How to integrate your Industrial Data in InfluxDB Using MQTT and N3uron

InfluxDB and N3uron Overview

Industry 4.0 has brought with it an explosive growth in real-time data. This data needs to be stored in such a way that reflects the sequential order of events so that it can be quickly queried and analyzed, which is why time-series databases like InfluxDB are paramount to providing interoperability between IT and OT. InfluxDB is an open-source database that is optimized for fast and highly available data storage and retrieval for time series data in use cases like IIoT, real-time analytics, operations monitoring, and more.

With over forty modules, ranging from traditional industrial protocols such as Modbus, DNP3, OPC UA, or OPC DA to more conventional protocols like MQTT or REST, **N3uron** provides a turnkey connectivity solution for the Industrial Internet of Things (IIoT). Nevertheless, **N3uron** is not just a connectivity platform. It is mainly intended to build a Unified Namespace (UNS) enterprise by modeling, aggregating, standardizing, and contextualizing data on the plant floor from a multitude of sources, whether that be PLCs, manufacturing execution systems, SCADA, or ERP, just to name a few, and make this data available to other third-party applications running both on-premise or in the cloud.

This article will explain in detail how to seamlessly push data models created within a **N3uron** node into InfluxDB Cloud using its MQTT Native Collector, effectively turning InfluxDB into an MQTT client. As depicted in the image below, we will also need to deploy an MQTT broker. In this case, we will use a HiveMQ Cloud cluster.

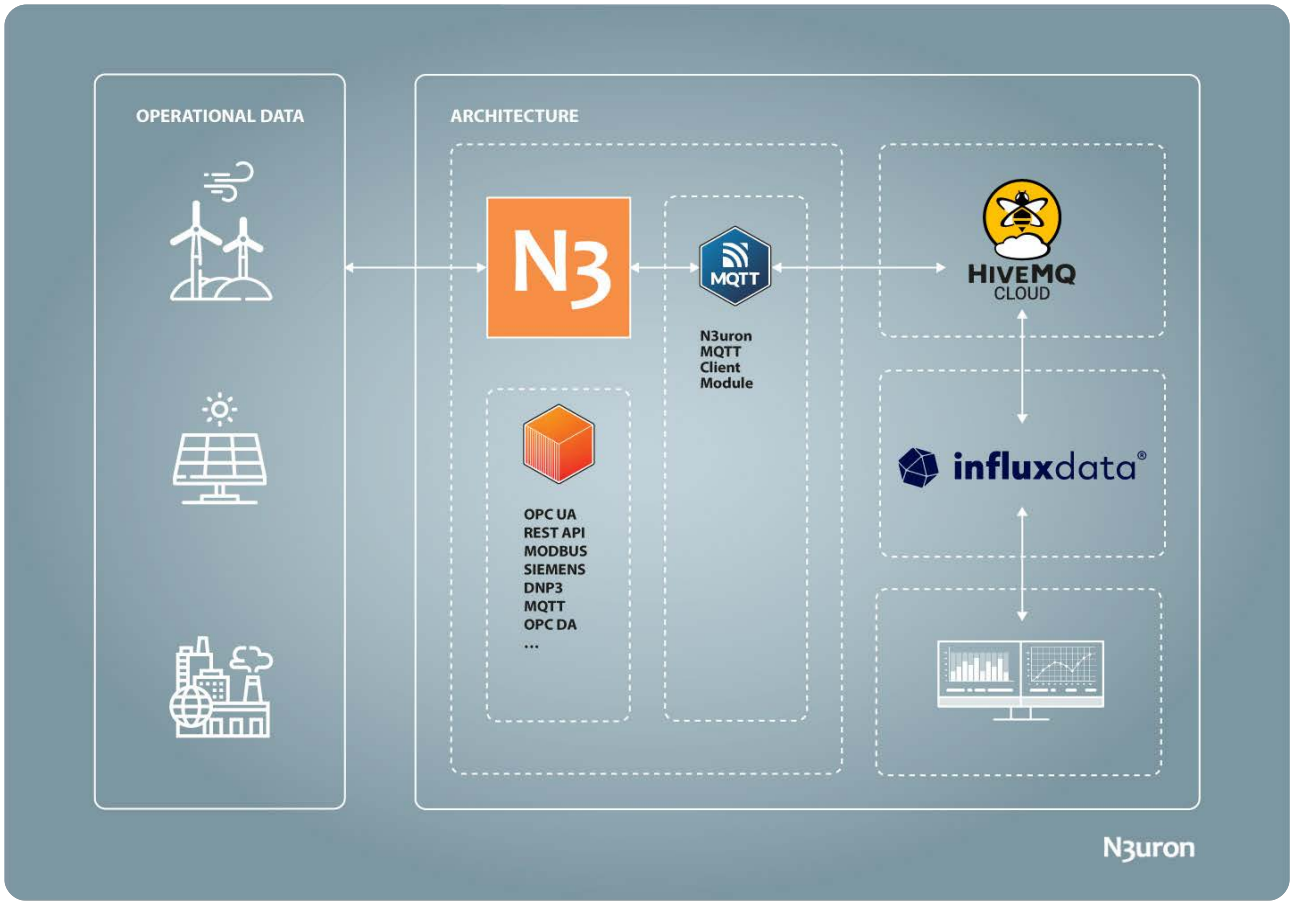


Diagram displaying how the MQTT Client module allows receiving and visualizing data into InfluxDB Cloud using HiveMQ as professional message broker software.

N3uron and InfluxDB Requirements

It is assumed that you already have an InfluxDB Cloud account. If not, you can create one at https://cloud2.influxdata.com/signup.

If you haven't downloaded **N3uron** yet, you can do so at https://n3uron.com/downloads/. If this is the first time you are installing N3uron, our Knowledge Base will guide you through the entire installation process.

HiveMQ Cloud MQTT Broker configuration

To configure the HiveMQ Cloud MQTT Broker, proceed as follows:

Step 1: Navigate to the HiveMQ website and access the "**Cloud**" tab at the top.

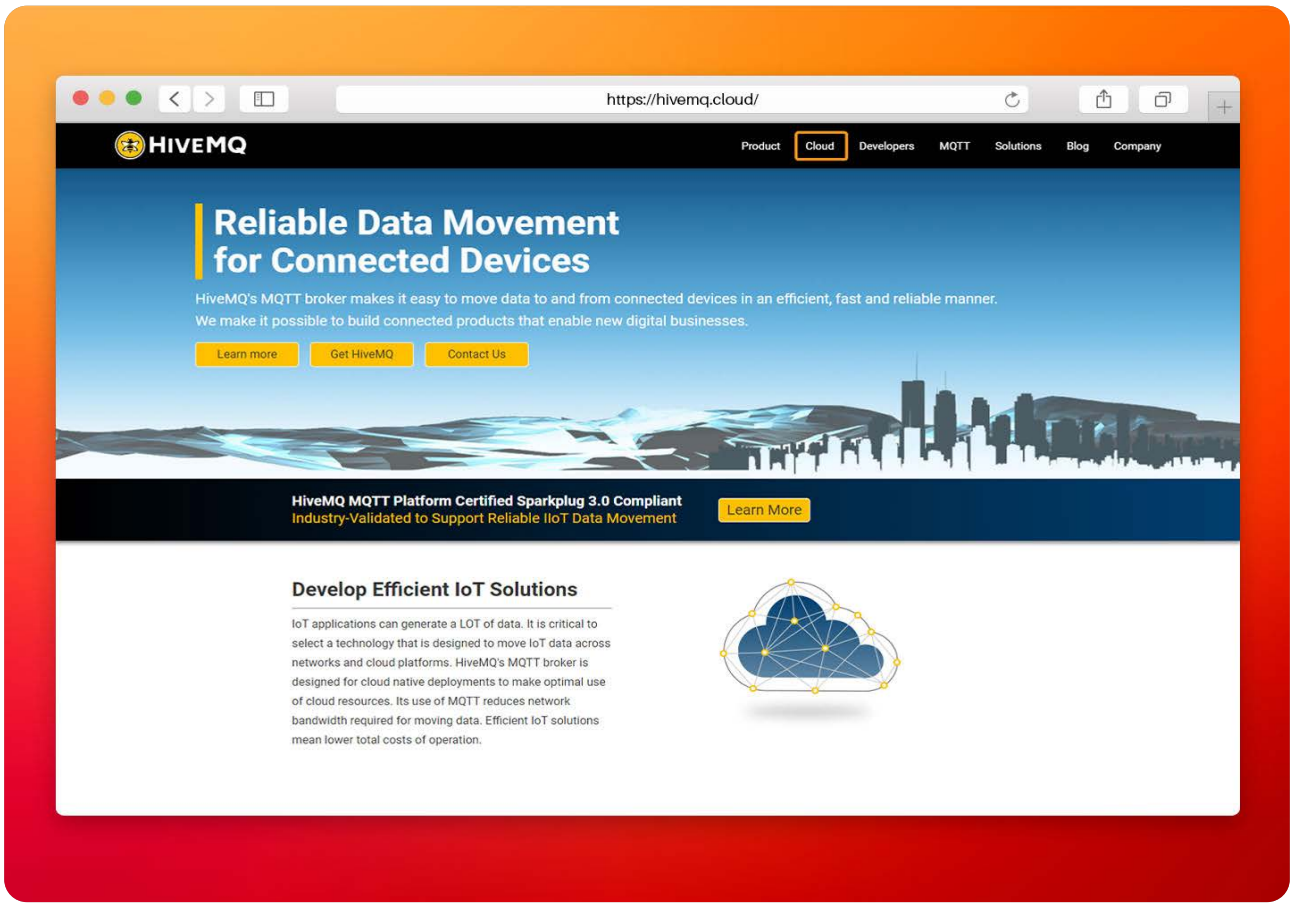
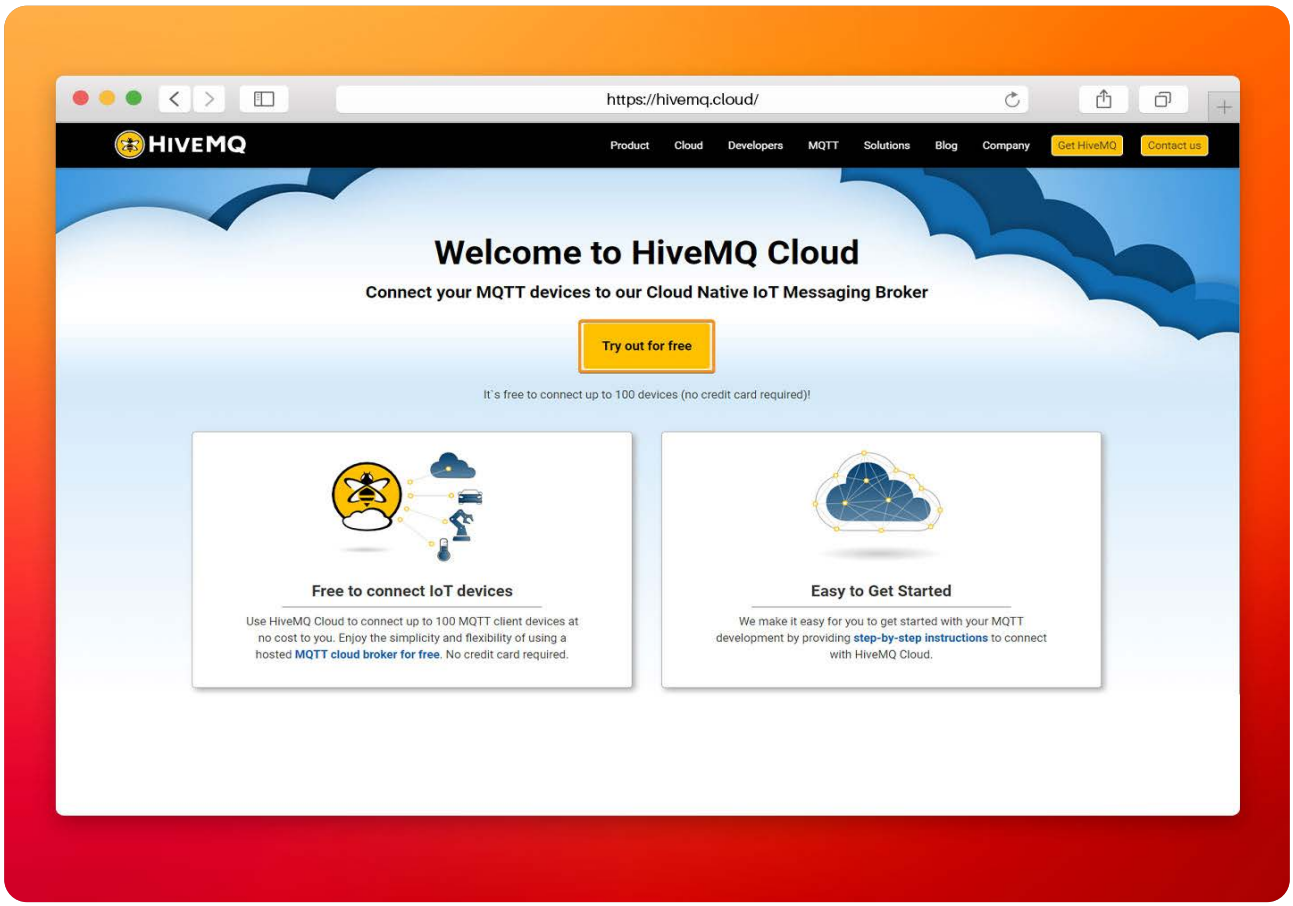


