MING Stack: The Power of Balena and N3uron in Industrial Data Management

Overview

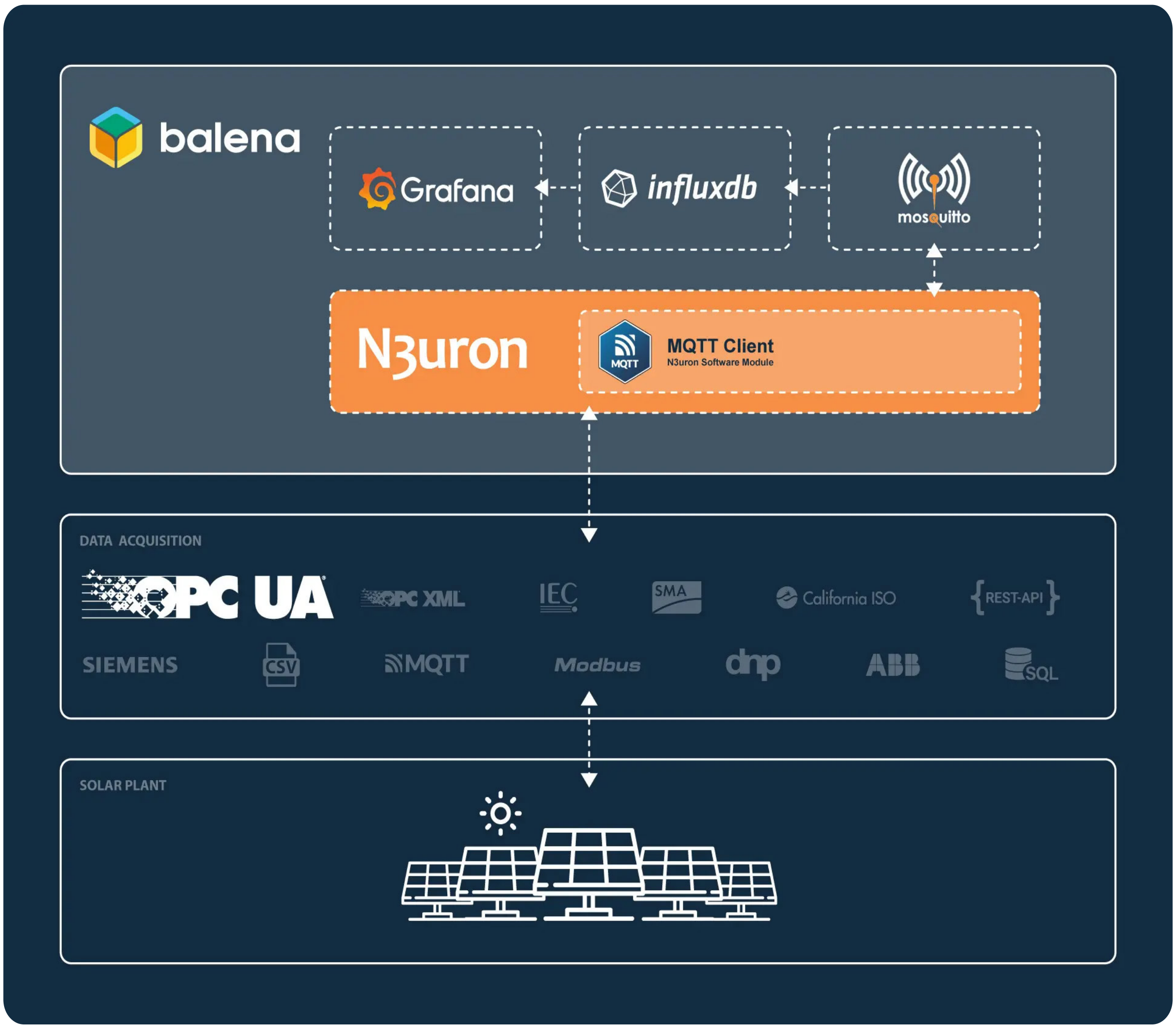
In a previous post, we described how to simplify industrial data management by deploying a complete Industrial Edge node using Docker containers and our customized MING stack (MQTT, InfluxDB, **N3uron** and Grafana). As we explained, this architecture enables consistent, versioned, portable and scalable deployments that increase operational efficiency.

Let's take a closer look at the components of this stack:

- **MQTT:** A lightweight and efficient, publish-subscribe protocol.
- **InfluxDB:** A time-series database optimized for storing, querying and visualizing time-stamped data.
- **N3uron:** A lightweight industrial platform for DataOps that runs on devices at the edge of the network.
- **Grafana:** A popular open-source interactive data visualization platform for visualizing and analyzing data from any source.

When it comes to scaling projects, remotely managing all connected devices can become a daunting task for an IT team. Deploying software updates and critical fixes, as well as performing scheduled maintenance across multiple Edge Nodes and vast geographical distances can be challenging. That's where balena comes in, balena is a complete IoT fleet management platform that allows deploying, managing, and scaling fleets of IoT Linux devices.

In this article, we present a variant of the above-mentioned architecture that uses balena. Additionally, for this example, we will substitute HiveMQ for Mosquitto to make the solution even lighter.



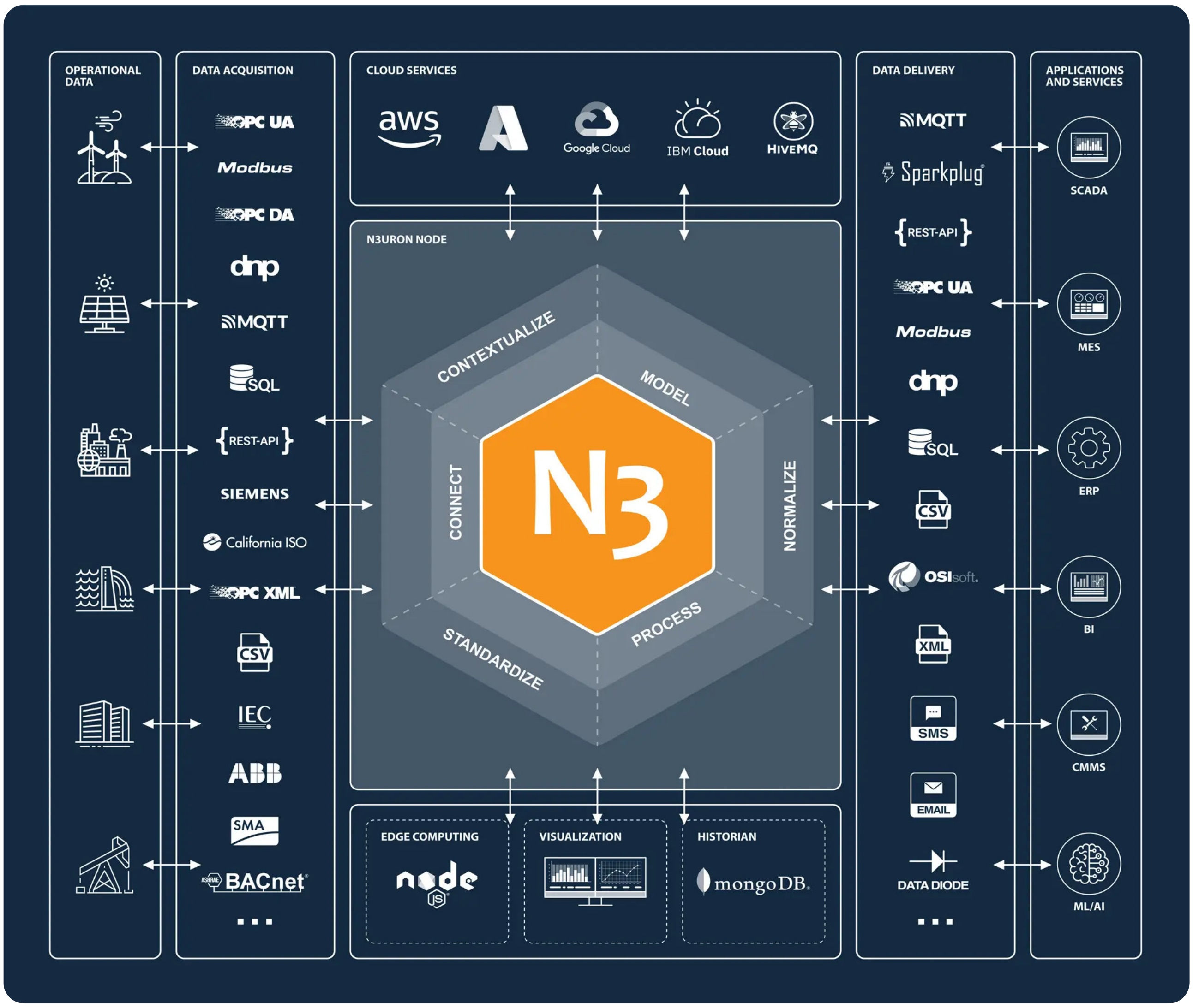
Infographic showing the Streamlined Data Management Process of MING Stack for Solar Plant Data Acquisition with OPC UA Protocol, N3uron MQTT Client Module, Mosquitto, InfluxDB, Grafana, and balena.

We have also drastically simplified the deployment process compared to our previous post. Now Telegraf (responsible for subscribing to the MQTT topics and inserting data into InfluxDB) and InfluxDB work out-of-the-box without any additional steps or configurations.

As you will see for yourself, the resulting solution is easier to manage, operate and scale to address organizations' data challenges than other legacy solutions on the market.

The addition of N3uron to the MING stack provides several advantages including:

● The ability to adapt and scale in complex and growing environments.
● Lightweight and powerful edge computing capabilities.
● Secure communications, modern authentication strategies, user auditing, and control over system access to safeguard data and applications.
● Built-in tools for remote control, data visualization and analysis, provided by modules such as Web Vision and Historian.
● Templates, which allow for rapid building of complex data structures and communication configurations which help drastically reduce the amount of required work for deployments.
● The licensing model is server-based and unlimited, allowing for easy deployment of unlimited web clients, tags, connections, and devices.



Infographic depicting N3uron platform's main features in operational data, data acquisition, delivery modules, applications, cloud services, edge computing, visualization, and historian functionalities.

By leveraging the capabilities of all these technologies working together, businesses can unlock new insights and efficiencies in their operations. Some potential use cases for the MING stack include:

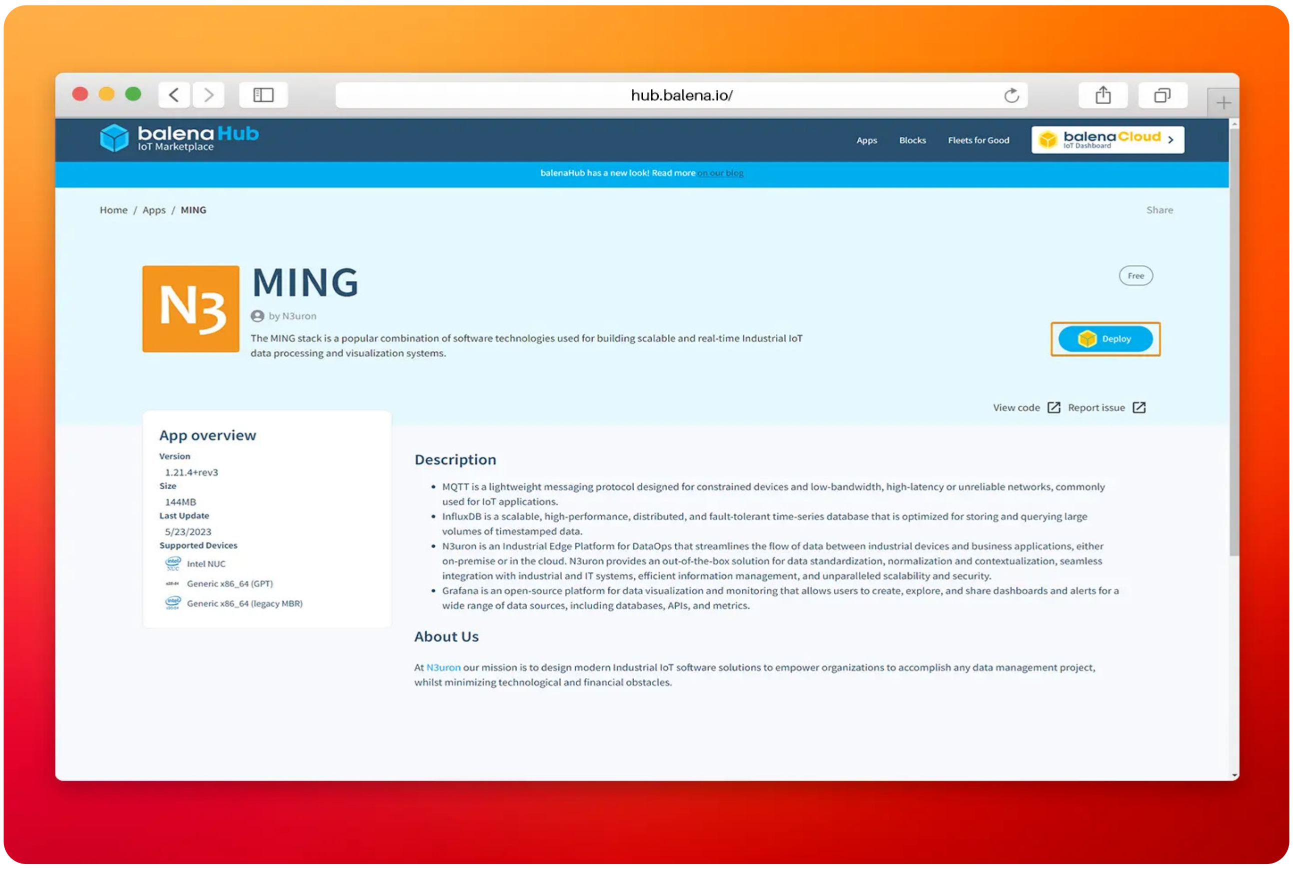
- Real-time monitoring and analysis of industrial equipment and processes
- Integration of industrial data with enterprise systems for business intelligence and decision-making
- Energy management and optimization for industrial facilities
- Remote monitoring and control of distributed assets
- Integration of legacy industrial systems with modern data platforms
- Quality control and assurance for manufacturing processes
- Environmental monitoring and compliance reporting
- Predictive maintenance and anomaly detection for industrial machinery
- Asset tracking and inventory management

Alike in the previous post, once you have completed the configuration steps outlined in the following sections, you will have a fully functional proof of concept with real-time data streaming from a Photovoltaic Plant located in Italy. The data, received through a remote OPC UA Server, will flow seamlessly through your local N3uron instance and be displayed on your Grafana dashboard.

Deploy MING with balena

Deploy from balenaHub (Recommended)

BalenaHub is a marketplace that offers ready-to-deploy applications for small device fleets. With just a few clicks, it is possible to deploy and manage these applications centrally using balenaCloud.



MING

by N3uron

The MING stack is a popular combination of software technologies used for building scalable and real-time Industrial IoT data processing and visualization systems.

App overview

Version
1.21.4+rev3
Size
144MB
Last Update
5/23/2023
Supported Devices
Intel NUC
Generic x86_64 (GPT)
Generic x86_64 (legacy MBR)

Description

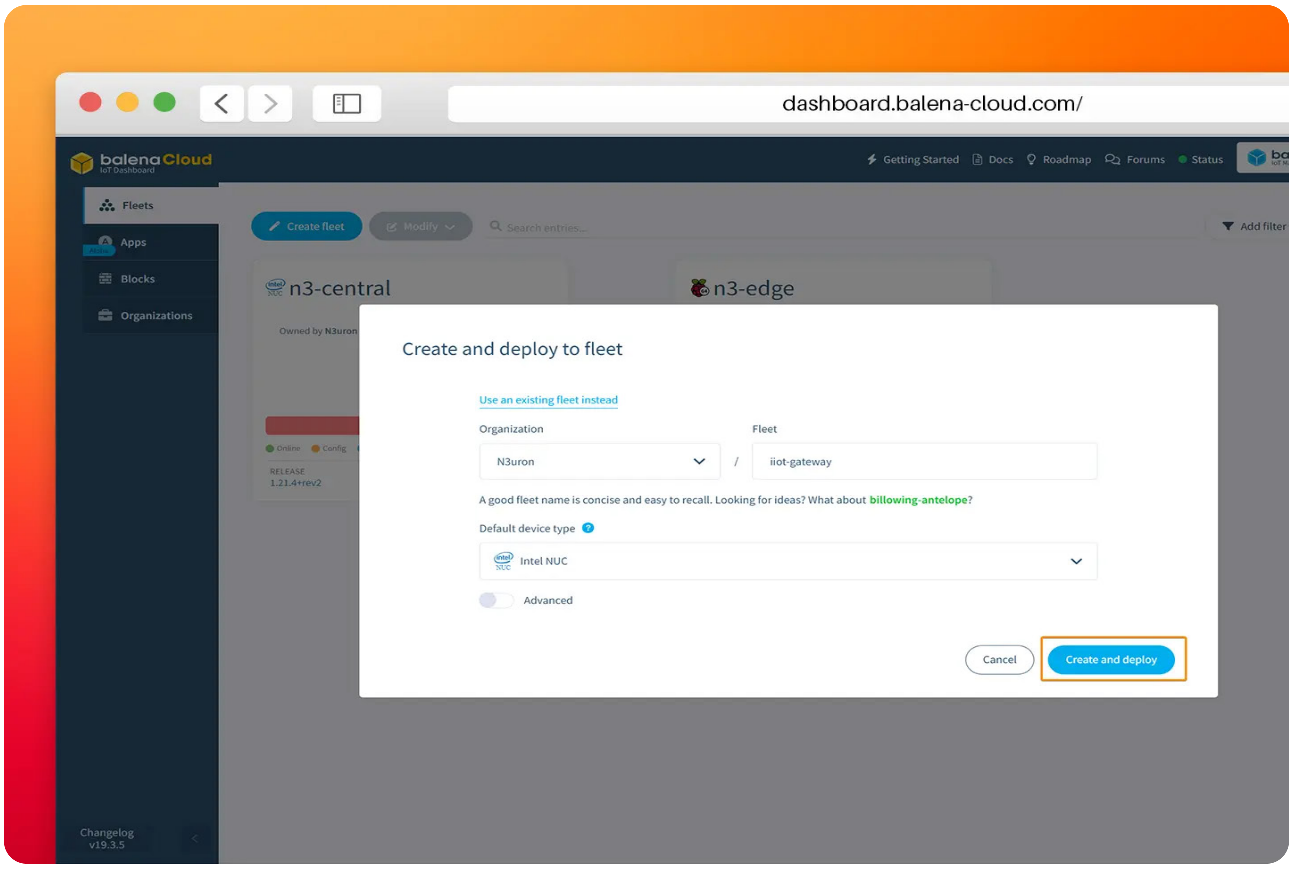
- MQTT is a lightweight messaging protocol designed for constrained devices and low-bandwidth, high-latency or unreliable networks, commonly used for IoT applications.
- InfluxDB is a scalable, high-performance, distributed, and fault-tolerant time-series database that is optimized for storing and querying large volumes of timestamped data.
- N3uron is an Industrial Edge Platform for DataOps that streamlines the flow of data between industrial devices and business applications, either on-premise or in the cloud. N3uron provides an out-of-the-box solution for data standardization, normalization and contextualization, seamless integration with industrial and IT systems, efficient information management, and unparalleled scalability and security.
- Grafana is an open-source platform for data visualization and monitoring that allows users to create, explore, and share dashboards and alerts for a wide range of data sources, including databases, APIs, and metrics.

