specific problem.
- In case you have a problem accessing a specific PACEdge application, say Node-RED or 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`.

## 10.2 Error in node-red-contrib-emerson Examples

**Note:** In the default PACEdge Node-RED installation, there are some Emerson prepared examples. These examples report a dependency on the node which is not being used, but mistakenly gets imported together with the flow. If you try to import one of these examples you will get the following error:

---

**Figure 119: Example Error**



```
Imported unrecognised type:  
• mui_base
```

At this point, if you have not pressed yet Deploy, you can discard the example being imported along with the error. If, however, you have Deployed the changes even deleting all flows will still generate the error above. To remove the error follow these steps:

1. In the Node-RED editor, upper right corner, click on the **Configuration Nodes** icon

---

**Figure 120: Node-RED Editor**



2. Next, click to expand option **On all flows**
3. You will see a list of nodes, that are assigned to different categories. At the bottom of that list, you will see one or more nodes in the **unknown** category

---

**Figure 121: Unknown Category**



4. Double-click on each node in the unknown category, then click on **Delete** in the new window that opens
5. After all the unknown nodes are deleted, click on **Deploy**, the error should be gone.

**Note:** after the steps above, the example flows are functional and can be used if desired.

## 10.3 PACEdge Files

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

### 10.3.1 Installation Test

PACEdge comes with a script to test if PACEdge installation is working as expected: *tst\_inst.sh*. To execute this script please change into folder: */home/admin/pacedge/* and run following command:

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

This script should produce an output similar to the following:

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

Wait 30 sec for finishing nodered start process

Check if all packages are installed  
apache2-utils : OK

Check if all images are available

Tag	Container Image Id	Size	Repository
emerson-aes-daemon 1.0.9-edgestack	eb2946d2bf57	182.7 MB	iiot.aventics.com/images/aes-daemon
emerson-chronograf 1.8.8	2f07c735db5e	194.8 MB	chronograf
emerson-grafana 7.5.7	bc8c9ea5532e	201.8 MB	grafana/grafana
emerson-influxdb 1.8.6	66f3325c5416	307.8 MB	influxdb
emerson-mqtt 2.0.11	c3a8baae5c3f	9.918 MB	eclipse-mosquitto
emerson-mqtt-internal-ipc 2.0.11	c3a8baae5c3f	9.918 MB	eclipse-mosquitto
emerson-mysql 10.5.11	8bd09b203b08	407.5 MB	mariadb
emerson-nginx 1.21.0	87560fc16fed	536.4 MB	emerson-nginx
emerson-nodered 1.3.5	2066beab39b7	612.5 MB	emerson-node-red
emerson-php f8ceec02a25f	406.8 MB		php fpm
emerson-portainer 2.5.1	45be17a5903a	209.1 MB	portainer/portainer-ce
emerson-progea latest	a16ebaded4b0	3.713 GB	emerson-progea
emerson-telegraf 1.18.3	34535e6964f5	314 MB	telegraf

```
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
```

## 10.4 Docker Commands

PACEdge 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 PACEdge and you only need some commands to manage PACEdge.

PACEdge uses “`docker -compose`” to start and configure the set of PACEdge 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:

### 10.4.1 `docker-compose up -d`

With this command, the PACEdge environment is started. All the containers, configured in the **`docker-compose.yml`** file, are created, configured, started, and connected. Normally there is no need to use this command, as the PACEdge environment is automatically started during system start. But it is used whenever the configuration has changed or PACEdge has been shut down by the following command:

### 10.4.2 `docker-compose down`

Use this command whenever you want to shutdown PACEdge. 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 PACEdge configuration (see the following command).

## 10.5 `./config-compose.sh`

The script `config-compose.sh`, located in the `/home/admin/pacedge/`, let you select the PACEdge 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, then 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 needs to be a member of the Linux "docker" group. Preinstalled PACEdge 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](mailto:customercare.mas@emerson.com)  
Technical Support: [support.mas@emerson.com](mailto: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](mailto:customercare.emea.mas@emerson.com)  
Technical Support: [support.mas.emea@emerson.com](mailto: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](mailto:customercare.cn.mas@emerson.com)  
Technical Support: [support.mas.apac@emerson.com](mailto:support.mas.apac@emerson.com)

Any escalation request should be sent to [mas.sfdcescalation@emerson.com](mailto:mas.sfdcescalation@emerson.com)

**Note:** If the product is purchased through an Authorized Channel Partner, please contact the seller directly for any 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 proper selection, use, and maintenance of any Emerson product remains solely with the purchaser.

© 2020 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.