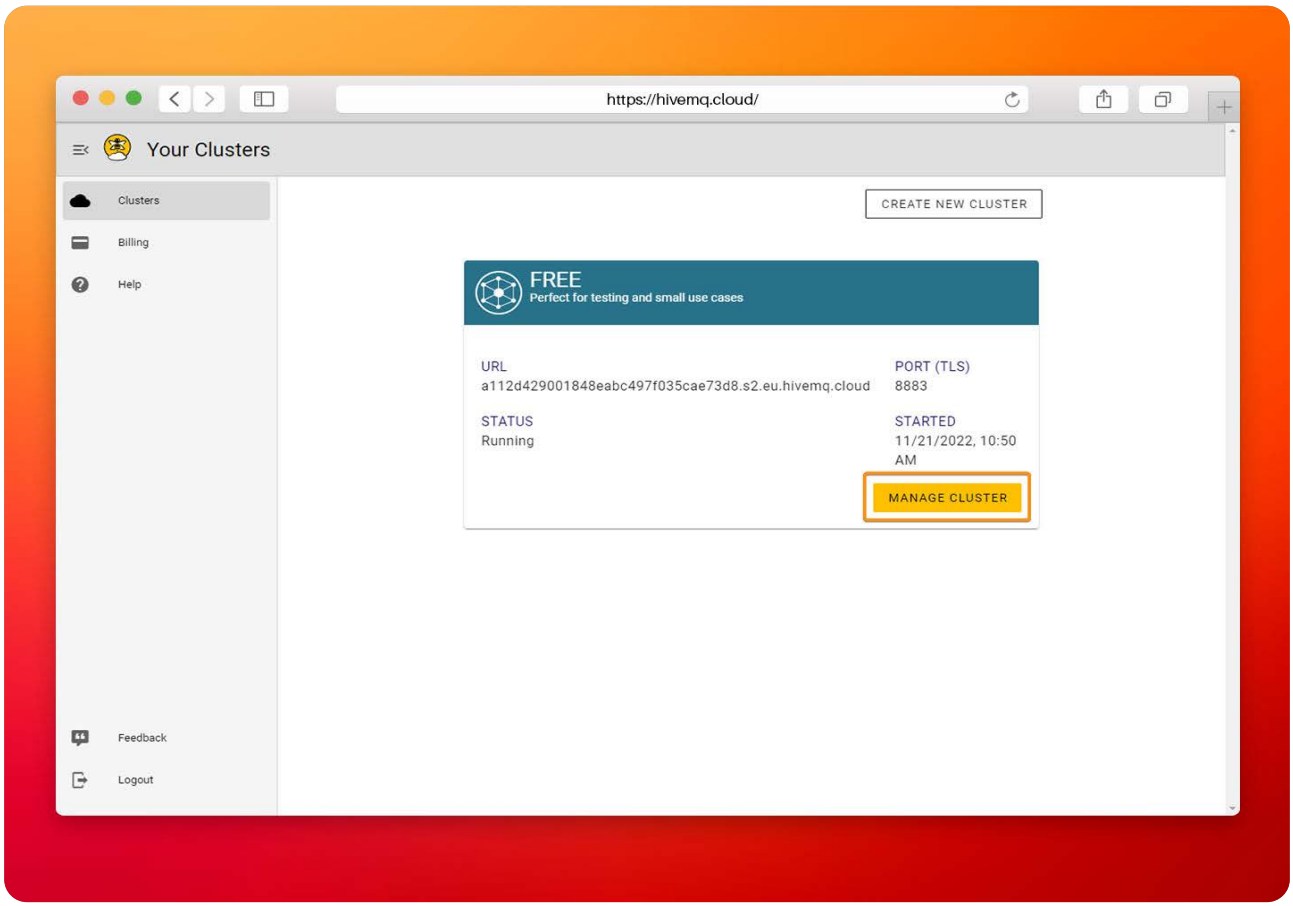
Image displaying the HiveMQ cloud access.

Step 2: Information related to HiveMQ cloud brokers and the different types of subscriptions they offer will now be shown on the page. Click the "**Try out for free**" button to get a free trial.



Screenshot showing the HiveMQ free trial button.

Step 3: Once you have signed up or logged in, select **Manage Cluster**.



Screenshot displaying the HiveMQ cloud cluster overview.

Step 4: Some important information will appear on the overview page, such as the MQTT Client active sessions, data traffic, and connection settings, where we will find the port and the cluster URL. Before beginning to establish a connection, click on the "**Access Management**" tab in the header.

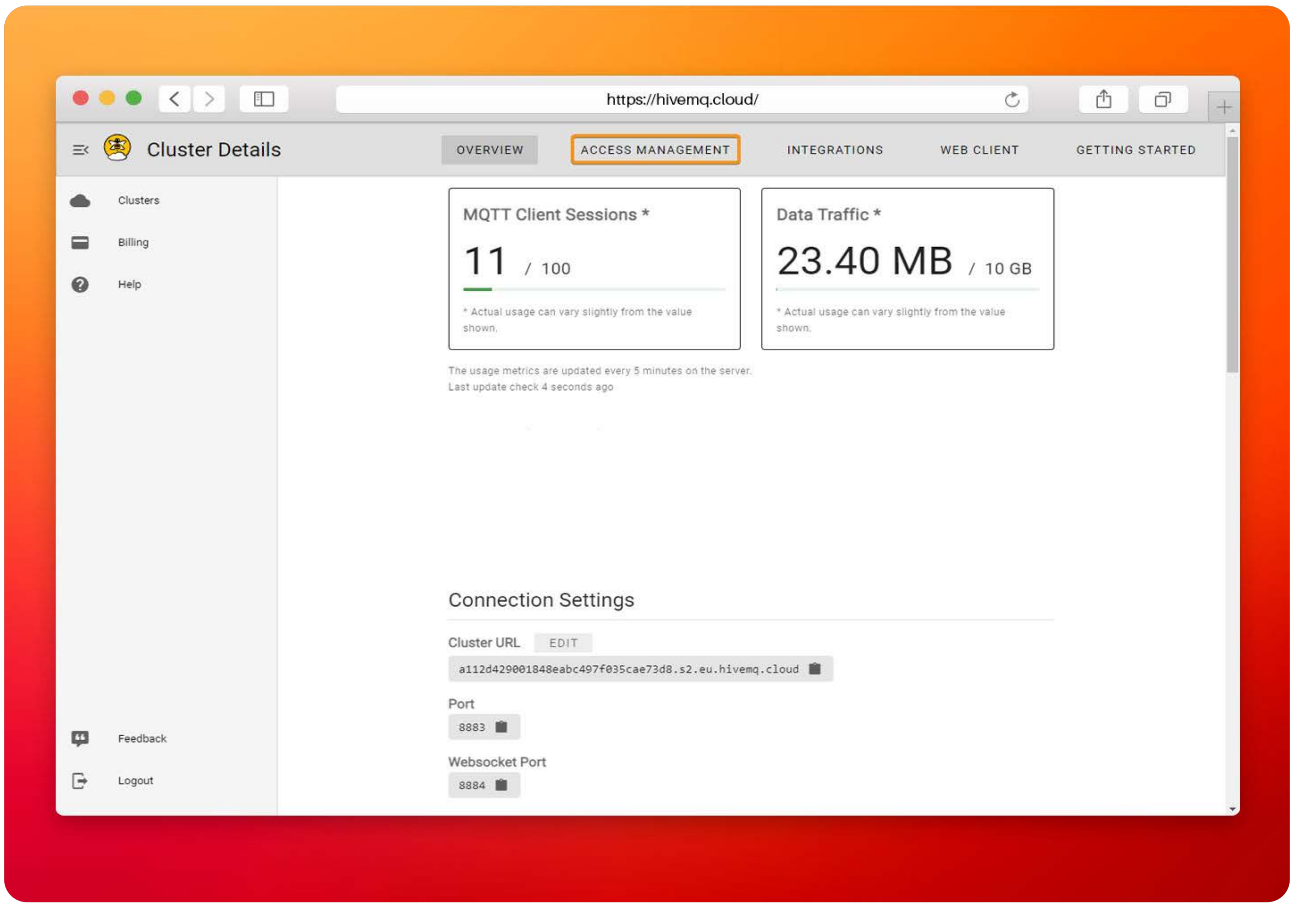
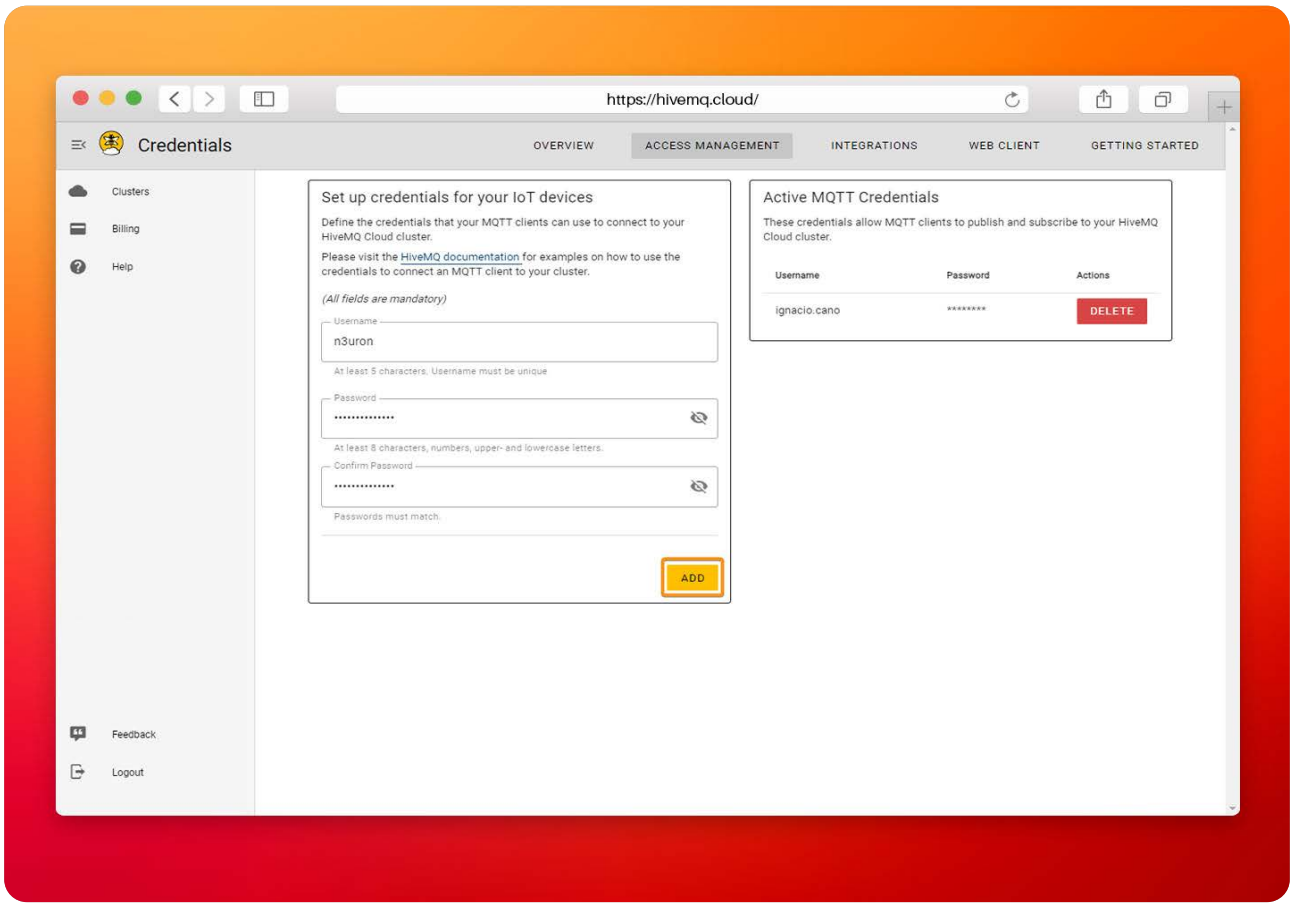