About Us

At N3uron our mission is to design modern Industrial IoT software solutions to empower organizations to accomplish any data management project, whilst minimizing technological and financial obstacles.

Image displaying N3uron MING Stack from balena Hub web page.

● **Step 1:** To deploy the N3uron application, simply visit https://hub.balena.io/apps/2047510/MING and click on the **Deploy** button. You will then be prompted to log in to your balenaCloud account. Once logged in, you can choose to deploy the app to either a new or an existing fleet.



Create and deploy to fleet

Use an existing fleet instead

Organization: N3uron / Fleet: iiot-gateway

A good fleet name is concise and easy to recall. Looking for ideas? What about billowing-antelope?

Default device type: Intel NUC

Advanced

Cancel | Create and deploy

Screenshot of the balena Cloud panel for creating and deploying to a fleet N3uron.

- **Step 2:** Click on **Add device**.

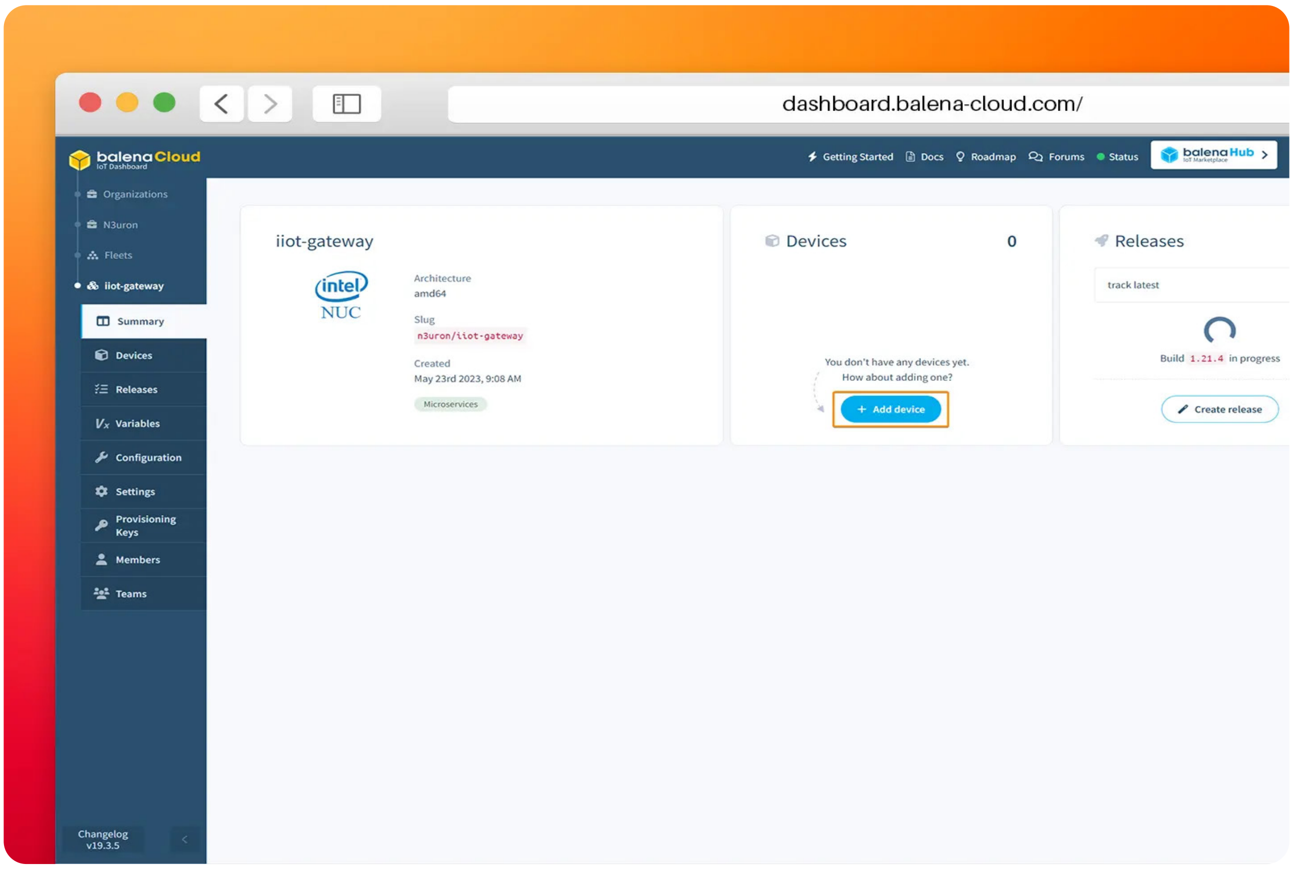
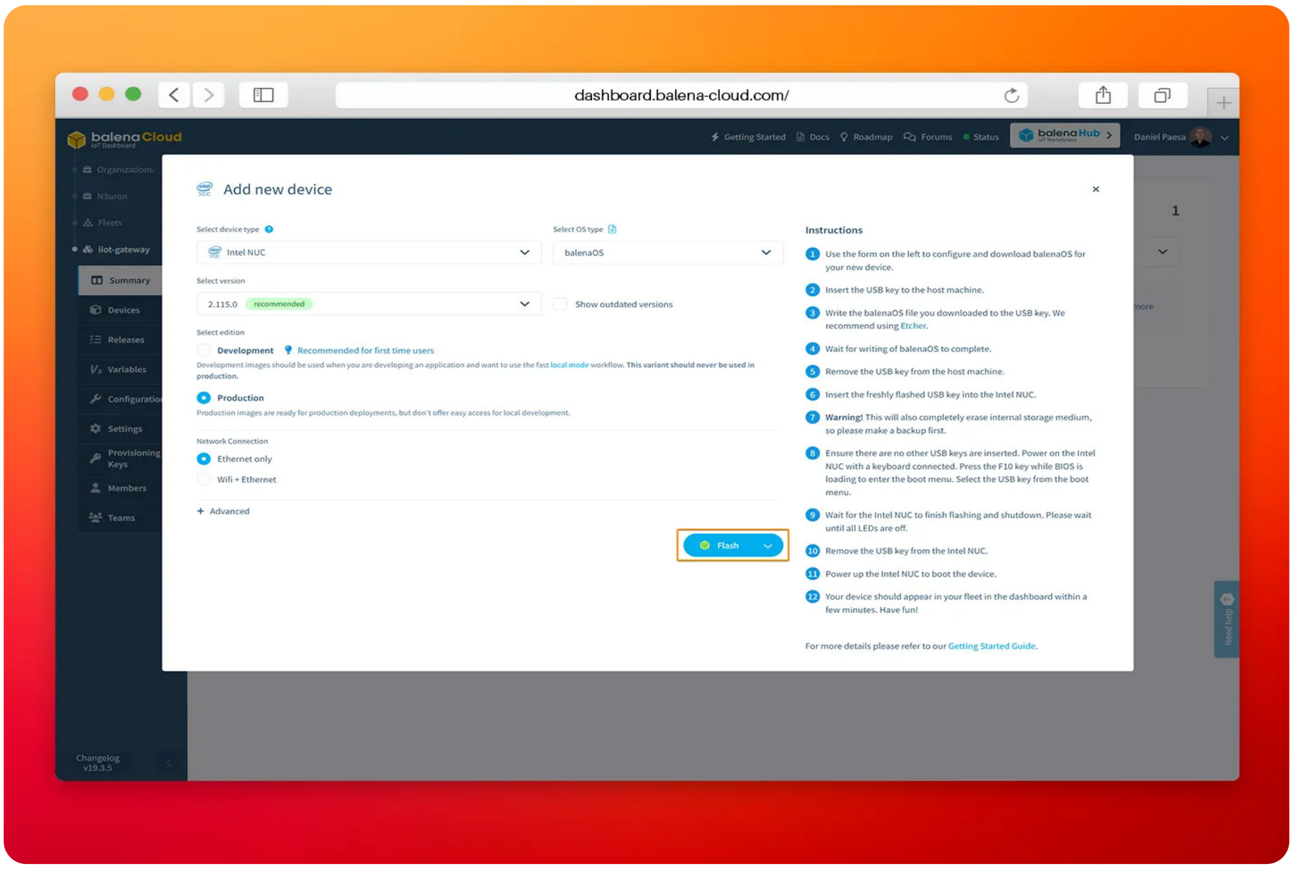


Image showing the balena Cloud summary panel for adding N3uron IIoT Gateway Device.

● **Step 3:** Configure the new device and flash it with balenaEtcher, follow the step-by-step instructions from balena.



Add new device

Select device type: Intel NUC

Select OS type: balenaOS

Select version: 2.115.0 recommended

Show outdated versions

Select edition

Development — Recommended for first time users

Development images should be used when you are developing an application and want to use the fast local mode workflow. **This variant should never be used in production.**

[✓ Production]

Production images are ready for production deployments, but don't offer easy access for local development.

Network Connection

[✓ Ethernet only]

Wifi + Ethernet

+ Advanced

Flash

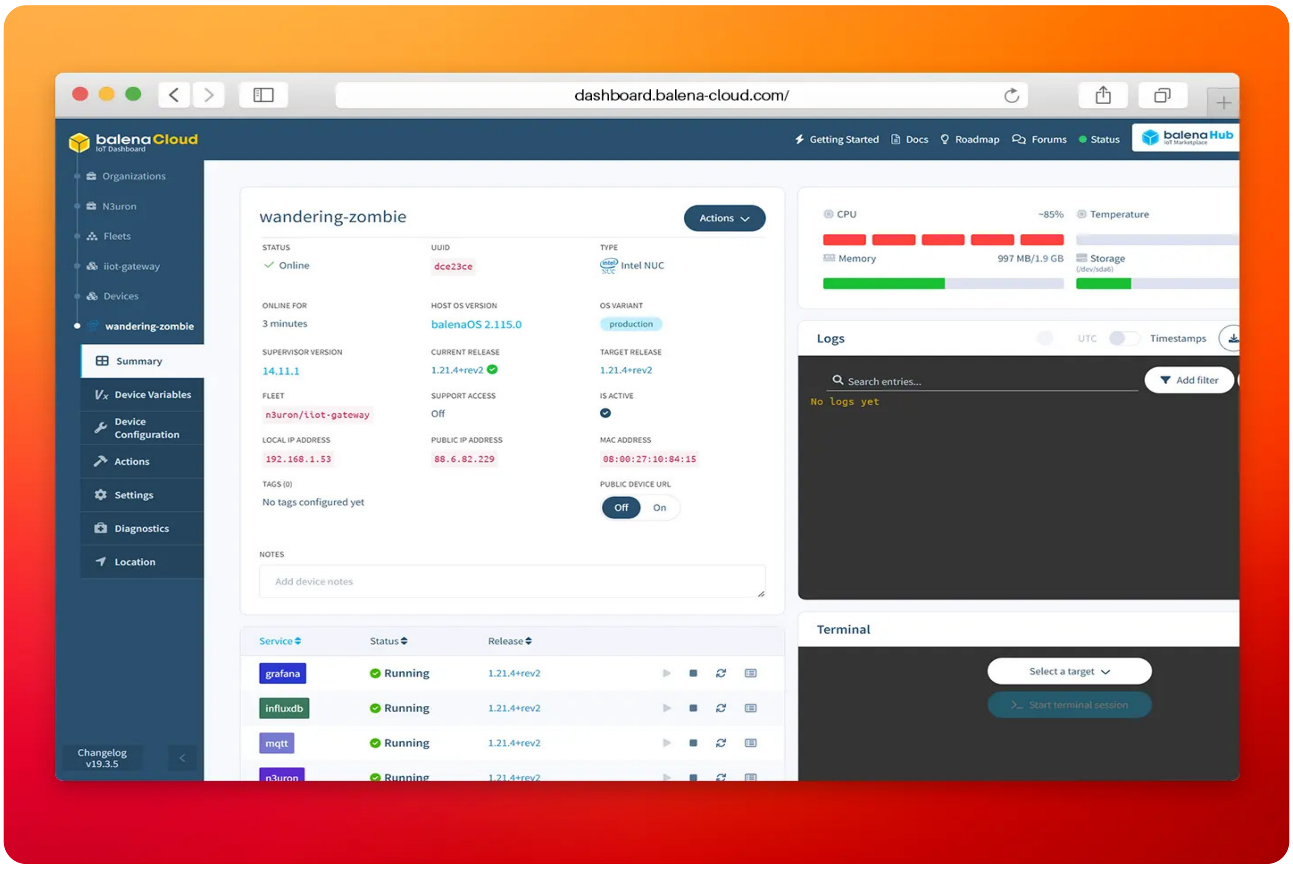
Instructions

1. Use the form on the left to configure and download balenaOS for your new device.
2. Insert the USB key to the host machine.
3. Write the balenaOS file you downloaded to the USB key. We recommend using Etcher.
4. Wait for writing of balenaOS to complete.
5. Remove the USB key from the host machine.
6. Insert the freshly flashed USB key into the Intel NUC.
7. **Warning!** This will also completely erase internal storage medium, so please make a backup first.
8. Ensure there are no other USB keys are inserted. Power on the Intel NUC with a keyboard connected. Press the F10 key while BIOS is loading to enter the boot menu. Select the USB key from the boot menu.
9. Wait for the Intel NUC to finish flashing and shutdown. Please wait until all LEDs are off.
10. Remove the USB key from the Intel NUC.
11. Power up the Intel NUC to boot the device.
12. Your device should appear in your fleet in the dashboard within a few minutes. Have fun!

For more details please refer to our Getting Started Guide.

Image displaying the field for selecting the version in the “add new device” panel of balenaEtcher.

● **Step 4:** After powering up the device it'll appear on your balenaCloud dashboard. All the services are automatically downloaded and launched by balena.



wandering-zombie

STATUS: Online	UUID: dce23ce	TYPE: Intel NUC
ONLINE FOR: 3 minutes	HOST OS VERSION: balenaOS 2.115.0	OS VARIANT: production
SUPERVISOR VERSION: 14.11.1	CURRENT RELEASE: 1.21.4+rev2	TARGET RELEASE: 1.21.4+rev2
FLEET: n3uron/iiot-gateway	SUPPORT ACCESS: Off	IS ACTIVE: ✓
LOCAL IP ADDRESS: 192.168.1.53	PUBLIC IP ADDRESS: 88.6.82.229	MAC ADDRESS: 08:00:27:10:84:15
TAGS (0): No tags configured yet		PUBLIC DEVICE URL: [✓ Off] On

NOTES: Add device notes

CPU ~85%
Memory 997 MB/1.9 GB
Temperature
Storage (/dev/sda6)

```
No logs yet
```

Service	Status	Release
grafana	Running	1.21.4+rev2
influxdb	Running	1.21.4+rev2
mqtt	Running	1.21.4+rev2
n3uron	Running	1.21.4+rev2

Image displaying Wandering Zombie Panel of balena Cloud.

Deploy from the Github repository (For advanced users)

If you are a balena CLI expert, feel free to use balena CLI. This option lets you configure in detail some aspects, like adding new services to your deploy or configure the existing but requires that you have both Git and the balena CLI installed on your system.

To do this, create an application in your balenaCloud dashboard, clone our repository, customize it to your needs and use the **balena push** command to deploy the stack to your fleet of devices.

```
git clone https://github.com/n3uron/ming-balena
```

```
cd ming-balena
```

```
balena login && balena push <fleet-name>
```

Configure N3uron

● **Step 1:** Open your web browser and go to http://<Local-IP>:8003 (in our case http://192.168.1.53:8003/) and login to N3uron's WebUI.