Image showing the HiveMQ cloud access management.

Step 5: A username and password must be defined in order to allow clients to access the cluster. Go to "**Set up credentials for your IoT devices**" and fill in these fields. When set, you will see the new user in "**Active MQTT credentials**".



Screenshot displaying the HiveMQ access management credentials.

Configuring the MQTT Client Module in N3uron

It is assumed that you have already created an MQTT Client instance. If not, please take a look at Create a new module instance.

Step 1: Once created, select your module instance from the **Explorer Panel**, click the button on the left-hand side of the **Model** header, select **New connection**, and give it a name. In this case, we have called it HiveMQ.

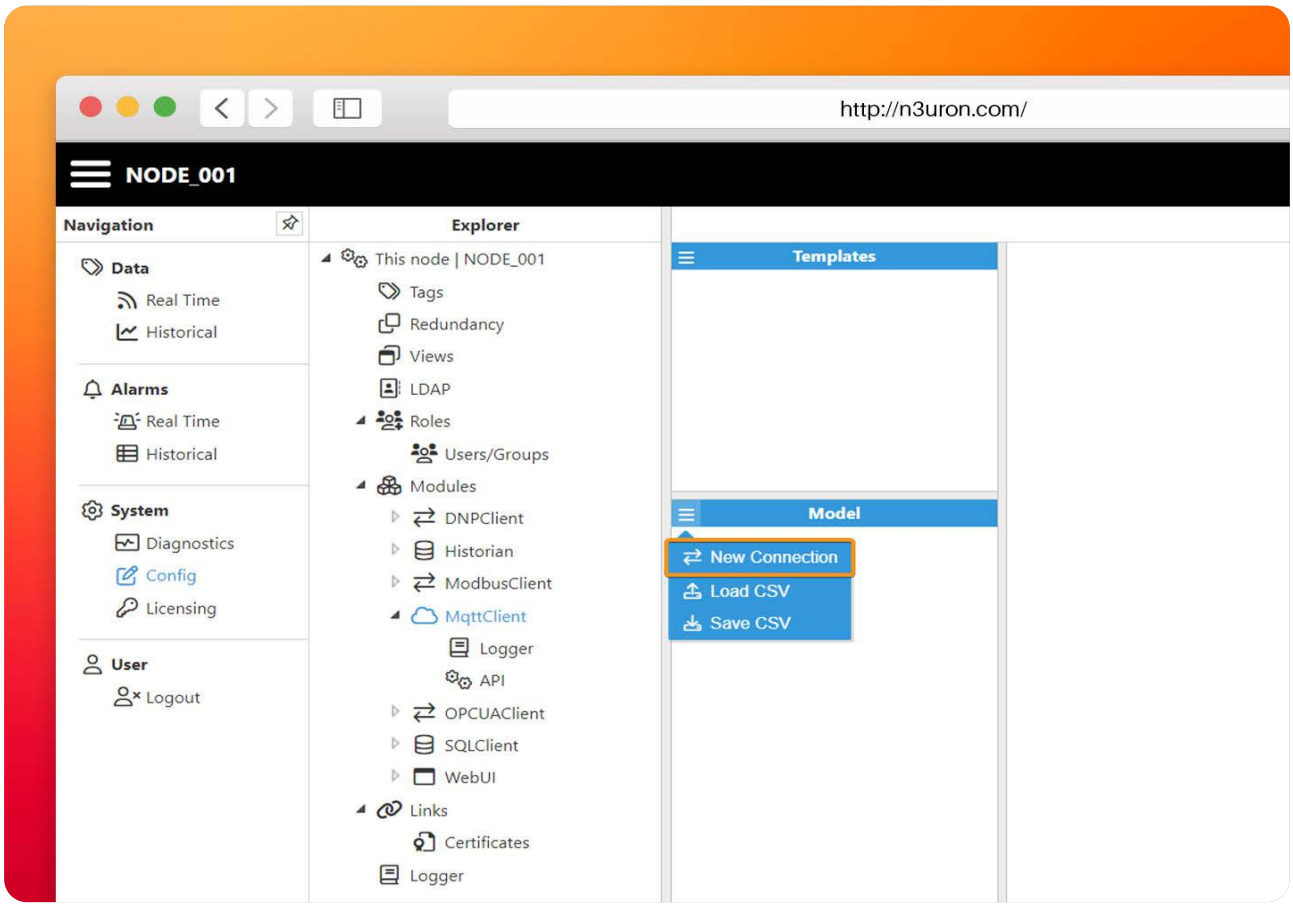


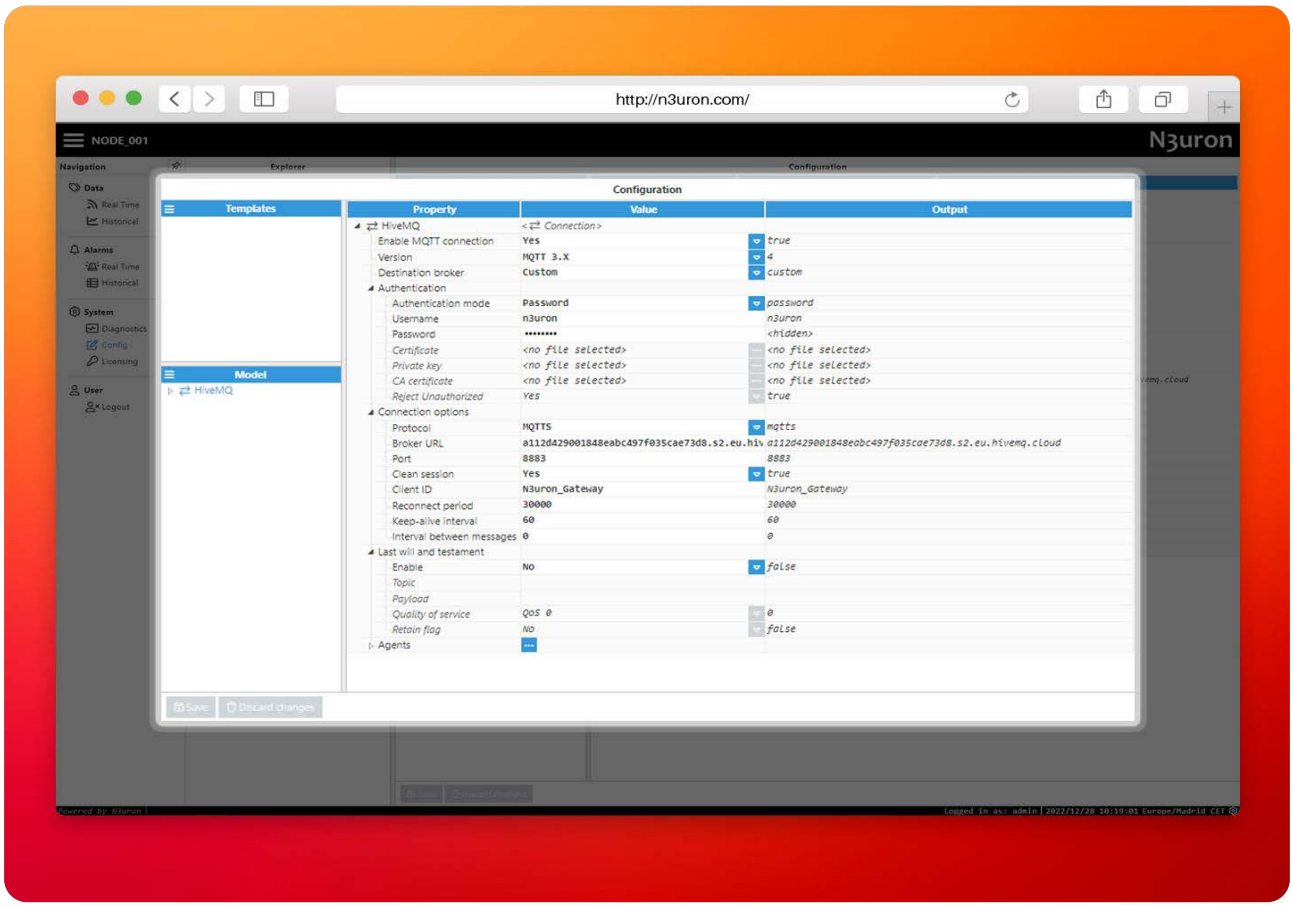
Image showing how to create a new MQTT connection.

Step 2: Configure the rest of the parameters as follows:

– Enable MQTT Connection: Yes.
– Version: MQTT 3.X.
– Destination Broker: Custom.
– Authentication:
Authentication mode: Password.
Username: Enter the username (the one you set up in your HiveMQ Cloud cluster credentials).
Password: Enter the password (the one you set up in your HiveMQ Cloud cluster credentials).

– Connection options:
Protocol: MQTTS.
Broker URL: Enter your HiveMQ Cluster URL.
Port: 8883.
Clean session: Yes.
Client ID: Select an id for your MQTT Client. For example, N3uron_Gateway.
Reconnect period: 30000.
Keep-alive interval: 60.
Interval between messages: 0.

– Last will and testament:
Enable: No.



Screenshot displaying the N3uron's MQTT client module configuration.

Step 3: Generate a publisher agent. Click on the ellipsis button to the right of "Agents" and select "**New publisher**".

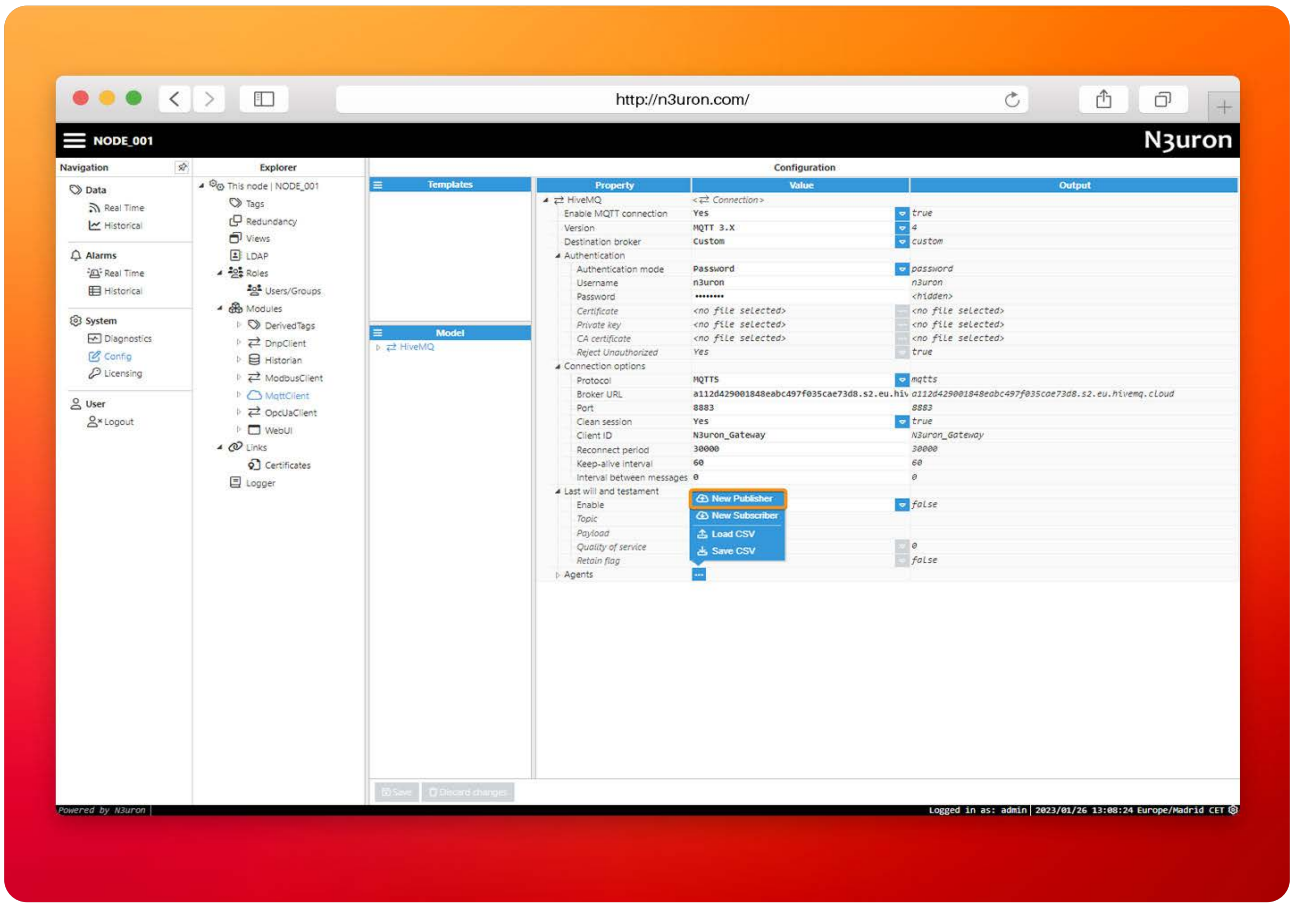
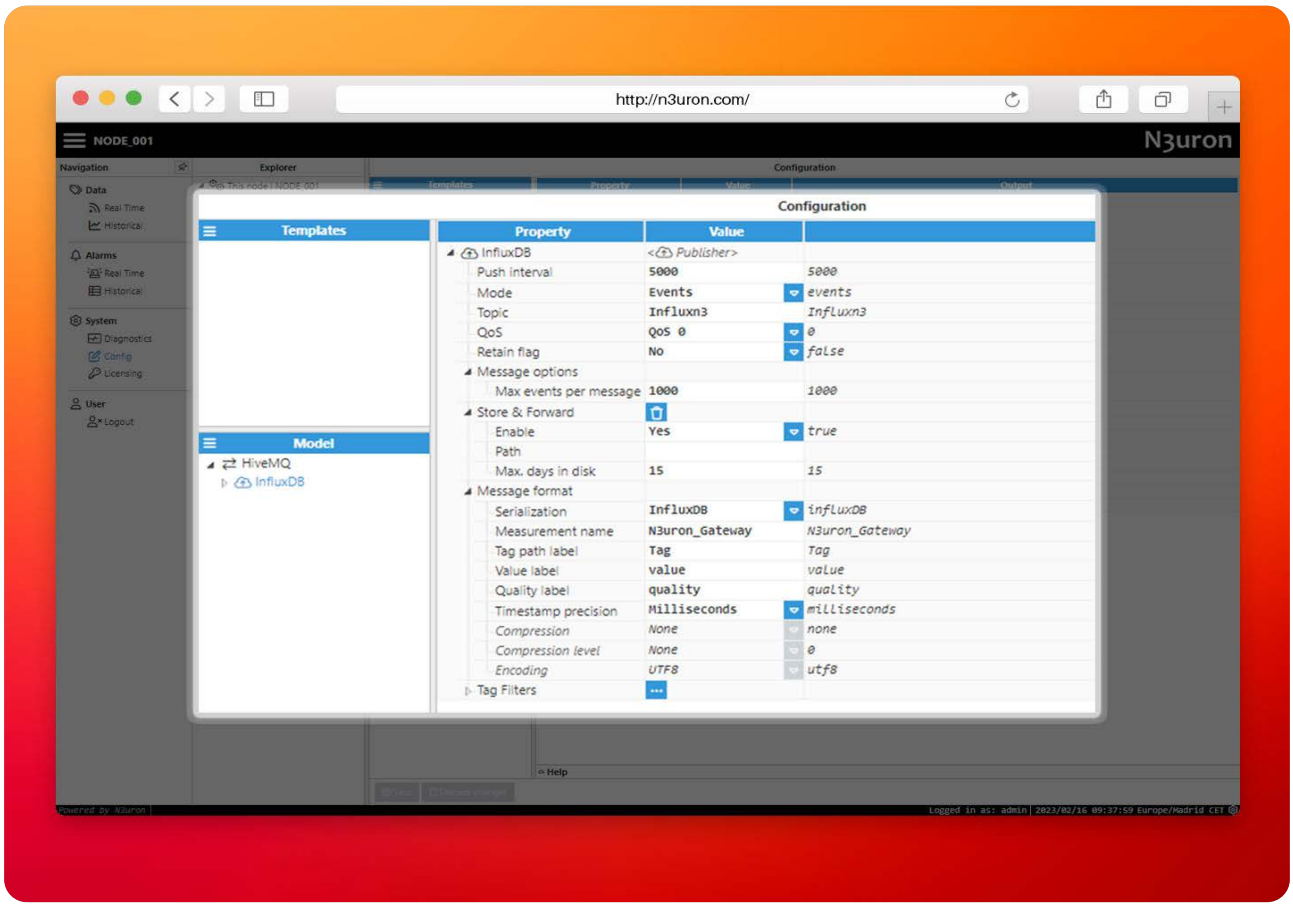


Image displaying how to create a new Publisher in the N3uron's MQTT client module.

Step 4: Give a name to the publisher (for example InfluxDB) and configure the rest of its parameters:

– **Push interval:** 5000.
– **Mode:** events.
– **Topic:** Define the topic for the subscribers to identify it. We called it Influxn3.
– **QoS:** 0.
– **Retain flag:** false.
– **Message options:**
Max events per message: 1000.
– **Store & Forward:**
Enable: true.
Path: If left blank, the path will be <program_root>/data/MqttClientInstance/ConnectionName/AgentName/.
Max. days in disk: 15.
– **Message format:**
Serialization: InfluxDB.
Measurement name: This will be the measurement name sent to InfluxDB. For example, N3uron_Gateway.
Tag path panel: Tag.
Value label: value.
Quality label: quality.
Timestamp precision: milliseconds.
Compression: None.
Encoding: UTF8.



Screenshot showing the MQTT publisher configuration.

Step 5: Create a **Tag Filter** and give it a name (for example, Voltage). You can leave the rest of the parameter default values or select only the tags you want to send by including or excluding the ones whose **path** is specified in the following field. In this case, we have only **included** the Voltage tags. Do not forget to **save changes** when you finish.