— **User:** admin
— **Password:** n3uron

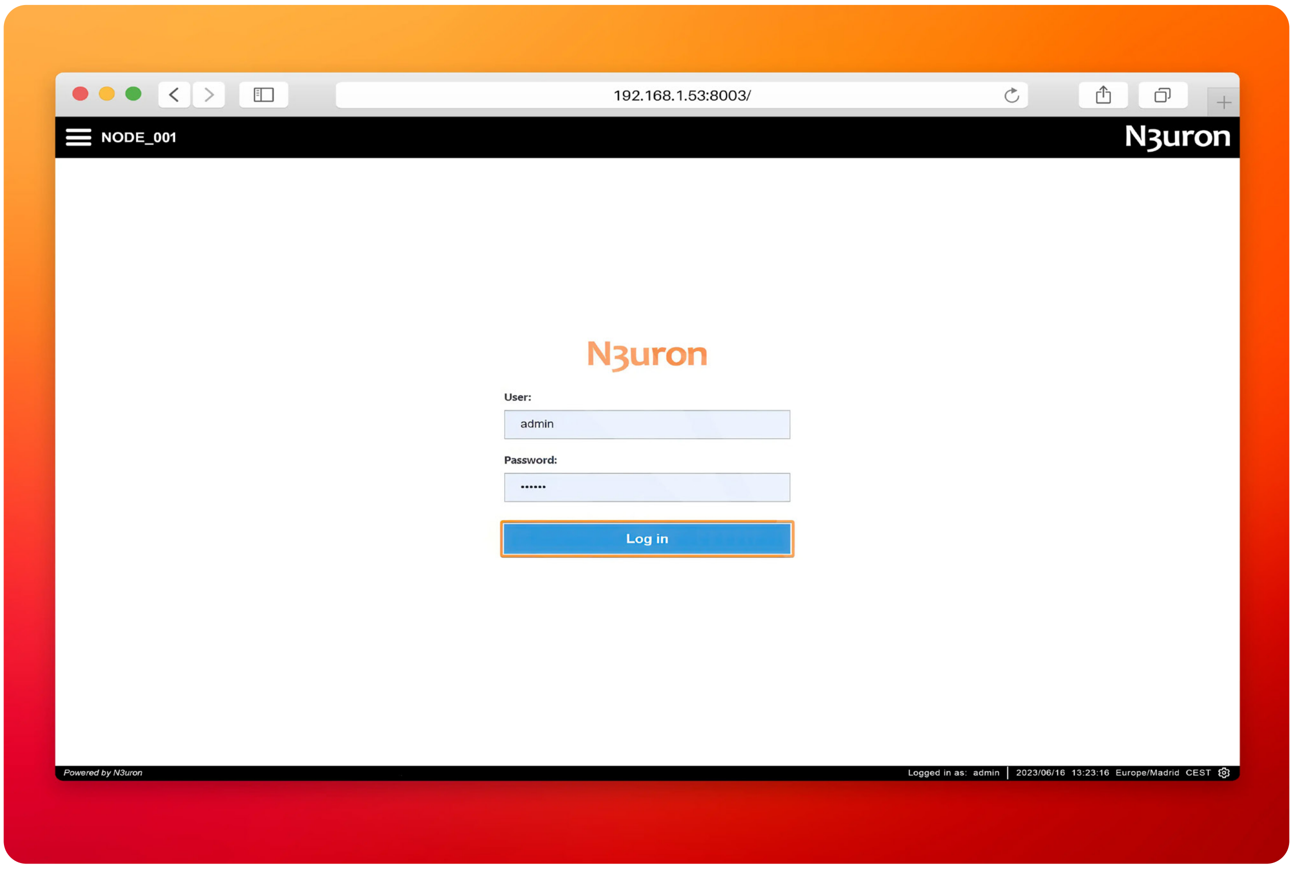
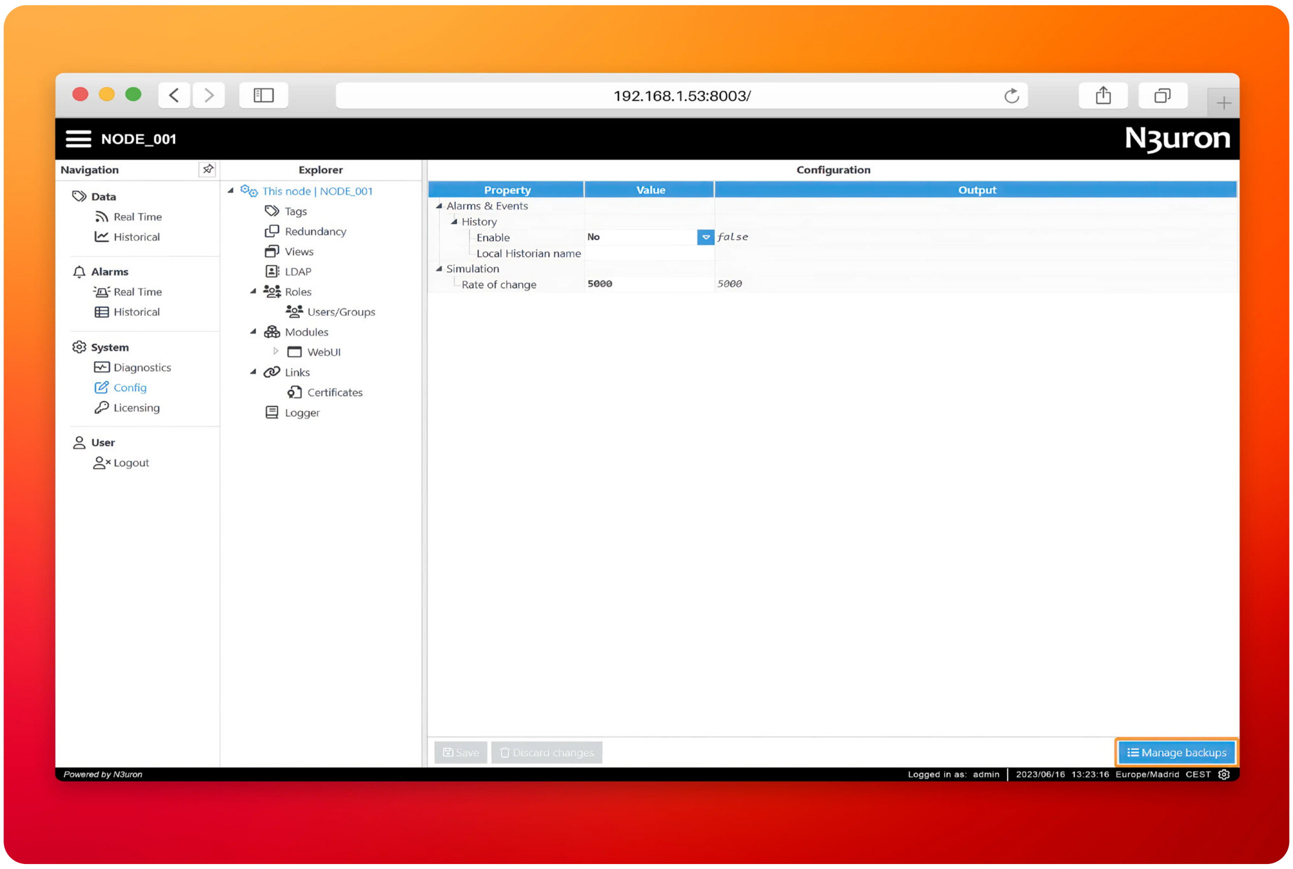


Image illustrating the login process to the N3uron web UI interface.

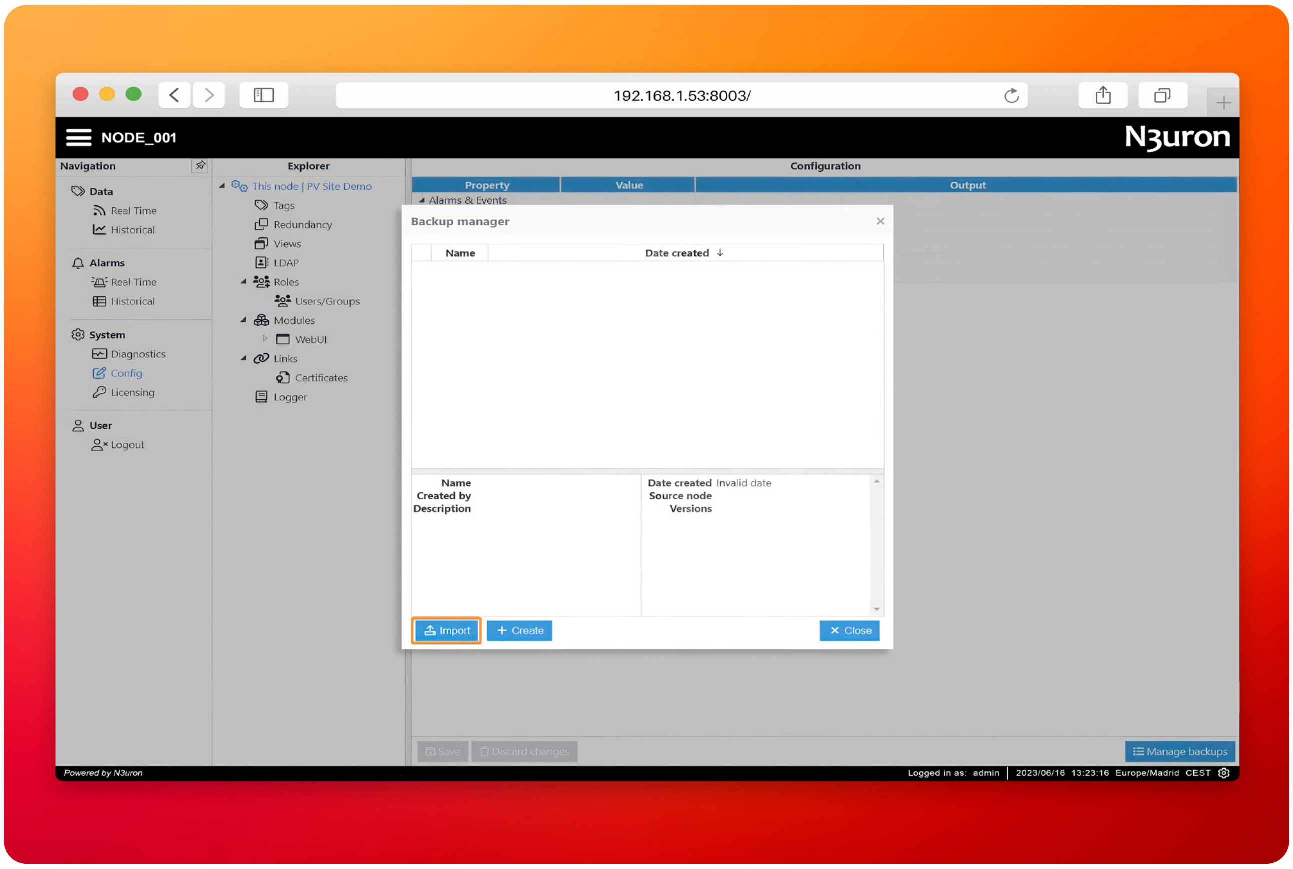
● **Step 2:** Navigate to the Config menu and click on **Manage backups**.



Property	Value	Output
Alarms & Events		
History		
Enable	No	false
Local Historian name		
Simulation		
Rate of change	5000	5000

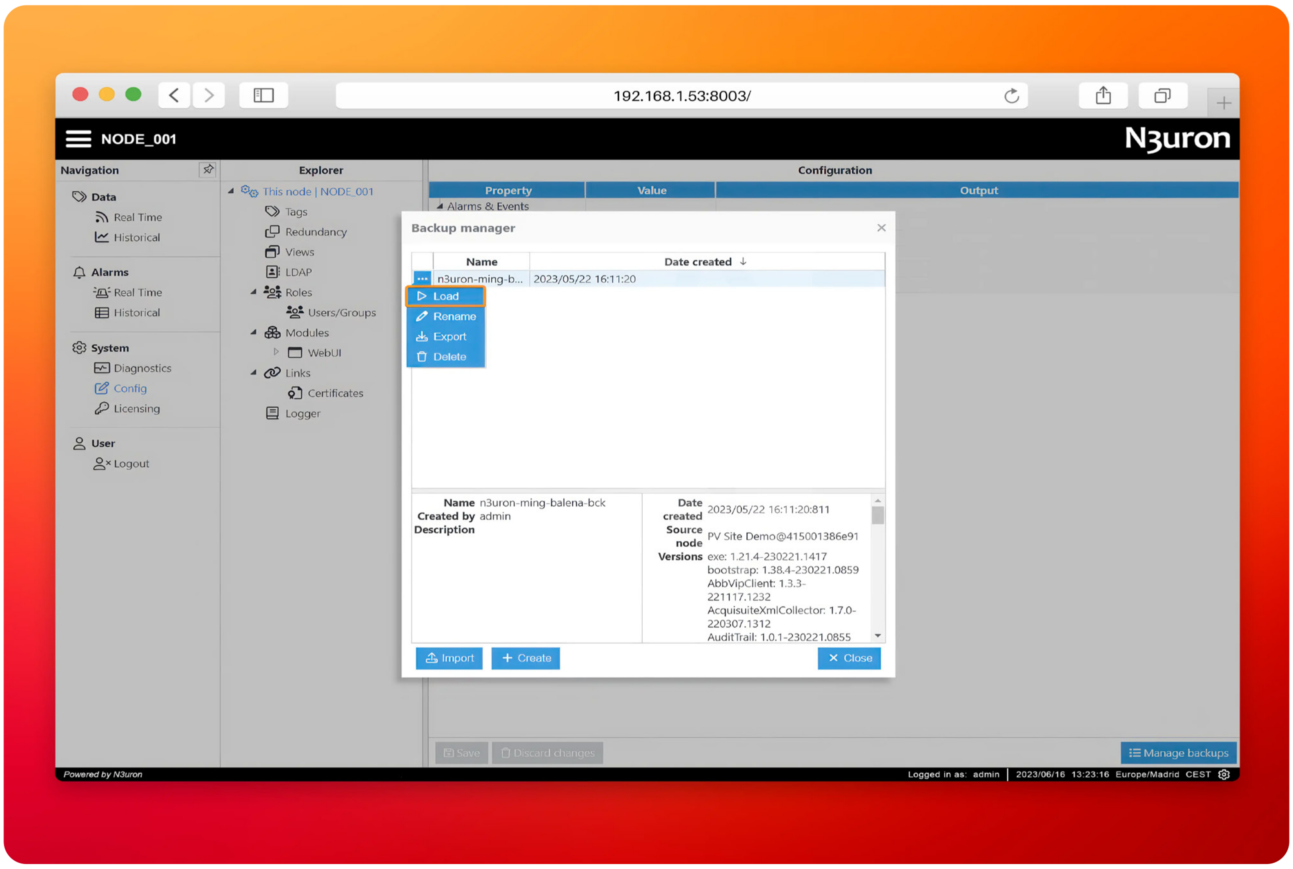
Image showcasing the management of backups through the N3uron WebUI interface for efficient backup operations.

- **Step 3:** Download the N3uron backup file for this demo, click on **Import** and select the file.



The image displays the Backup Manager window of N3uron, illustrating the required steps for users to import their backup files.

● **Step 4:** Next, load the backup. When prompted to restart, select **Close** and manually restart the service from the balenaCloud dashboard.