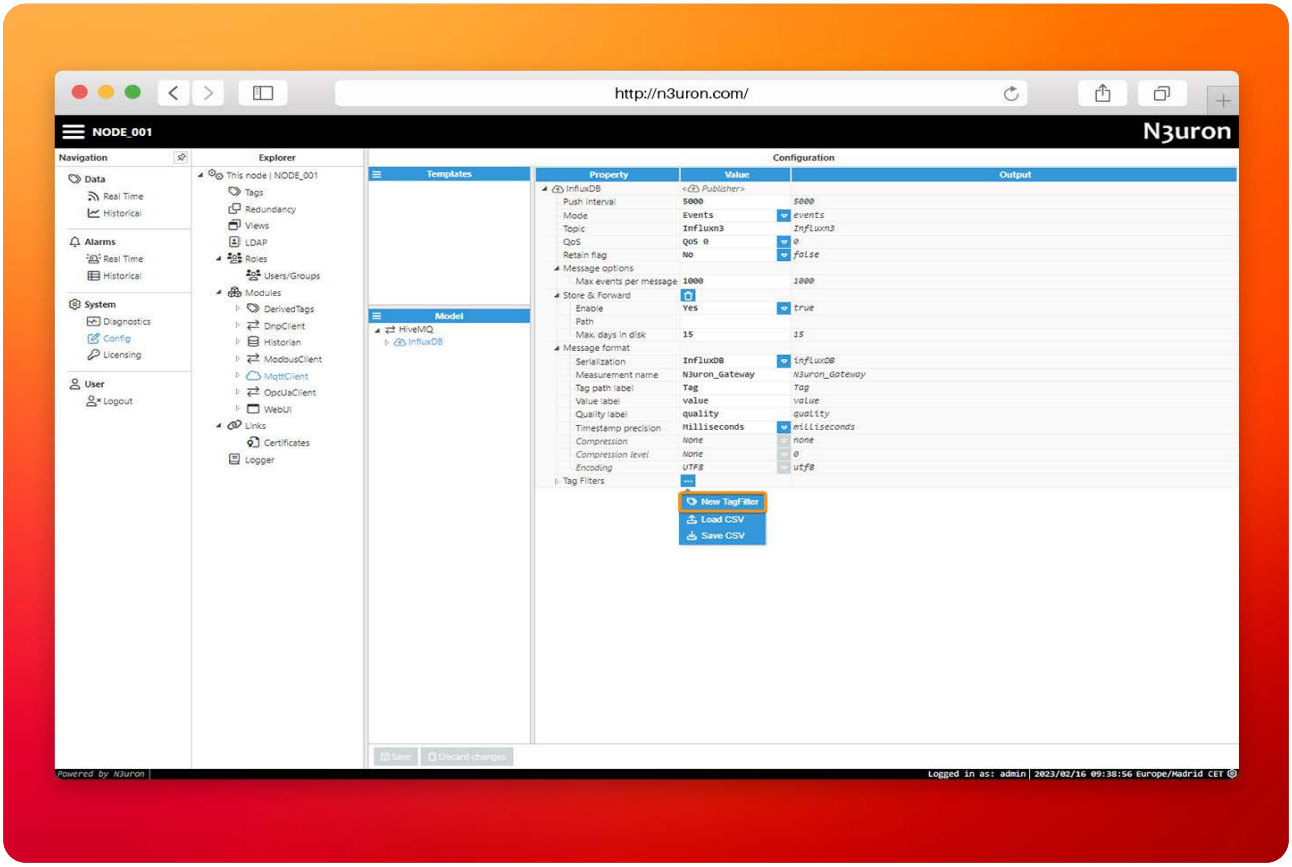
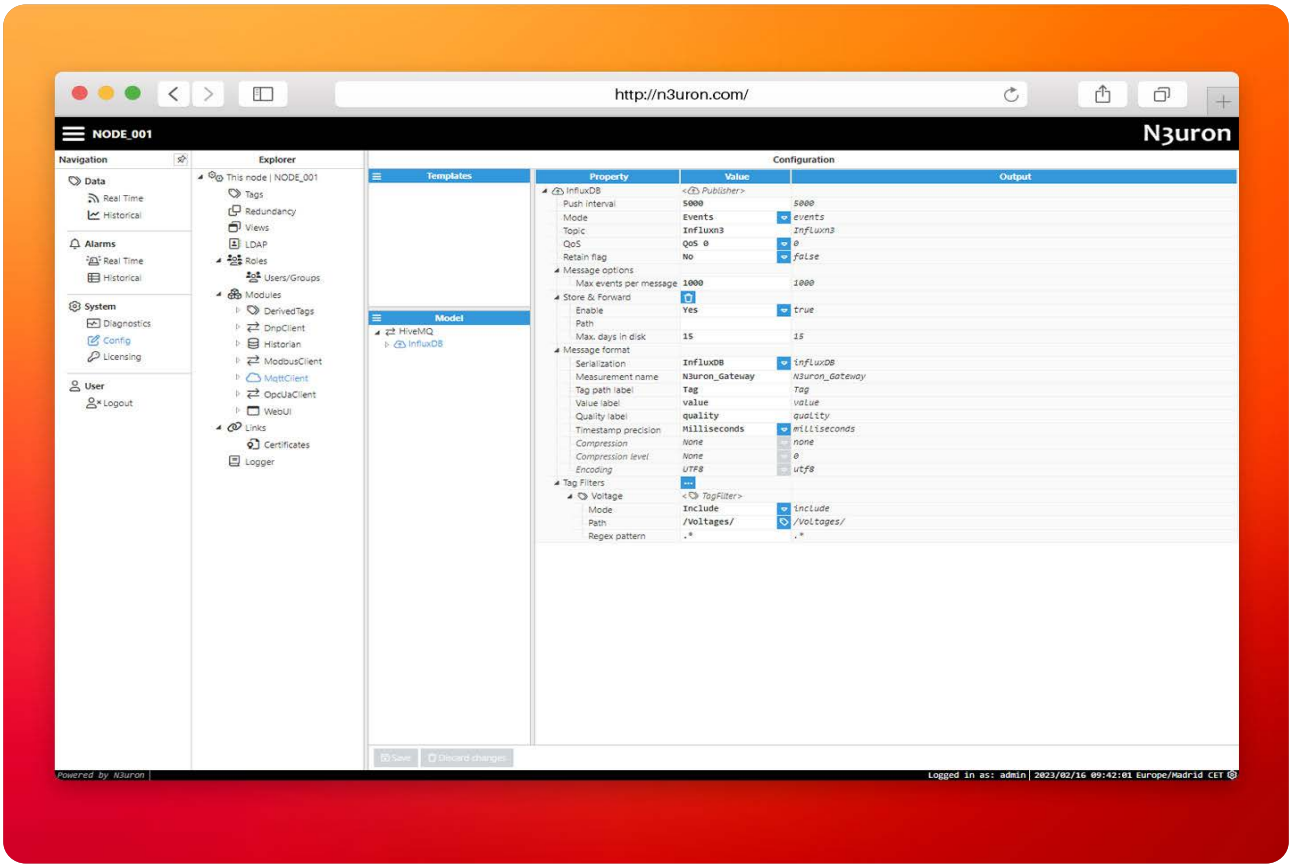


Image displaying how to create a new TagFilter in the N3uron's MQTT client module.



Screenshot displaying the MQTT TagFilter configuration.

InfluxDB Cloud Configuration for Receiving MQTT Data

The last step to completing your connection starts by gaining access to InfluxDB Cloud. If you do not have an account yet, you can get a new one for free, or register using your google or Microsoft account.

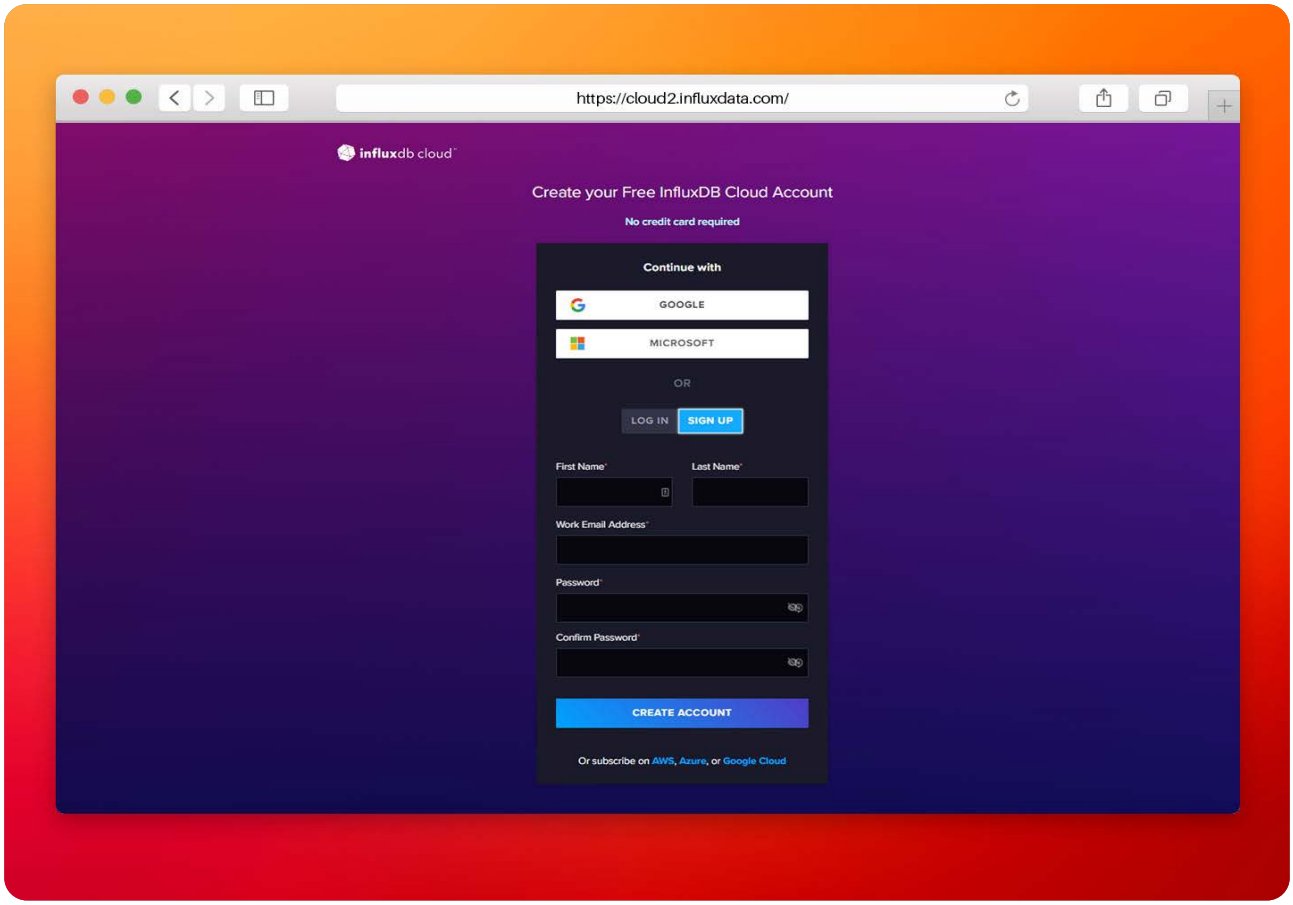
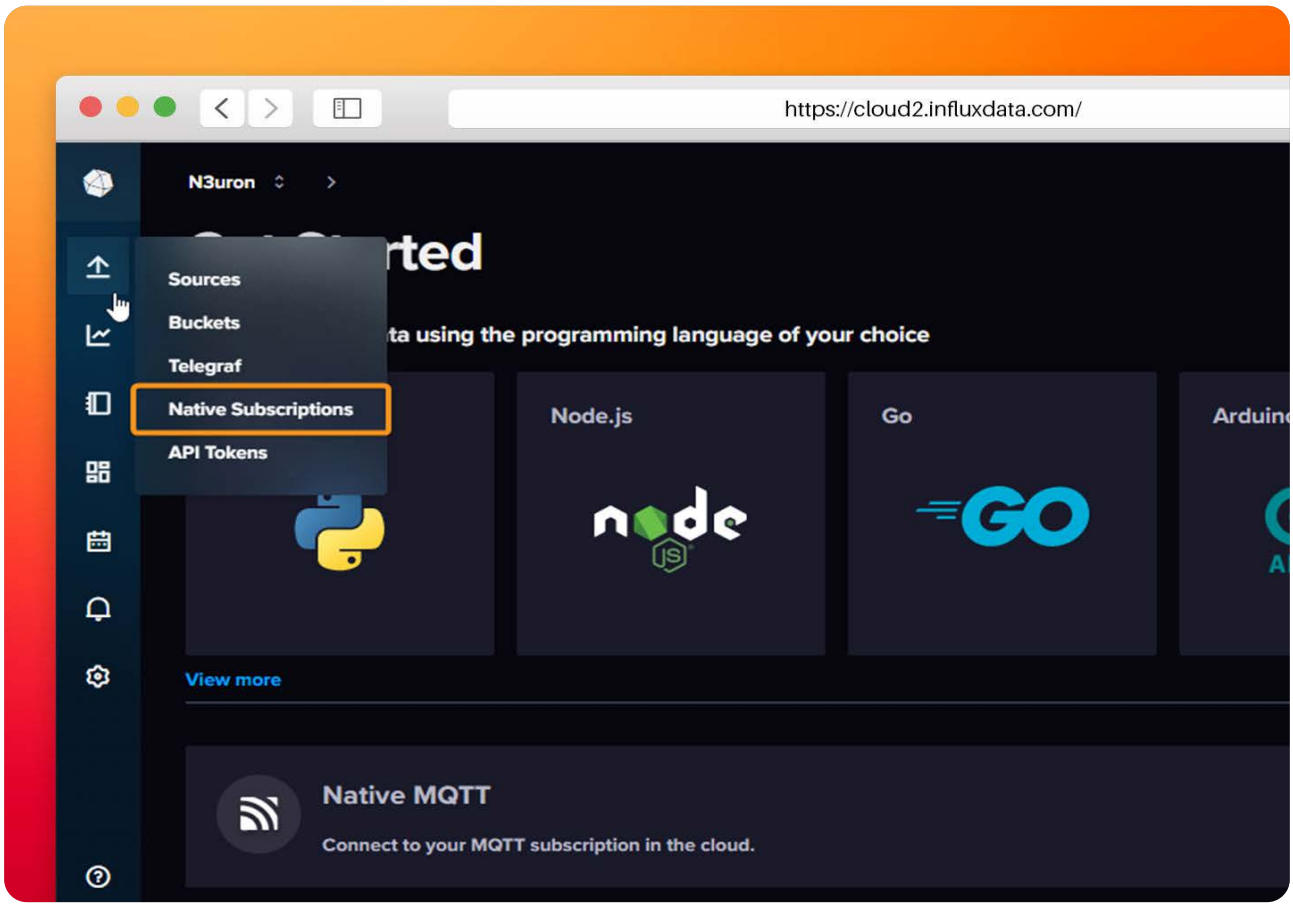


Image showing the InfluxDB cloud access.

Once you are signed in, follow these steps to store your MQTT data in influxDB:

Step 1: Navigate to "**Native subscriptions**" by clicking on the upward facing arrow on the left of the screen.



Screenshot displaying the InfluxDB native subscriptions button.

Step 2: When the new page loads, enter the following information to complete your subscription to the HiveMQ broker:

– Broker details:

Subscription name: Choose a name for the subscription. We recommend choosing the same as the N3uron measurement name. In this case, it's N3uron_Gateway).

Description: This field gives you the possibility of adding extra information about the connection and subscription topic.

Hostname or IP Address: Enter your HiveMQ Cluster URL.

Port: 8883.

Enable SSL: Enabled.

Client ID: A new Client ID will be automatically generated.

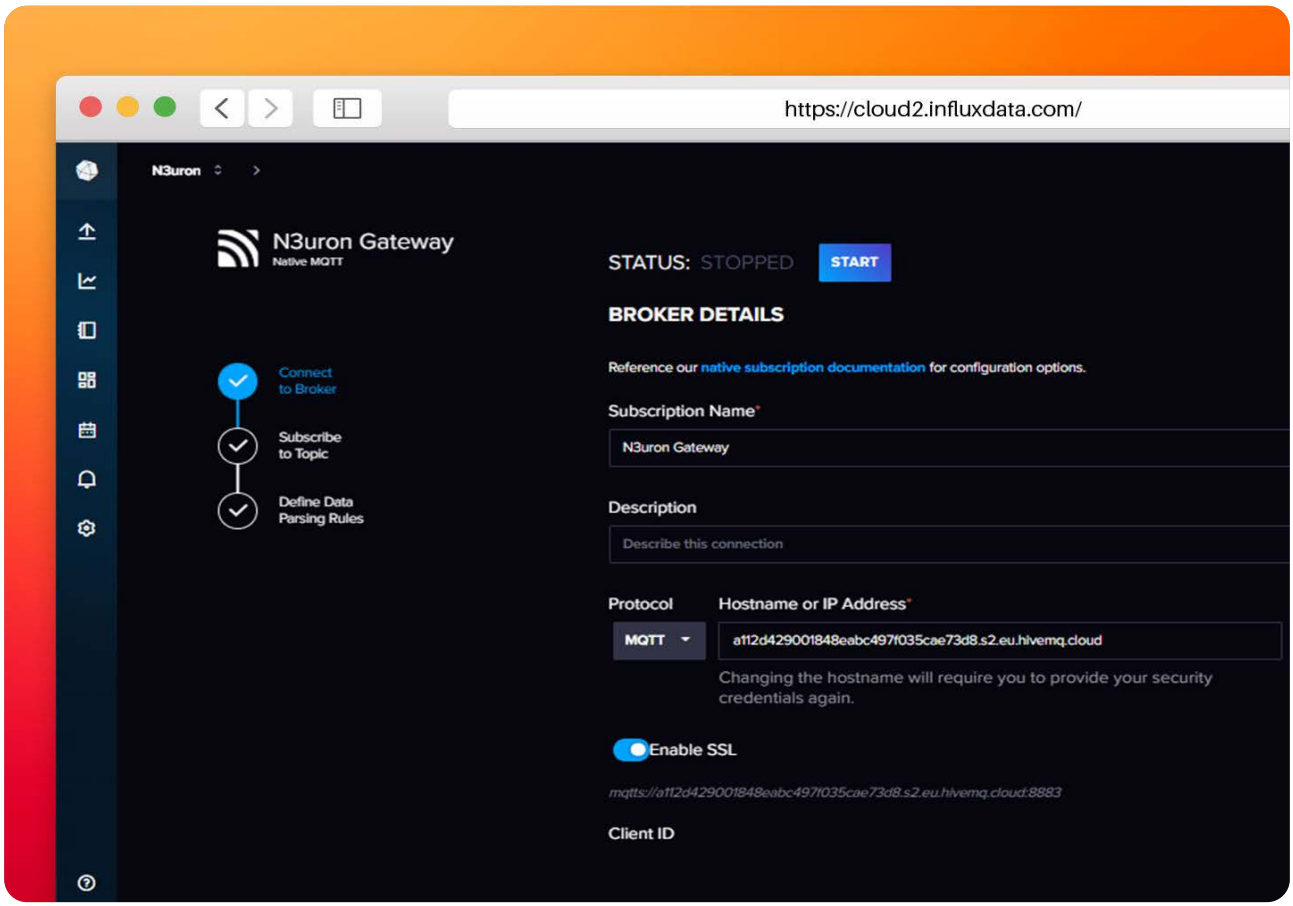
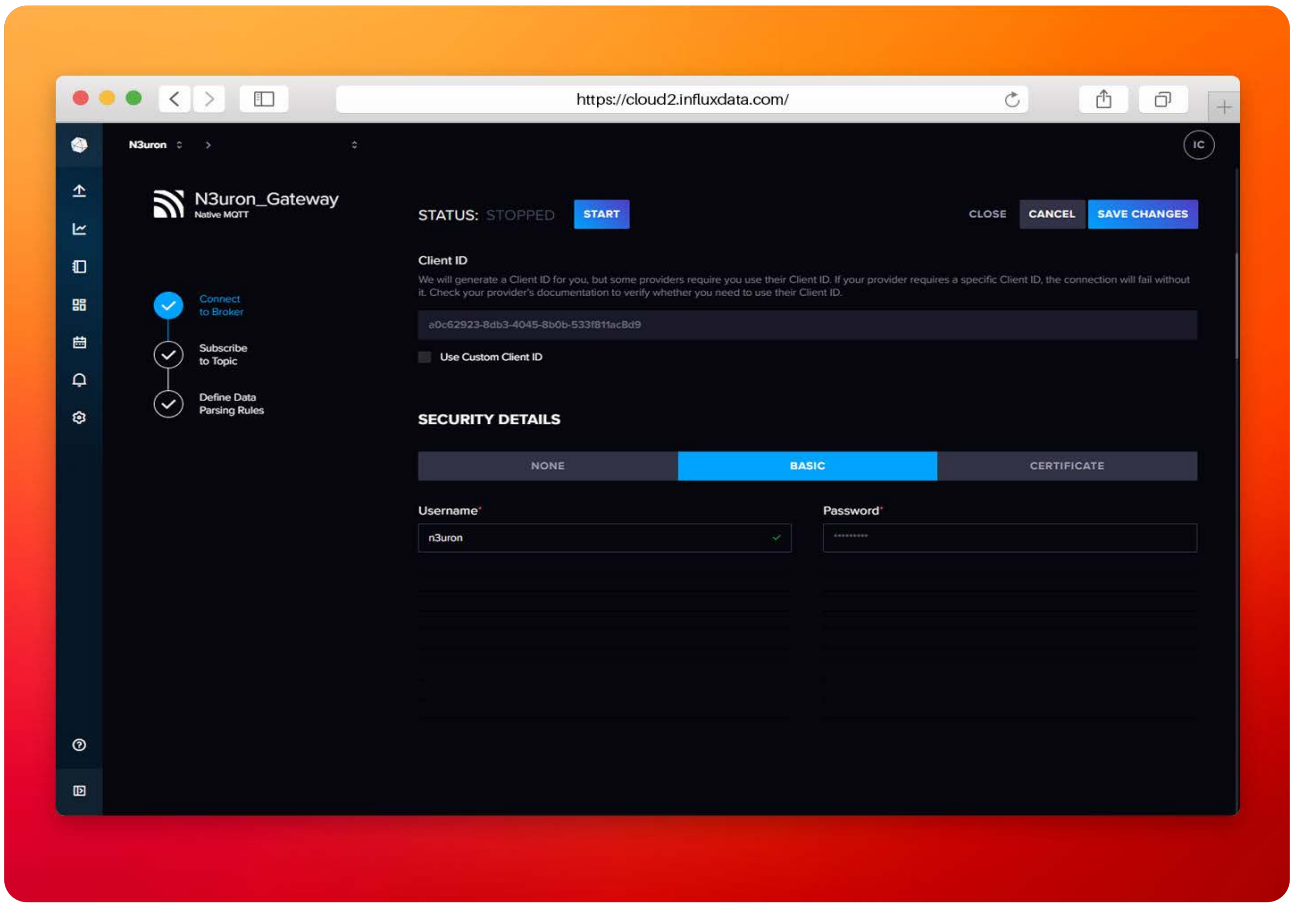


Image showing the InfluxDB subscription broker details.