Backup manager

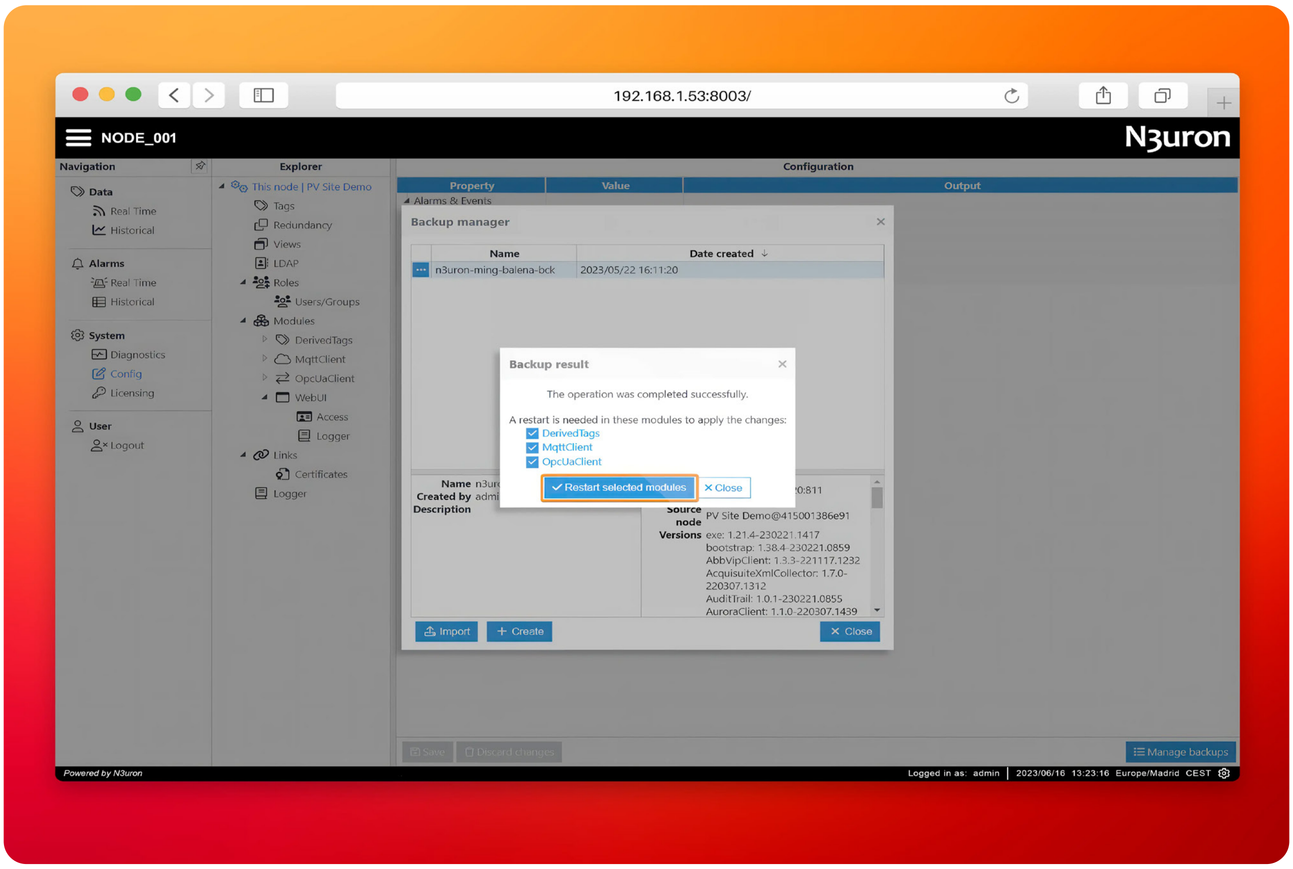
Name	Date created
n3uron-ming-b...	2023/05/22 16:11:20

Name n3uron-ming-balena-bck
Created by admin
Description

Date created 2023/05/22 16:11:20:811
Source node PV Site Demo@415001386e91
Versions exe: 1.21.4-230221.1417
bootstrap: 1.38.4-230221.0859
AbbVipClient: 1.3.3-221117.1232
AcquisuiteXmlCollector: 1.7.0-220307.1312
AuditTrail: 1.0.1-230221.0855

Logged in as: admin | 2023/06/16 13:23:16 Europe/Madrid CEST

This image displays the process of loading a N3uron backup file from the Backup Manager panel in the WebUI interface.



Backup manager

Name	Date created
n3uron-ming-balena-bck	2023/05/22 16:11:20

Backup result

The operation was completed successfully.

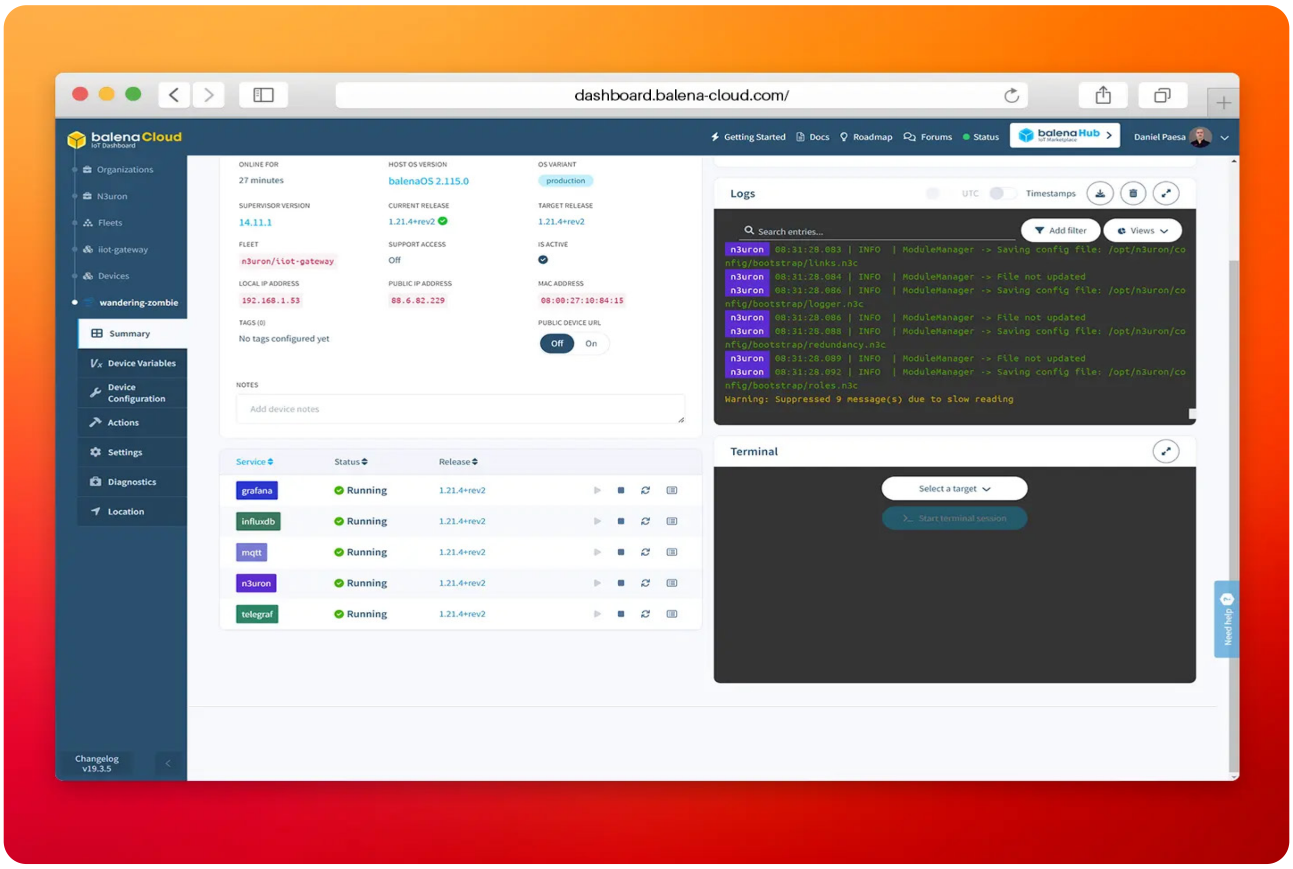
A restart is needed in these modules to apply the changes:
- [✓] DerivedTags
- [✓] MqttClient
- [✓] OpcUaClient

Restart selected modules | Close

Source node PV Site Demo@415001386e91
Versions exe: 1.21.4-230221.1417
bootstrap: 1.38.4-230221.0859
AbbVipClient: 1.3.3-221117.1232
AcquisuiteXmlCollector: 1.7.0-220307.1312
AuditTrail: 1.0.1-230221.0855
AuroraClient: 1.1.0-220307.1439

Logged in as: admin | 2023/06/16 13:23:16 Europe/Madrid CEST

The image highlights the step of restarting the N3uron service within the WebUI interface Using the Backup Result pop-up.



ONLINE FOR: 27 minutes	HOST OS VERSION: balenaOS 2.115.0	OS VARIANT: production
SUPERVISOR VERSION: 14.11.1	CURRENT RELEASE: 1.21.4+rev2	TARGET RELEASE: 1.21.4+rev2
FLEET: n3uron/iiot-gateway	SUPPORT ACCESS: Off	IS ACTIVE: ✓
LOCAL IP ADDRESS: 192.168.1.53	PUBLIC IP ADDRESS: 88.6.82.229	MAC ADDRESS: 08:00:27:10:84:15
TAGS (0): No tags configured yet		PUBLIC DEVICE URL: [Off] On

NOTES
Add device notes

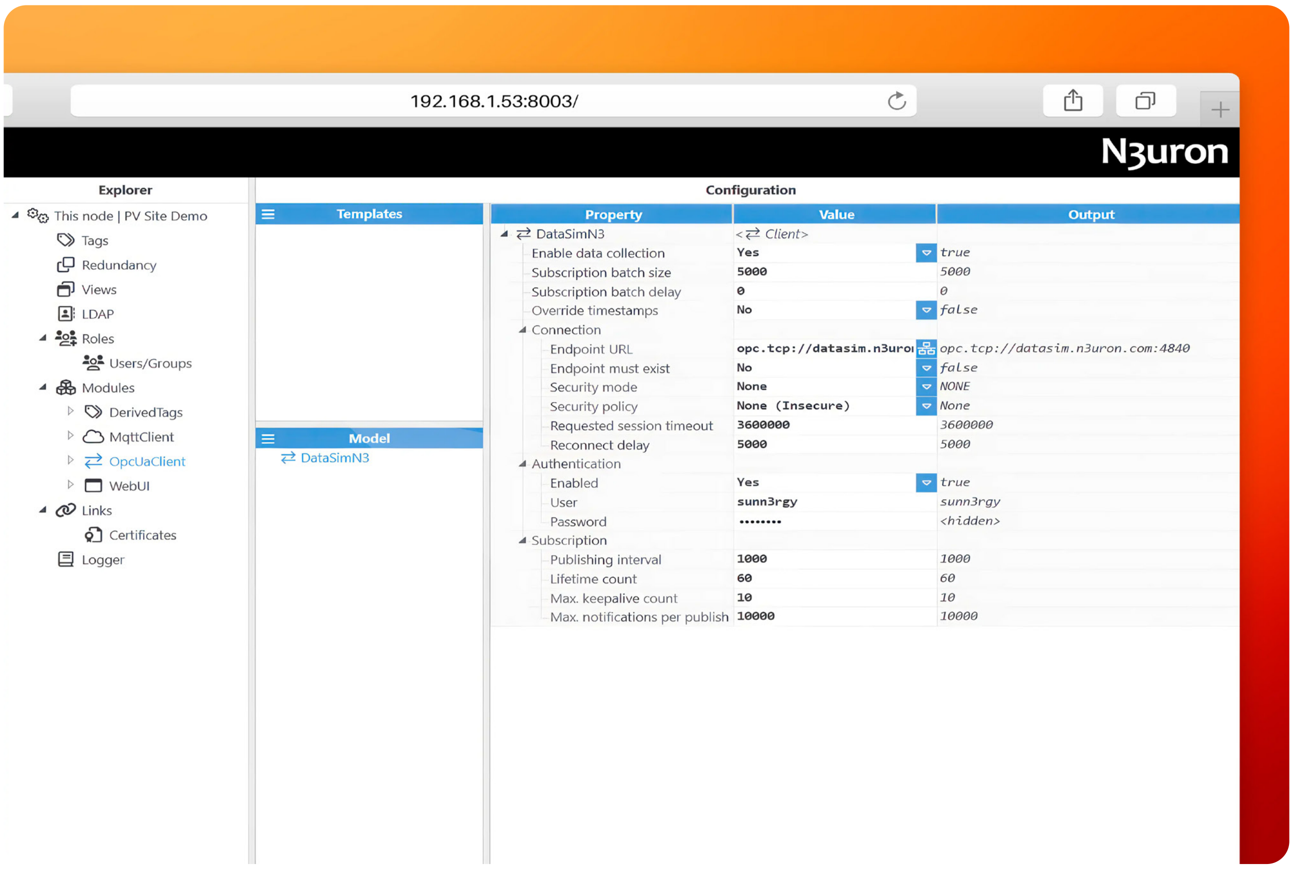
Service	Status	Release
grafana	Running	1.21.4+rev2
influxdb	Running	1.21.4+rev2
mqtt	Running	1.21.4+rev2
n3uron	Running	1.21.4+rev2
telegraf	Running	1.21.4+rev2

```
n3uron 08:31:28.083 | INFO  | ModuleManager -> Saving config file: /opt/n3uron/co
nfig/bootstrap/links.n3c
n3uron 08:31:28.084 | INFO  | ModuleManager -> File not updated
n3uron 08:31:28.086 | INFO  | ModuleManager -> Saving config file: /opt/n3uron/co
nfig/bootstrap/logger.n3c
n3uron 08:31:28.086 | INFO  | ModuleManager -> File not updated
n3uron 08:31:28.088 | INFO  | ModuleManager -> Saving config file: /opt/n3uron/co
nfig/bootstrap/redundancy.n3c
n3uron 08:31:28.089 | INFO  | ModuleManager -> File not updated
n3uron 08:31:28.092 | INFO  | ModuleManager -> Saving config file: /opt/n3uron/co
nfig/bootstrap/roles.n3c
Warning: Suppressed 9 message(s) due to slow reading
```

Screenshot showing the restarted balenaCloud Dashboard page.

After restoring the node backup you will get a fully configured N3uron instance. Take some time to review the node and its configuration.

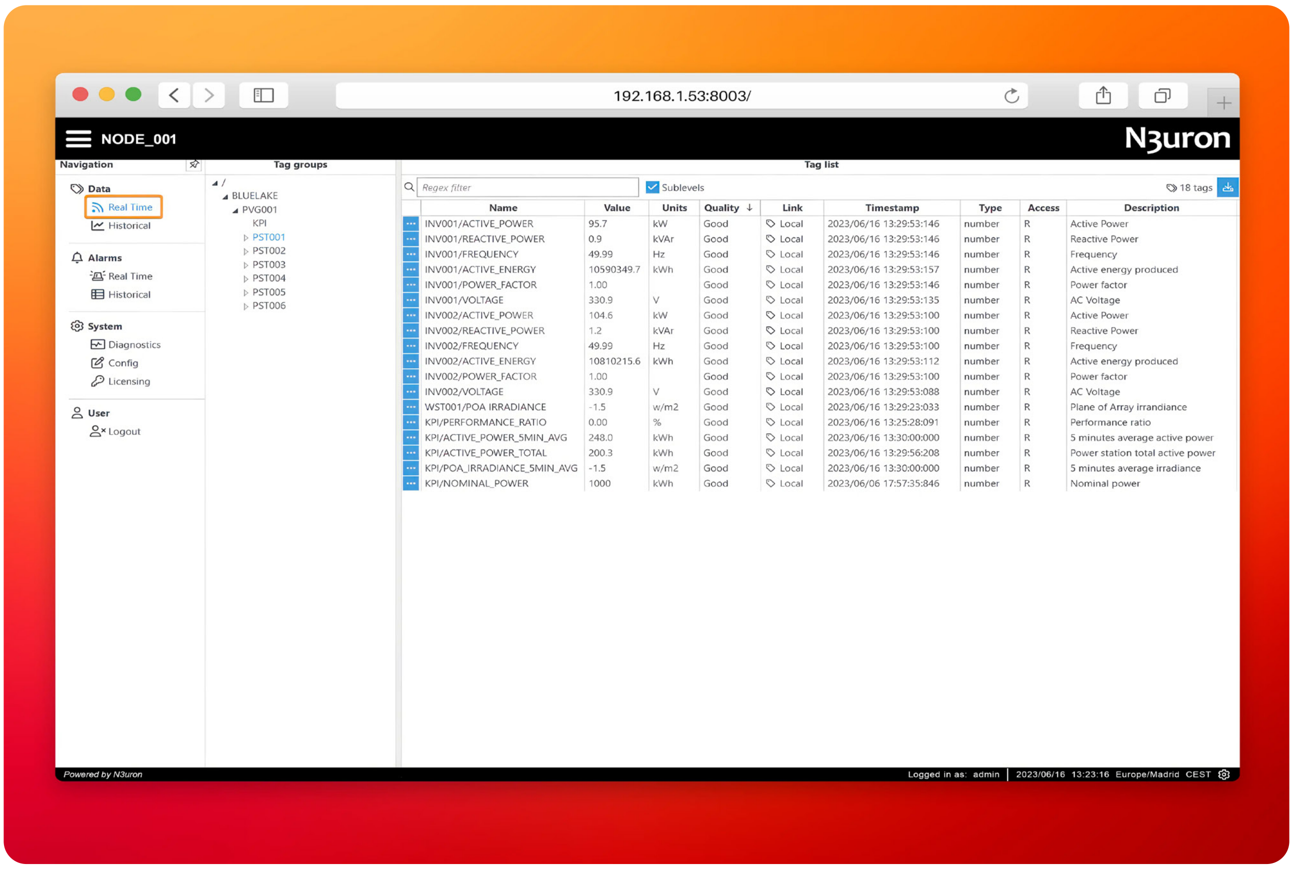
For example, you may see an OPC UA Client connected to datasim.n3uron.com:4840, which is receiving real-time data from a PV Plant located in Europe.



Property	Value	Output
DataSimN3	<Client>	
Enable data collection	Yes	true
Subscription batch size	5000	5000
Subscription batch delay	0	0
Override timestamps	No	false
Connection		
Endpoint URL	opc.tcp://datasim.n3uro	opc.tcp://datasim.n3uron.com:4840
Endpoint must exist	No	false
Security mode	None	NONE
Security policy	None (Insecure)	None
Requested session timeout	3600000	3600000
Reconnect delay	5000	5000
Authentication		
Enabled	Yes	true
User	sunn3rgy	sunn3rgy
Password	••••••••	<hidden>
Subscription		
Publishing interval	1000	1000
Lifetime count	60	60
Max. keepalive count	10	10
Max. notifications per publish	10000	10000

The image showcases the configuration panel of the OPC UA Client within the N3uron Web UI interface.

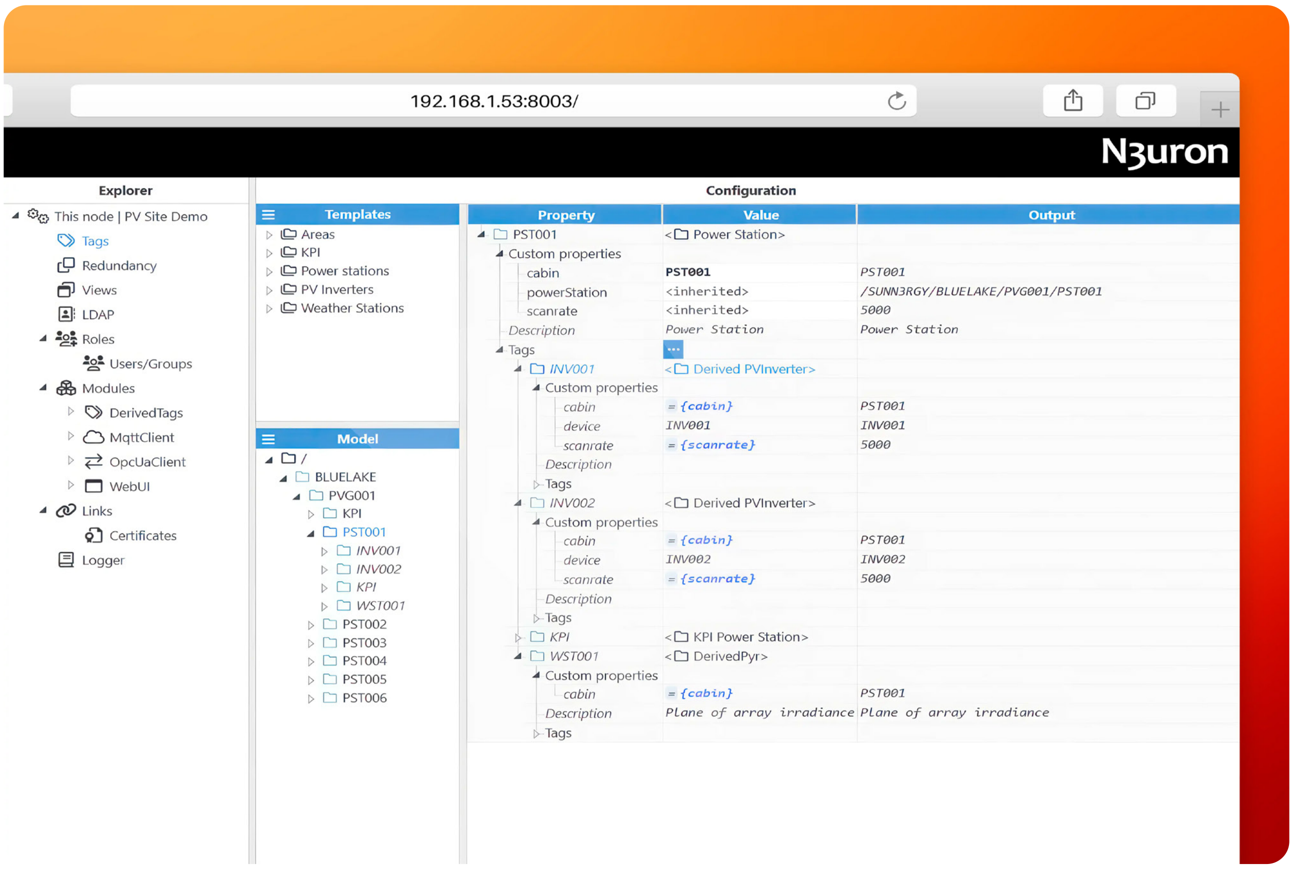
To visualize all the tags and their real time values, go to the **Data→Real Time**. In the example shown below, the Power Station 1 (PST001) is selected and the **Sublevels** checkbox is enabled.



Name	Value	Units	Quality ↓	Link	Timestamp	Type	Access	Description
INV001/ACTIVE_POWER	95.7	kW	Good	Local	2023/06/16 13:29:53:146	number	R	Active Power
INV001/REACTIVE_POWER	0.9	kVAr	Good	Local	2023/06/16 13:29:53:146	number	R	Reactive Power
INV001/FREQUENCY	49.99	Hz	Good	Local	2023/06/16 13:29:53:146	number	R	Frequency
INV001/ACTIVE_ENERGY	10590349.7	kWh	Good	Local	2023/06/16 13:29:53:157	number	R	Active energy produced
INV001/POWER_FACTOR	1.00		Good	Local	2023/06/16 13:29:53:146	number	R	Power factor
INV001/VOLTAGE	330.9	V	Good	Local	2023/06/16 13:29:53:135	number	R	AC Voltage
INV002/ACTIVE_POWER	104.6	kW	Good	Local	2023/06/16 13:29:53:100	number	R	Active Power
INV002/REACTIVE_POWER	1.2	kVAr	Good	Local	2023/06/16 13:29:53:100	number	R	Reactive Power
INV002/FREQUENCY	49.99	Hz	Good	Local	2023/06/16 13:29:53:100	number	R	Frequency
INV002/ACTIVE_ENERGY	10810215.6	kWh	Good	Local	2023/06/16 13:29:53:112	number	R	Active energy produced
INV002/POWER_FACTOR	1.00		Good	Local	2023/06/16 13:29:53:100	number	R	Power factor
INV002/VOLTAGE	330.9	V	Good	Local	2023/06/16 13:29:53:088	number	R	AC Voltage
WST001/POA IRRADIANCE	-1.5	w/m2	Good	Local	2023/06/16 13:29:23:033	number	R	Plane of Array irrandiance
KPI/PERFORMANCE_RATIO	0.00	%	Good	Local	2023/06/16 13:25:28:091	number	R	Performance ratio
KPI/ACTIVE_POWER_5MIN_AVG	248.0	kWh	Good	Local	2023/06/16 13:30:00:000	number	R	5 minutes average active power
KPI/ACTIVE_POWER_TOTAL	200.3	kWh	Good	Local	2023/06/16 13:29:56:208	number	R	Power station total active power
KPI/POA_IRRADIANCE_5MIN_AVG	-1.5	w/m2	Good	Local	2023/06/16 13:30:00:000	number	R	5 minutes average irradiance
KPI/NOMINAL_POWER	1000	kWh	Good	Local	2023/06/06 17:57:35:846	number	R	Nominal power

The image showcases the N3uron real-time panel view within the N3uron Web UI interface.

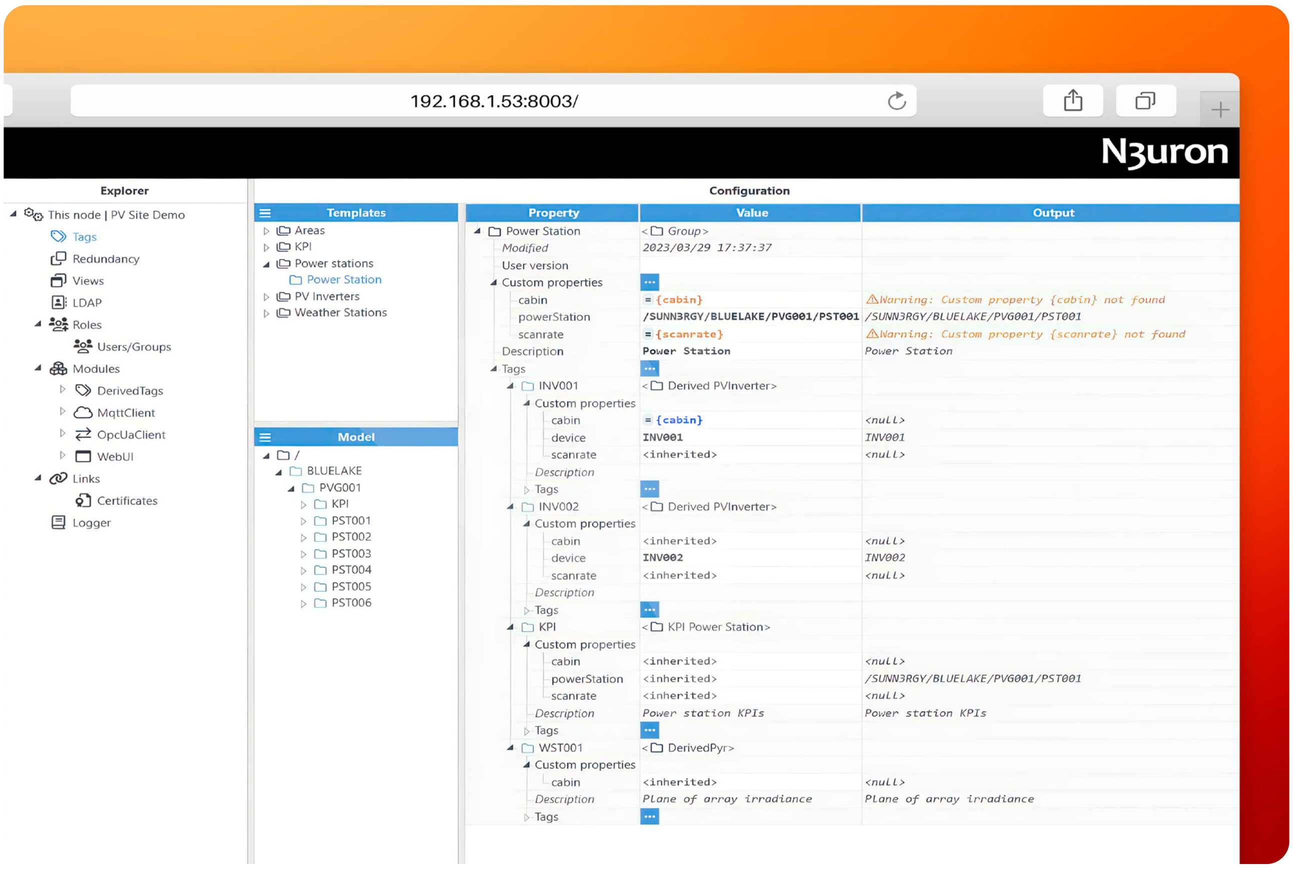
The data model used in this demo represents a Photovoltaic Plant with inverters and weather stations. By using templates you can quickly generate new instances to rapidly build complex data structures. Any changes made to a template definition will be inherited by all its instances.



Property	Value	Output
PST001	<Power Station>	
Custom properties		
cabin	**PST001**	PST001
powerStation	<inherited>	/SUNN3RGY/BLUELAKE/PVG001/PST001
scanrate	<inherited>	5000
Description	Power Station	Power Station
Tags	···	
INV001	<Derived PVInverter>	
Custom properties		
cabin	= {cabin}	PST001
device	INV001	INV001
scanrate	= {scanrate}	5000
Description		
Tags		
INV002	<Derived PVInverter>	
Custom properties		
cabin	= {cabin}	PST001
device	INV002	INV002
scanrate	= {scanrate}	5000
Description		
Tags		
KPI	<KPI Power Station>	
WST001	<DerivedPyr>	
Custom properties		
cabin	= {cabin}	PST001
Description	Plane of array irradiance	Plane of array irradiance
Tags		

The image showcases the Tags panel view in the N3uron Web UI interface, demonstrating the seamless process of generating templates for efficient data management.

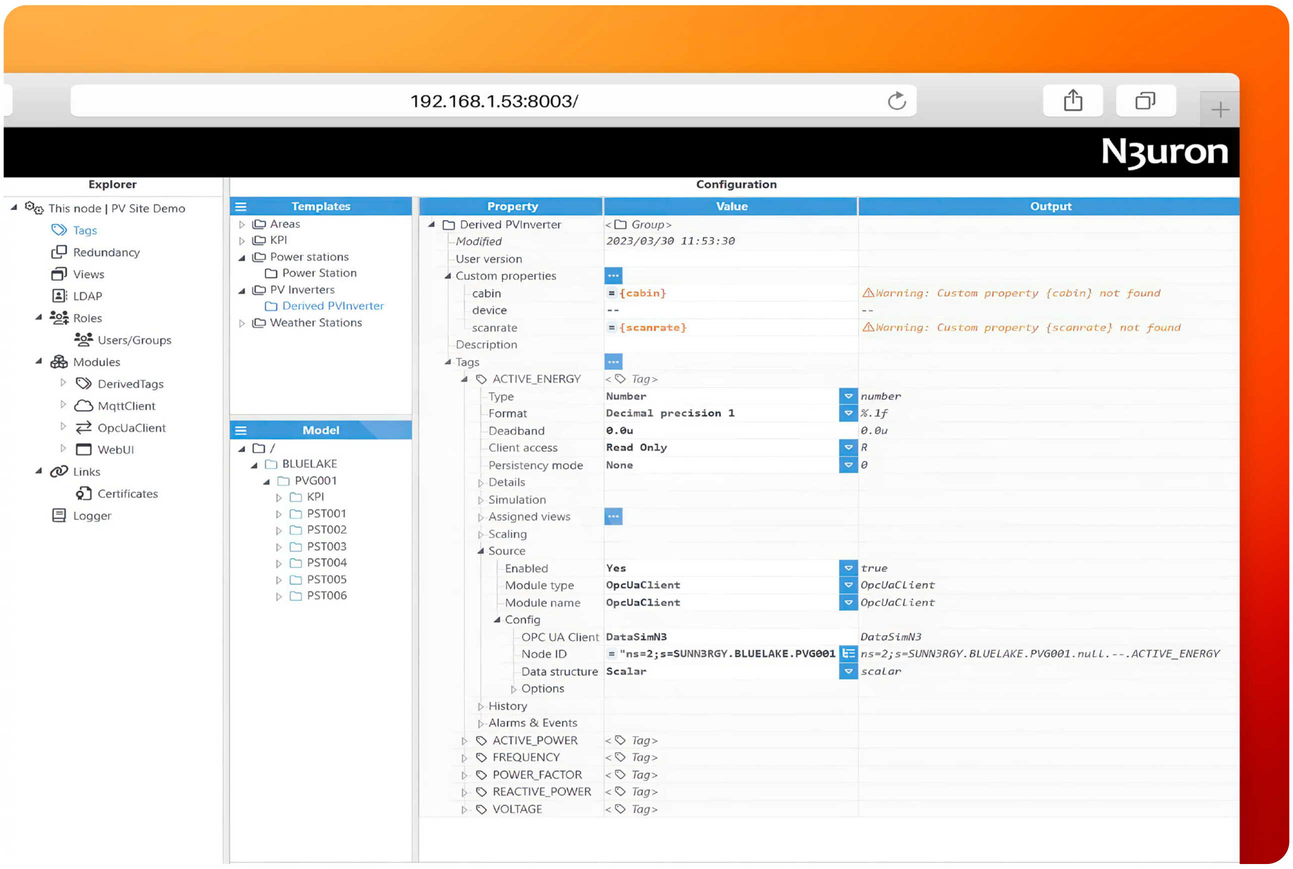
As shown below, the template definition for Power Stations includes two inverters (INV01 and INV02), a Weather Station (WST001), and a KPI folder with some calculated and aggregated data computed using the Derived Tags module. A template usually has several custom properties that act as variables and can be referenced by the items within the template.



Property	Value	Output
Power Station	< Group >	
Modified	2023/03/29 17:37:37	
User version		
Custom properties	…	
cabin	= {cabin}	⚠Warning: Custom property {cabin} not found
powerStation	/SUNN3RGY/BLUELAKE/PVG001/PST001	/SUNN3RGY/BLUELAKE/PVG001/PST001
scanrate	= {scanrate}	⚠Warning: Custom property {scanrate} not found
Description	Power Station	Power Station
Tags	…	
INV001	< Derived PVInverter>	
Custom properties		
cabin	= {cabin}	<null>
device	INV001	INV001
scanrate	<inherited>	<null>
Description		
Tags	…	
INV002	< Derived PVInverter>	
Custom properties		
cabin	<inherited>	<null>
device	INV002	INV002
scanrate	<inherited>	<null>
Description		
Tags	…	
KPI	< KPI Power Station>	
Custom properties		
cabin	<inherited>	<null>
powerStation	<inherited>	/SUNN3RGY/BLUELAKE/PVG001/PST001
scanrate	<inherited>	<null>
Description	Power station KPIs	Power station KPIs
Tags	…	
WST001	< DerivedPyr>	
Custom properties		
cabin	<inherited>	<null>
Description	Plane of array irradiance	Plane of array irradiance
Tags	…	

The image illustrates the Tags panel view in N3uron, highlighting the template structure for Power Stations.

The image below shows how to reference **Custom Properties** in expressions, when configuring tags.