– **Security details:**
Plan: Basic.
User and password: Enter your HiveMQ broker Active MQTT credentials.



Screenshot displaying the InfluxDB subscription security details.

Step 3: In the next section, you'll need to complete the following fields related to the topic subscription:

– Topic subscription:

Topic: Enter the name of the **topic** you want to receive MQTT messages from, followed by "**/#**" (in this case: Influxn3/#).

– Write destination:

Bucket: Select the **bucket** where you want to write your messages to (a bucket is a named location where time series data is stored). If there are no buckets yet, simply create a new one and give it a name (for example: N3uron Tags).

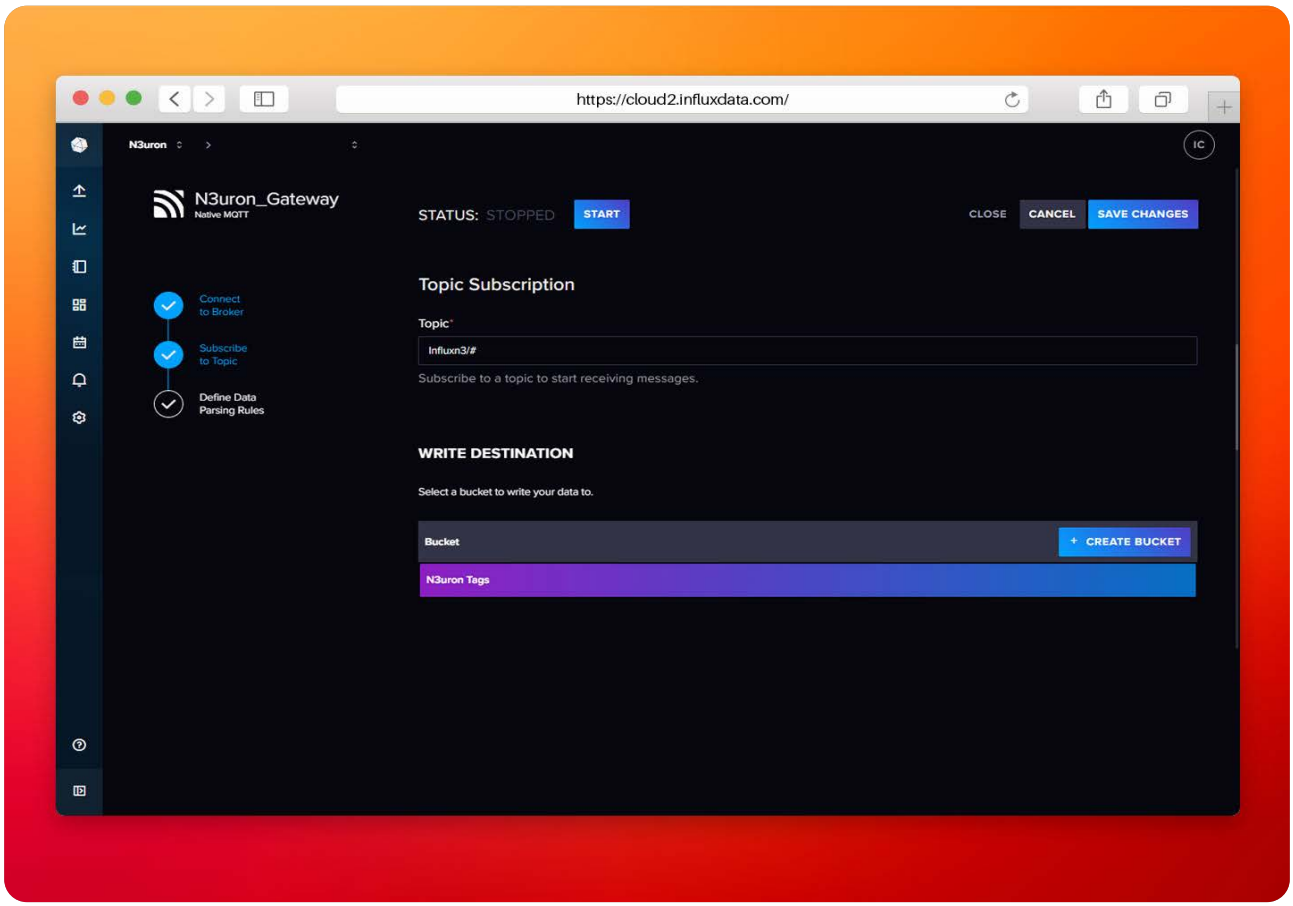
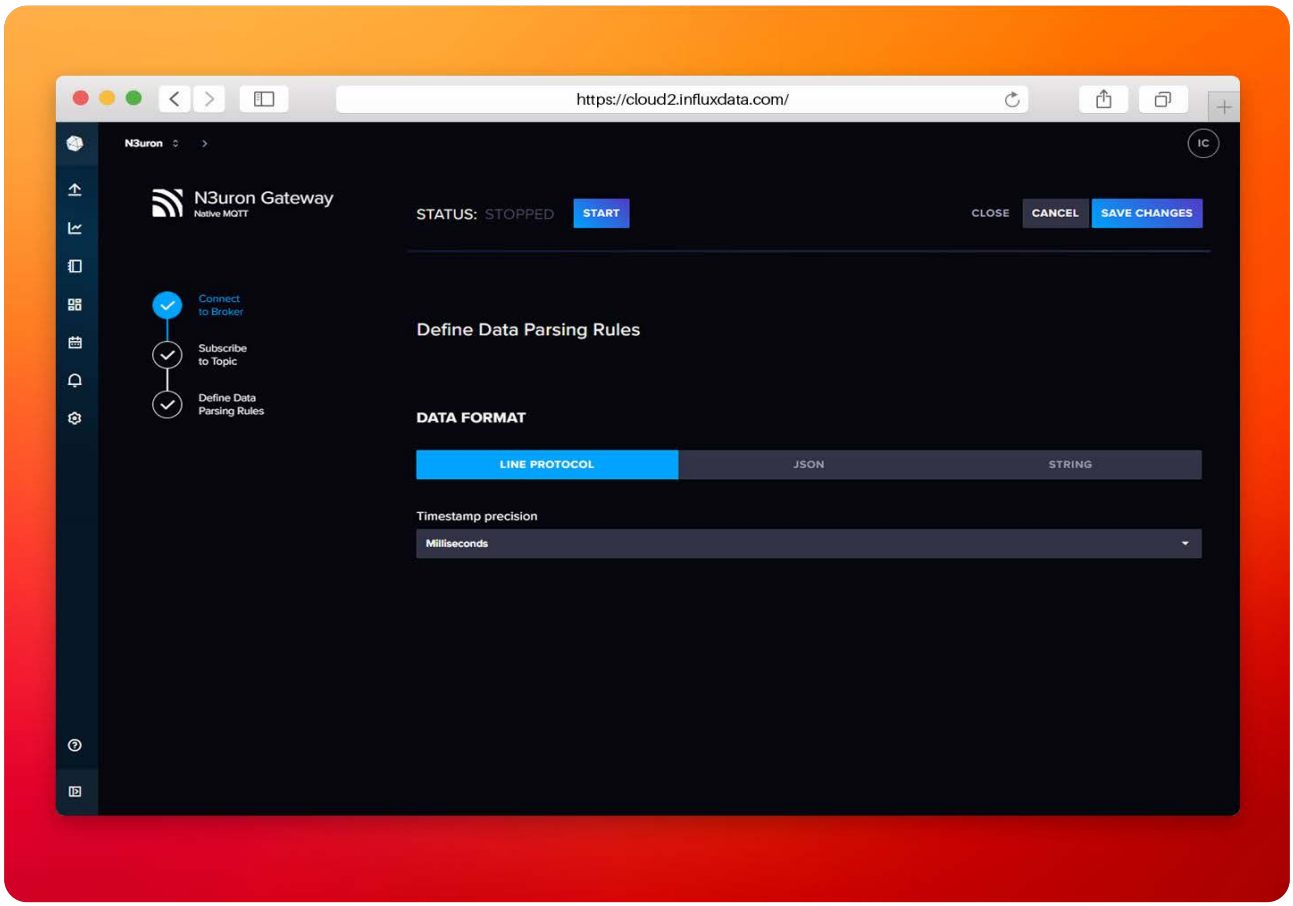


Image showing the InfluxDB susbscription topic.

Step 4: Finally, define the data parsing rules and save the subscription:

– **Data format:** Select Line protocol.
Timestamp precision: milliseconds.



Screenshot displaying the InfluxDB subscription parsing rules.

Visualizing MQTT Data in InfluxDB

InfluxDB allows users to display data received in many different formats. In order to visualize data pushed from N3uron, proceed as follows:

Step 1: Make sure your subscription is running. Once created, select the "**Native subscriptions**" tab and check its **status**.