192.168.1.53:8003/

N3uron

Explorer
- This node | PV Site Demo
  - Tags
  - Redundancy
  - Views
  - LDAP
  - Roles
    - Users/Groups
  - Modules
    - DerivedTags
    - MqttClient
    - OpcUaClient
    - WebUI
  - Links
    - Certificates
  - Logger

Templates
- Areas
- KPI
- Power stations
  - Power Station
- PV Inverters
  - Derived PVInverter
- Weather Stations

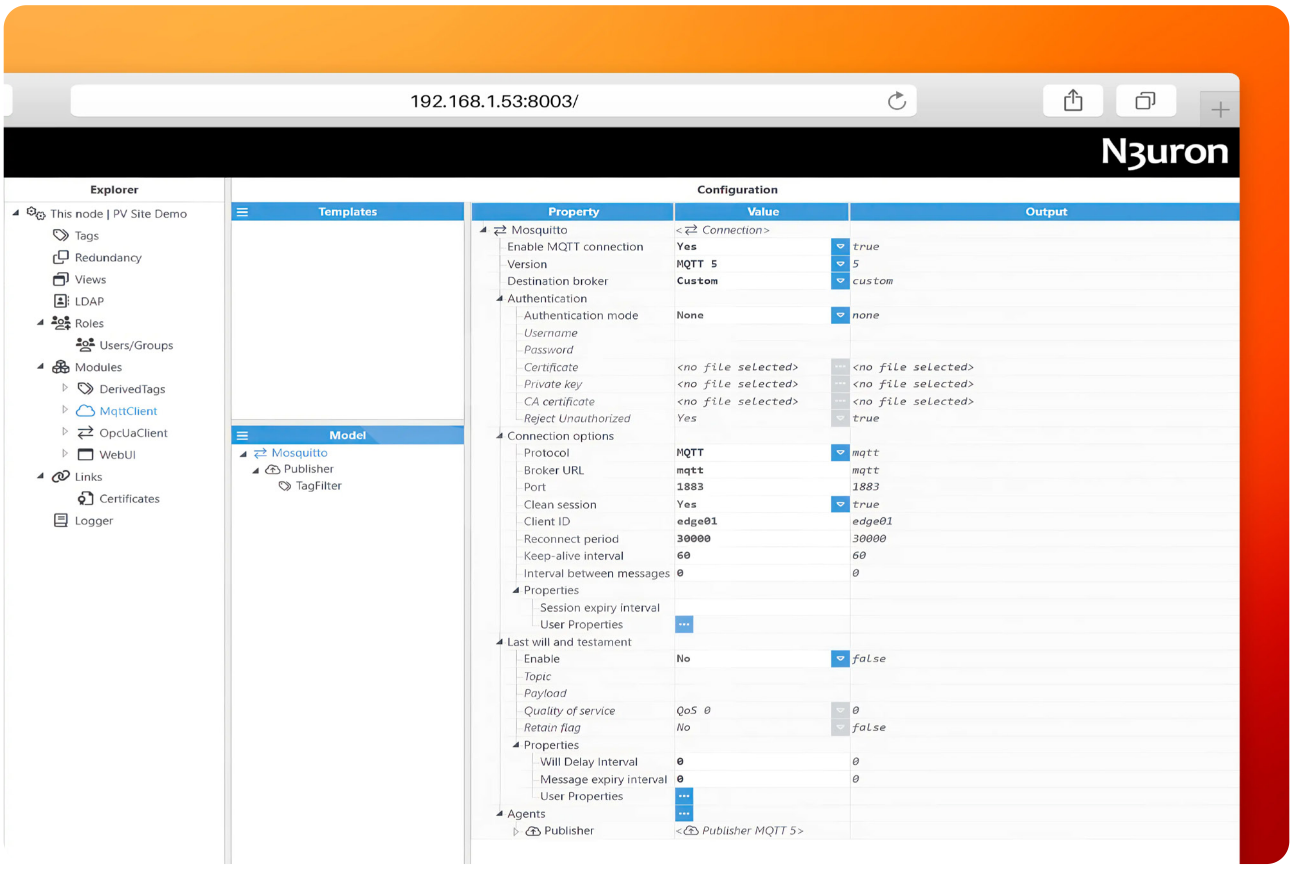
Model
- /
  - BLUELAKE
    - PVG001
      - KPI
      - PST001
      - PST002
      - PST003
      - PST004
      - PST005
      - PST006

Configuration

Property	Value	Output
Derived PVInverter	< Group>	
Modified	*2023/03/30 11:53:30*	
User version		
Custom properties	···	
cabin	= {cabin}	⚠Warning: Custom property {cabin} not found
device	--	--
scanrate	= {scanrate}	⚠Warning: Custom property {scanrate} not found
Description		
Tags	···	
ACTIVE_ENERGY	< Tag>	
Type	Number	number
Format	Decimal precision 1	%.1f
Deadband	0.0u	0.0u
Client access	Read Only	R
Persistency mode	None	0
Details		
Simulation		
Assigned views	···	
Scaling		
Source		
Enabled	Yes	true
Module type	OpcUaClient	OpcUaClient
Module name	OpcUaClient	OpcUaClient
Config		
OPC UA Client	DataSimN3	DataSimN3
Node ID	= "ns=2;s=SUNN3RGY.BLUELAKE.PVG001	ns=2;s=SUNN3RGY.BLUELAKE.PVG001.null.--.ACTIVE_ENERGY
Data structure	Scalar	scalar
Options		
History		
Alarms & Events		
ACTIVE_POWER	< Tag>	
FREQUENCY	< Tag>	
POWER_FACTOR	< Tag>	
REACTIVE_POWER	< Tag>	
VOLTAGE	< Tag>	

In the image, you can see the Tags panel view in N3uron. It showcases how to reference Custom Properties in expressions when setting up tags.

As you continue to explore the N3uron instance, you will also find that the node includes an MQTT Client that is already configured to push data to the local Mosquitto MQTT broker.

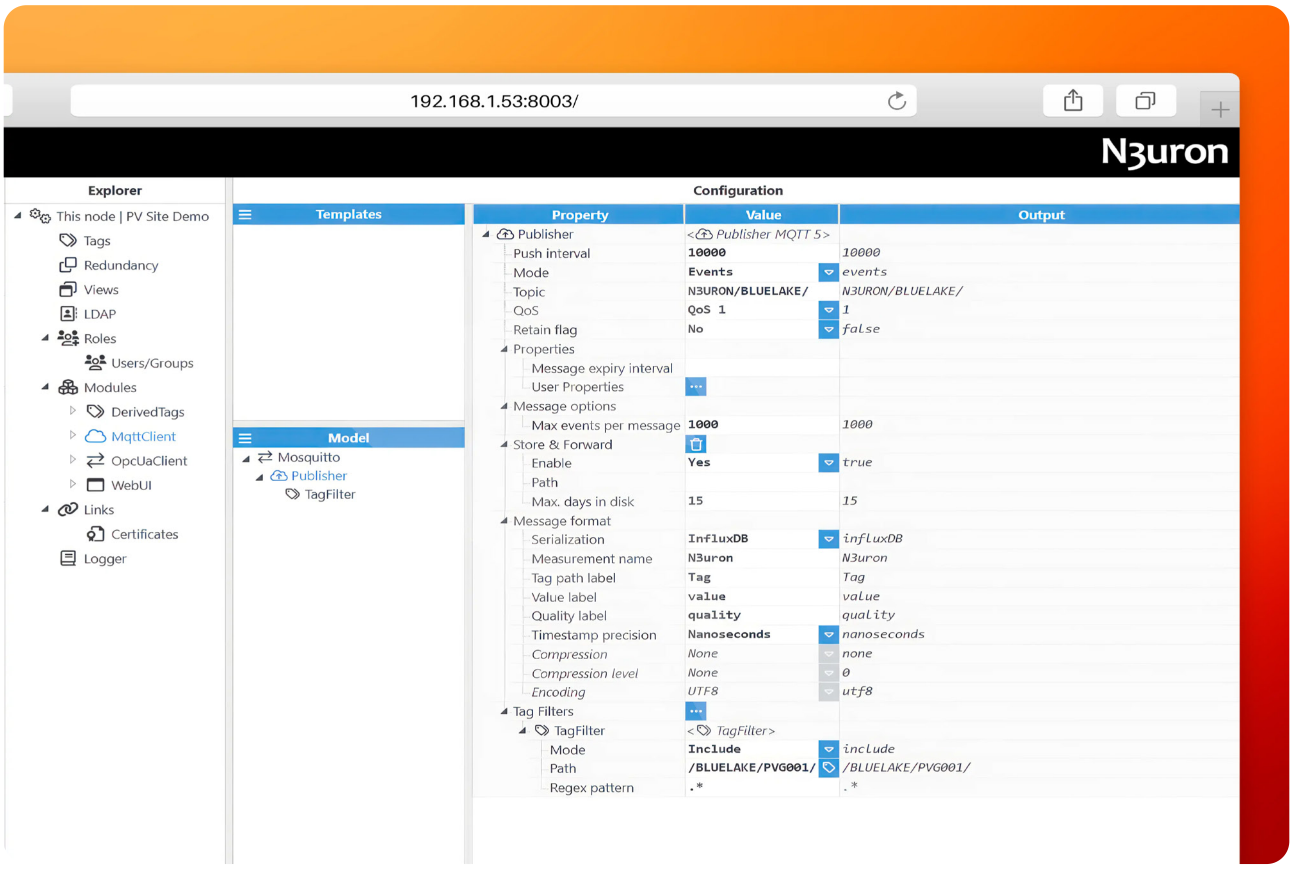


Property	Value	Output
Mosquitto	<Connection>	
Enable MQTT connection	Yes	true
Version	MQTT 5	5
Destination broker	Custom	custom
Authentication		
Authentication mode	None	none
Username		
Password		
Certificate	*<no file selected>*	*<no file selected>*
Private key	*<no file selected>*	*<no file selected>*
CA certificate	*<no file selected>*	*<no file selected>*
Reject Unauthorized	*Yes*	*true*
Connection options		
Protocol	MQTT	mqtt
Broker URL	mqtt	mqtt
Port	1883	1883
Clean session	Yes	true
Client ID	edge01	edge01
Reconnect period	30000	30000
Keep-alive interval	60	60
Interval between messages	0	0
Properties		
Session expiry interval		
User Properties	···	
Last will and testament		
Enable	No	false
Topic		
Payload		
Quality of service	*QoS 0*	*0*
Retain flag	*No*	*false*
Properties		
Will Delay Interval	0	0
Message expiry interval	0	0
User Properties	···	
Agents	···	
Publisher	*<Publisher MQTT 5>*	

This picture displays how to set up the MQTT Client on N3uron, which allows for easy transmission of data to the local Mosquito MQTT broker.

A Publisher has been created to push all tags within the /BLUELAKE/PVG001/ directory to an MQTT topic named N3URON/BLUELAKE/.

The InfluxDB payload format has been selected in the configuration for seamless message serialization.



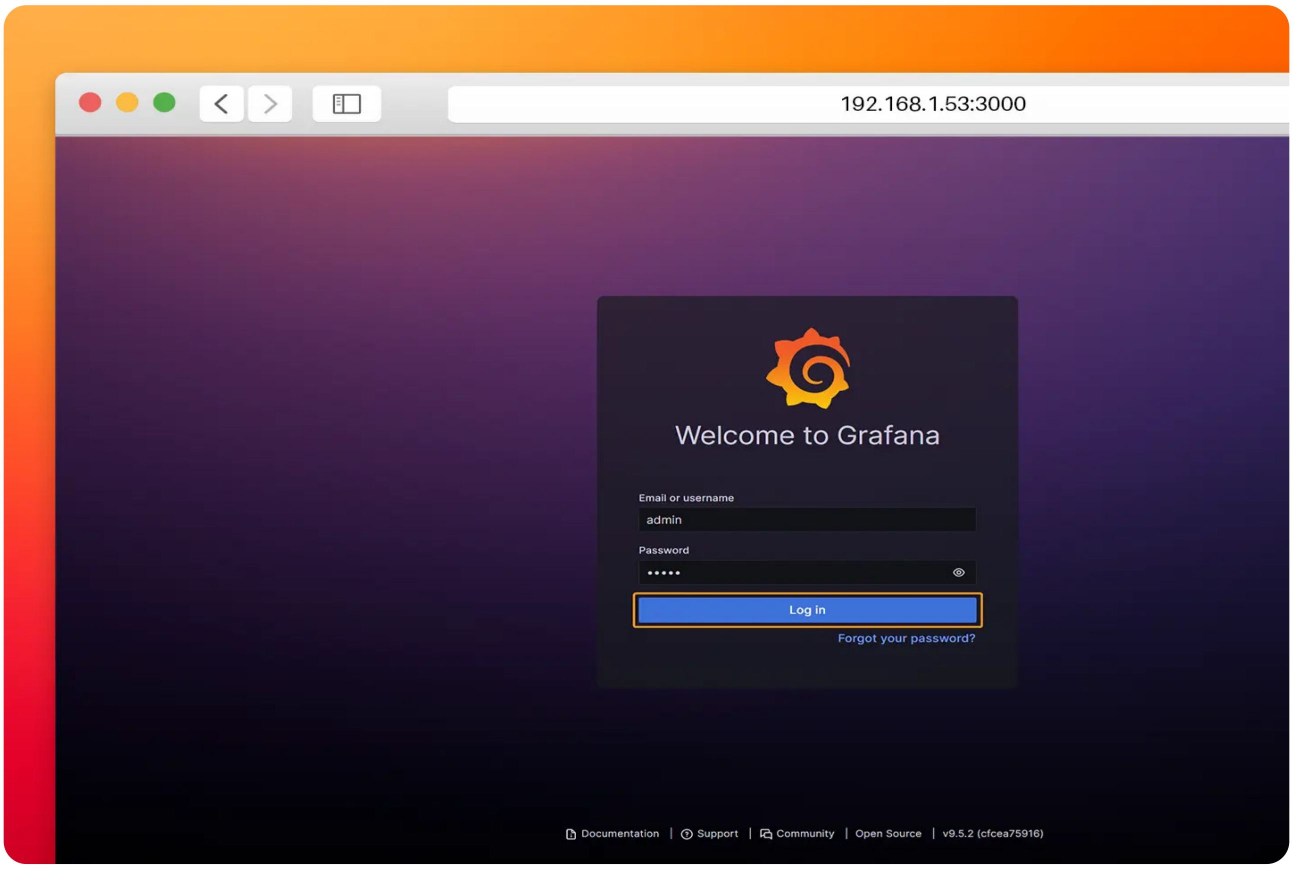
Property	Value	Output
Publisher	<Publisher MQTT 5>	
Push interval	10000	10000
Mode	Events	events
Topic	N3URON/BLUELAKE/	N3URON/BLUELAKE/
QoS	QoS 1	1
Retain flag	No	false
Properties		
Message expiry interval		
User Properties	···	
Message options		
Max events per message	1000	1000
Store & Forward		
Enable	Yes	true
Path		
Max. days in disk	15	15
Message format		
Serialization	InfluxDB	influxDB
Measurement name	N3uron	N3uron
Tag path label	Tag	Tag
Value label	value	value
Quality label	quality	quality
Timestamp precision	Nanoseconds	nanoseconds
Compression	*None*	none
Compression level	*None*	0
Encoding	*UTF8*	utf8
Tag Filters	···	
TagFilter	<TagFilter>	
Mode	Include	include
Path	/BLUELAKE/PVG001/	/BLUELAKE/PVG001/
Regex pattern	.*	.*

This image displays how to configure an MQTT Publisher in N3uron with the InfluxDB payload format.

Configure Grafana

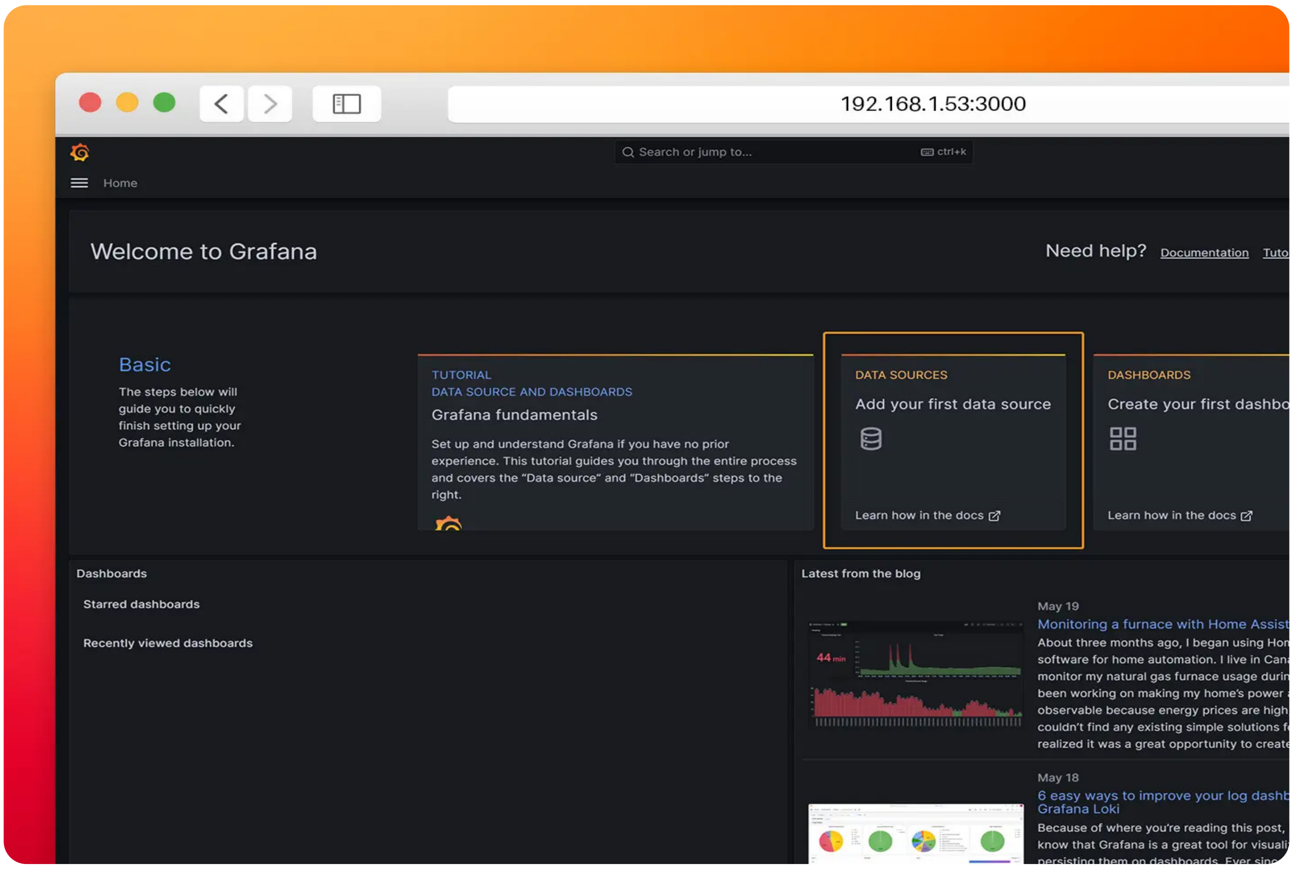
● **Step 1:** Go to http://<Local-IP>:3000/ and login to Grafana.

— **User:** admin
— **Password:** n3uron



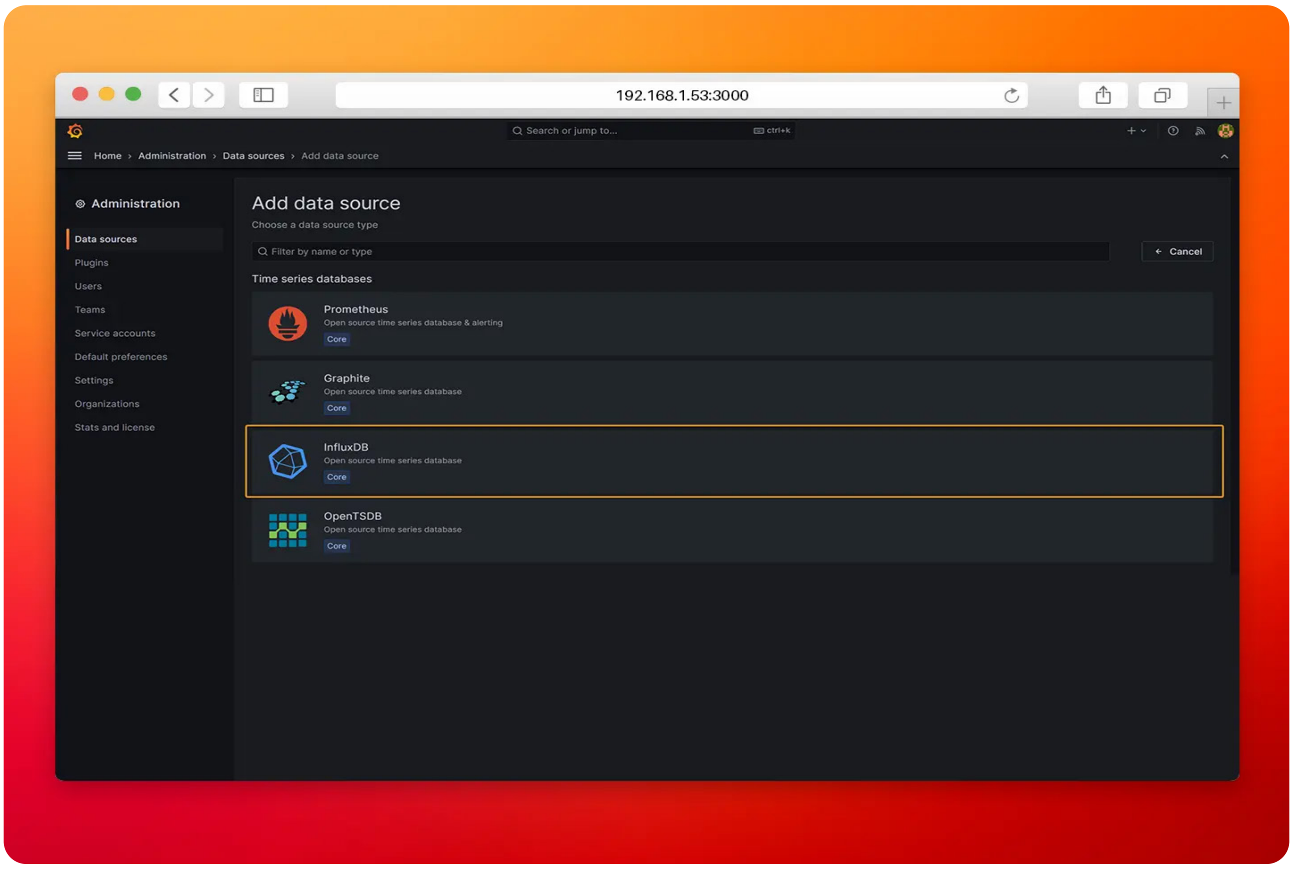
This image shows Grafana's logging page.

- **Step 2:** Click on **Add your first data source**.


192.168.1.53:3000
Search or jump to...
ctrl+k
Home
Welcome to Grafana
Need help?
Documentation
Basic
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.
DATA SOURCES
Add your first data source
Learn how in the docs
DASHBOARDS
Learn how in the docs
Dashboards
Starred dashboards
Recently viewed dashboards
Latest from the blog
May 19
May 18

The image illustrates the second step of the configuration process, where users are guided to click on "Add your first data source" in Grafana.

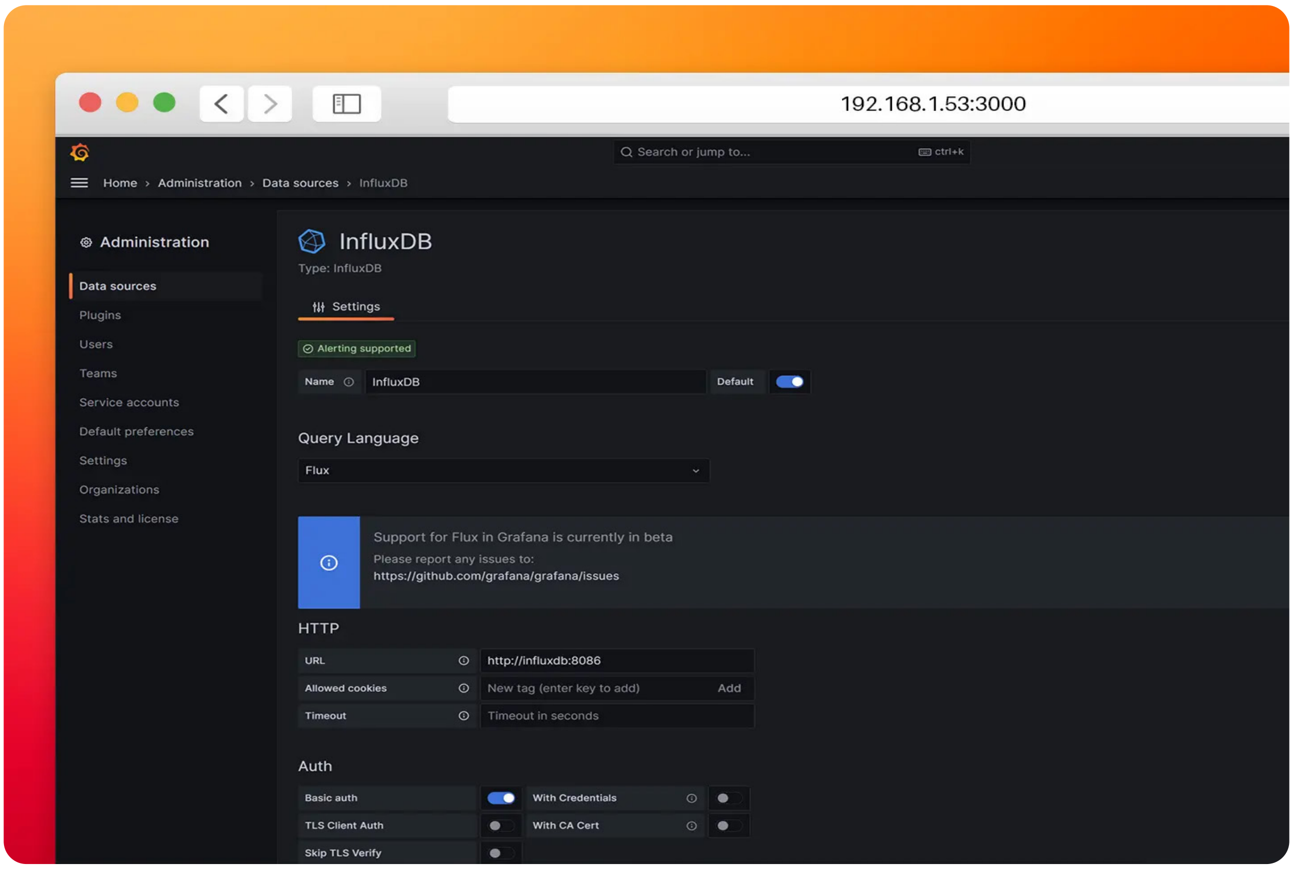
● **Step 3:** Select **InfluxDB** as data source.



The image represents the third step of the configuration process, where users are guided to select InfluxDB as the data source in Grafana.

● **Step 4:** Configure the InfluxDB connection as follows.

— **Query Language:** Flux
— **URL:** http://influxdb:8086
— **Organization:** N3uron
— **Token:** iXm3eMP5ioUUntSZuHYWzjJPT
— **Default Bucket:** demo



192.168.1.53:3000

Home › Administration › Data sources › InfluxDB

Administration

- Data sources
- Plugins
- Users
- Teams
- Service accounts
- Default preferences
- Settings
- Organizations
- Stats and license

InfluxDB

Type: InfluxDB

Settings

Alerting supported

Name: InfluxDB — Default: on

Query Language: Flux

Support for Flux in Grafana is currently in beta
Please report any issues to:
https://github.com/grafana/grafana/issues

HTTP

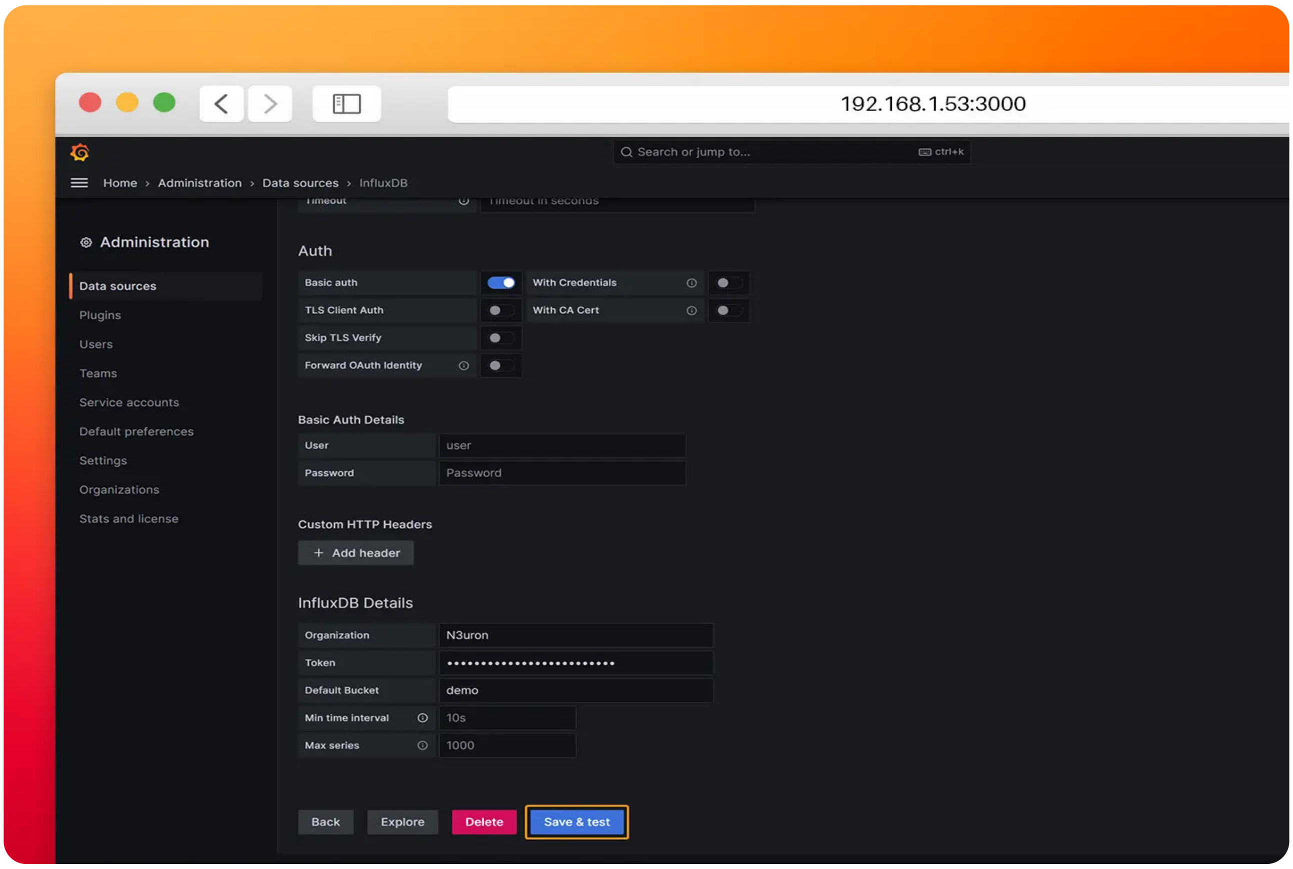
URL	http://influxdb:8086
Allowed cookies	New tag (enter key to add) — Add
Timeout	Timeout in seconds

Auth

Basic auth	on	With Credentials	off
TLS Client Auth	off	With CA Cert	off
Skip TLS Verify	off		

The image illustrates the fourth step of the configuration process, where users are guided to configure the InfluxDB connection in Grafana

● **Step 5:** Click on **Save & Test**. If you encounter any error with the data source review your settings.



192.168.1.53:3000

Home › Administration › Data sources › InfluxDB

Auth

Basic auth	on	With Credentials	off
TLS Client Auth	off	With CA Cert	off
Skip TLS Verify	off		
Forward OAuth Identity	off		

Basic Auth Details

User	user
Password	Password

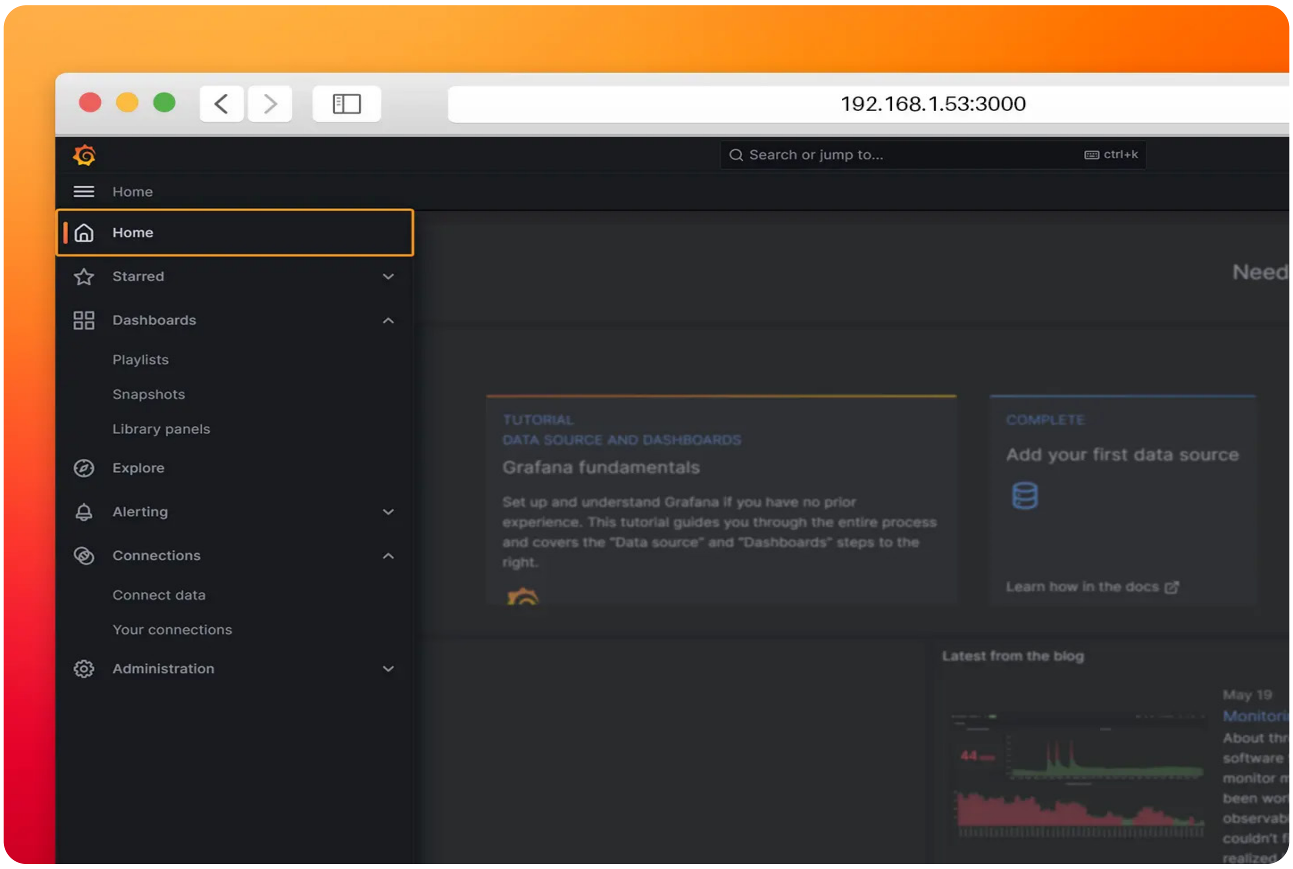
Custom HTTP Headers

InfluxDB Details

Organization	N3uron
Token	•••••••••••••••••••••••••
Default Bucket	demo
Min time interval	10s
Max series	1000

The image depicts the fifth step of the configuration process, where users are instructed to click on “Save & Test” in Grafana.

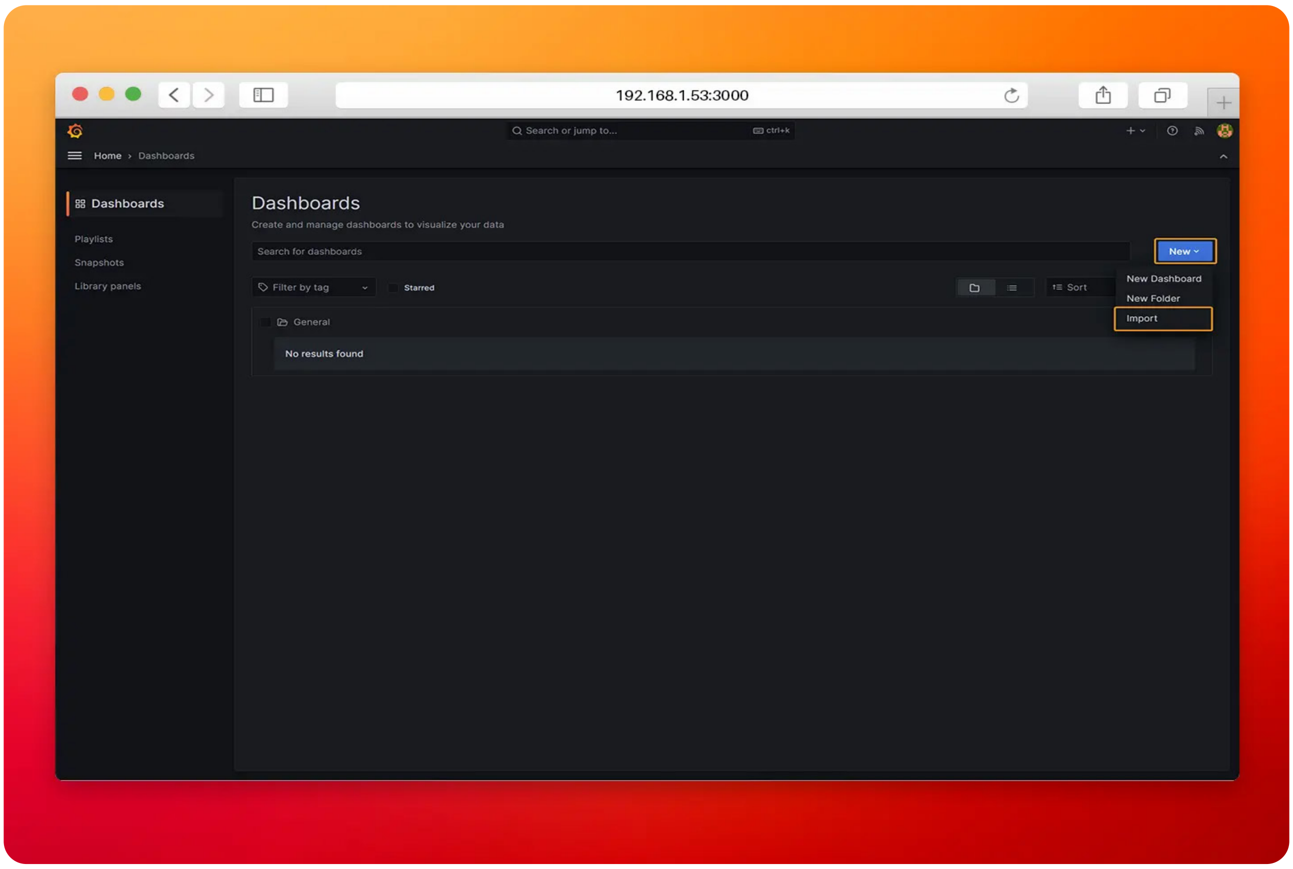
- **Step 6:** Hover the mouse over the Home menu and select **Dashboards**.



Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

The image showcases the sixth step of the navigation process, where users are instructed to hover the mouse over the Home menu in Grafana and select Dashboards.

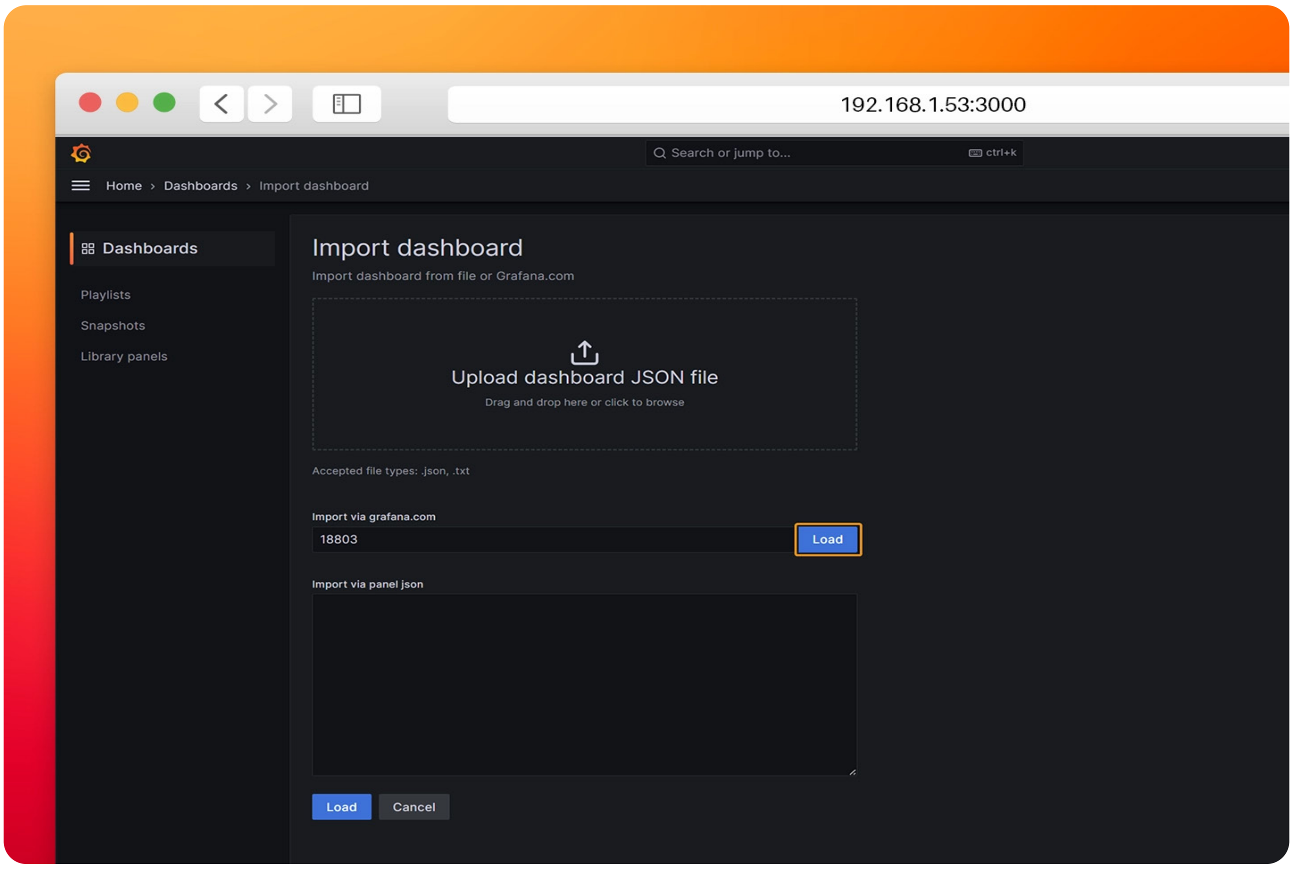
- **Step 7:** Click on **New→Import**.



The image represents the seventh step of the navigation process, where users are guided to click on "New" and select "Import" in Grafana.

● **Step 8:** Load the dashboard from Grafana.com.

— **ID:** 18803


192.168.1.53:3000
Search or jump to...
ctrl+k
Home › Dashboards › Import dashboard
Dashboards
Playlists
Snapshots
Library panels
Upload dashboard JSON file
Drag and drop here or click to browse
Load
Load
Cancel

Import dashboard

Import dashboard from file or Grafana.com

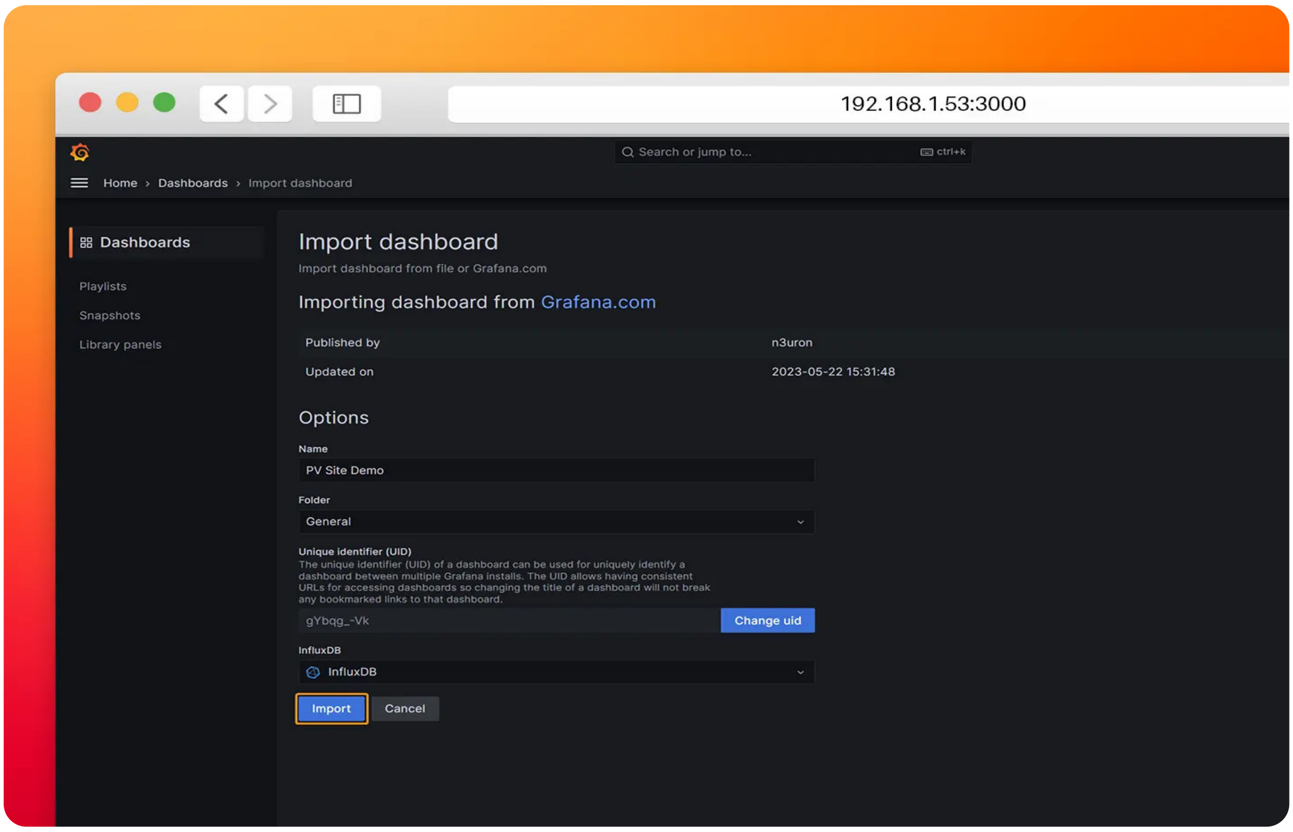
Accepted file types: .json, .txt

Import via grafana.com: 18803

Import via panel json

The image showcases the eighth step of the dashboard import process, where users are instructed to load a specific dashboard from Grafana.com.

- **Step 9:** Select the previously created InfluxDB data source and click on **Import**.


192.168.1.53:3000

Search or jump to... ctrl+k

Home › Dashboards › Import dashboard

Dashboards
Playlists
Snapshots
Library panels

Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from Grafana.com

Published by	n3uron
Updated on	2023-05-22 15:31:48

Options

Name: PV Site Demo

Folder: General

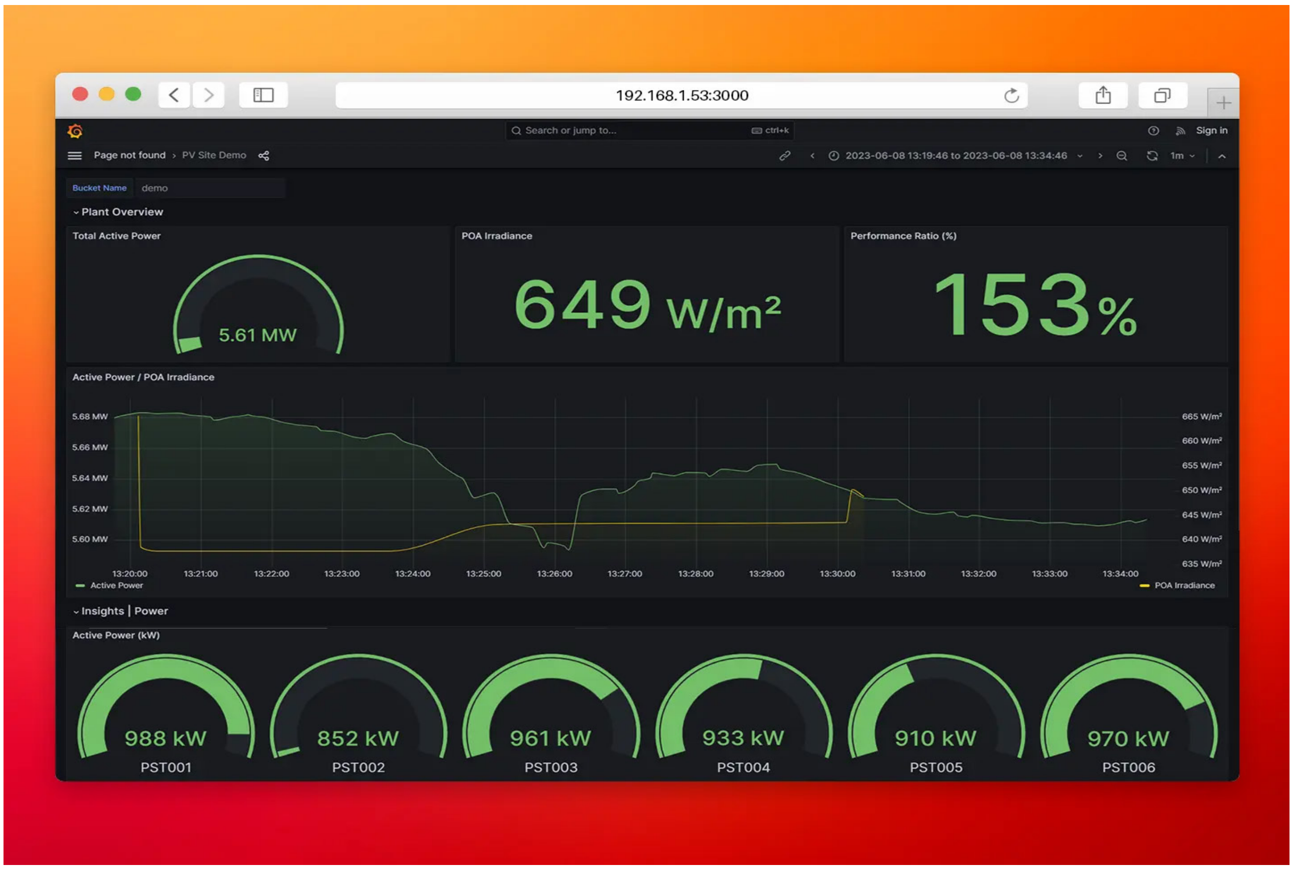
Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

gYbqg_-Vk — Change uid

InfluxDB: InfluxDB

Import — Cancel

The image represents the ninth step of the dashboard import process, where users are guided to select the previously created InfluxDB data source and click on Import in Grafana.



The image showcases the visual representation of a Grafana Dashboard after the successful import of the InfluxDB data source.

Conclusion

In summary, we have seen how the integration of balena with **N3uron** and other applications in a containerized architecture provides businesses with a powerful tool to harness the full potential of their data.

Deploying and managing this solution at the Industrial Edge has never been easier, thanks to balena's user-friendly features. With simple provisioning for new hardware, automatic OS updates, seamless software deployments and unified management from their dashboard, balena streamlines operational processes. With **N3uron** and balena, the possibilities are endless.

Optimize your operations, discover new insights and stay ahead of the curve with **N3uron**. Take the first step on your digital journey by downloading the containerized version of the **N3uron** platform from Downloads. Select the modules you need to build your custom solution and unlock the full potential of your data.