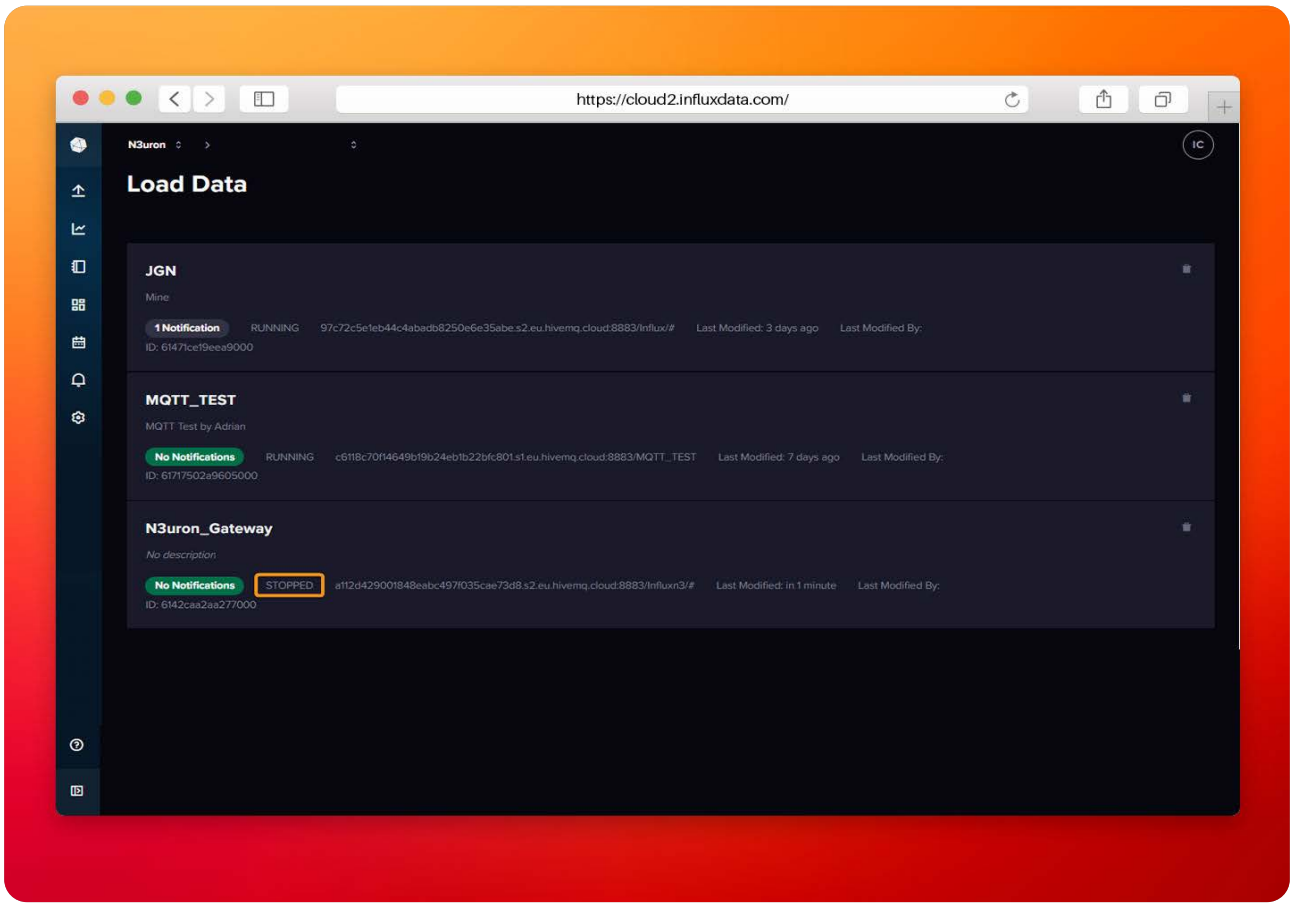
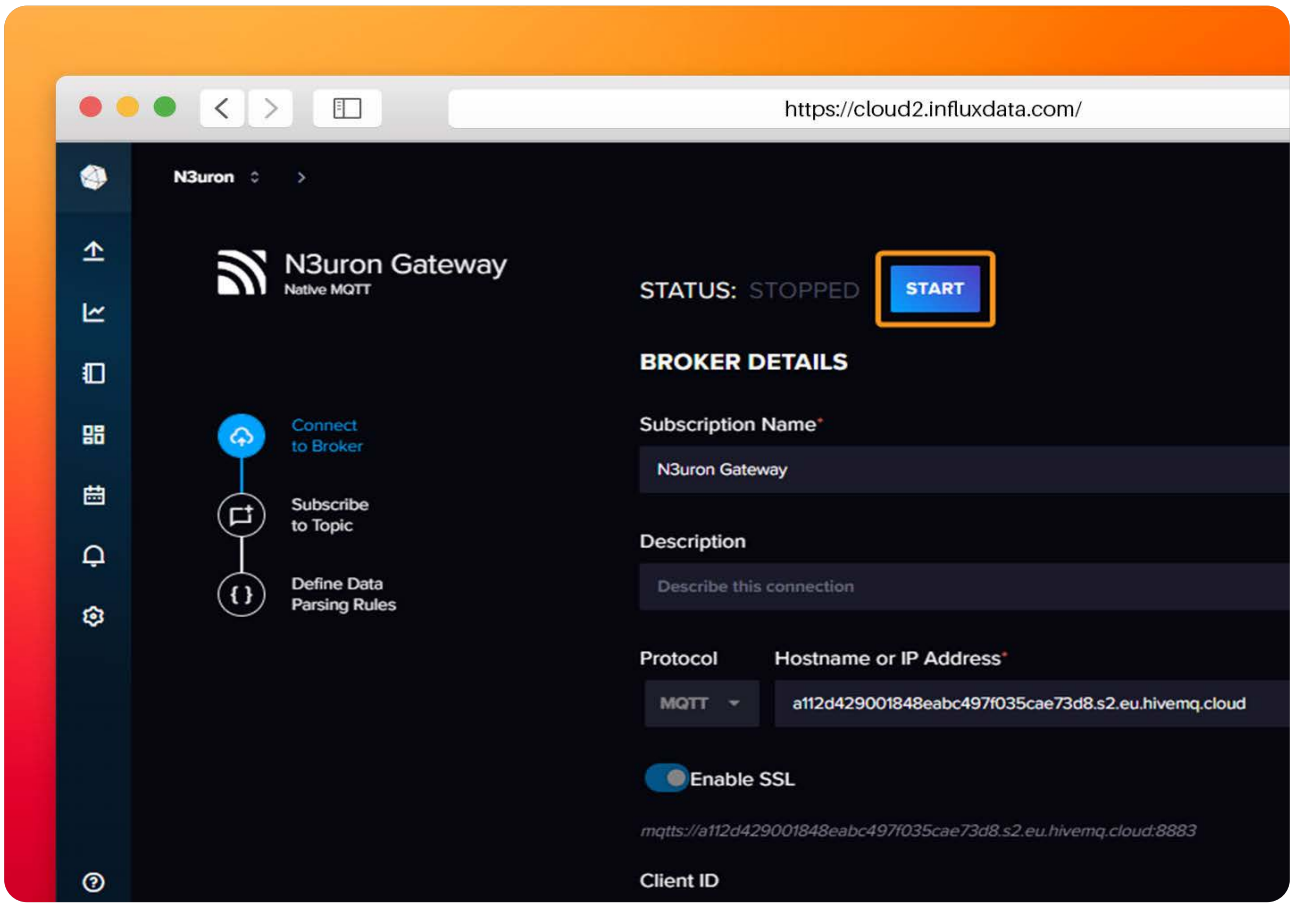


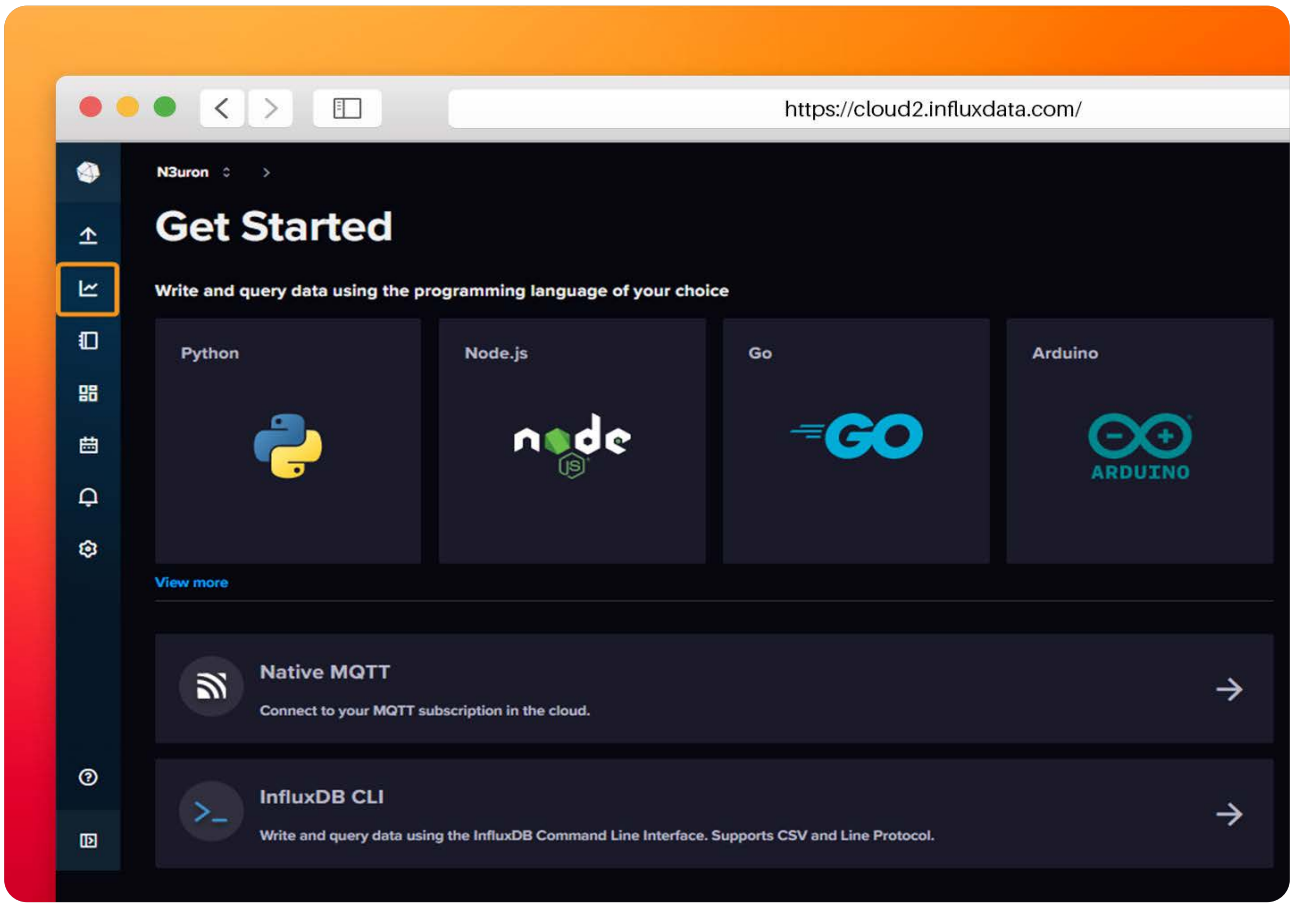
Image showing the InfluxDB subscription status check.

Step 2: If it has stopped, access your subscription and click on the "**Start**" button.



Screenshot displaying the InfluxDB subscription start button.

Step 3: Once the subscription is running, navigate to "**Data explorer**" by clicking on the second button starting from the top on the left-hand side of the screen.



Screenshot displaying the InfluxDB cloud data explorer button.

Step 4: Select the bucket you want to retrieve information from, as well as the Measurement, tag values, and fields, and start visualizing your data.

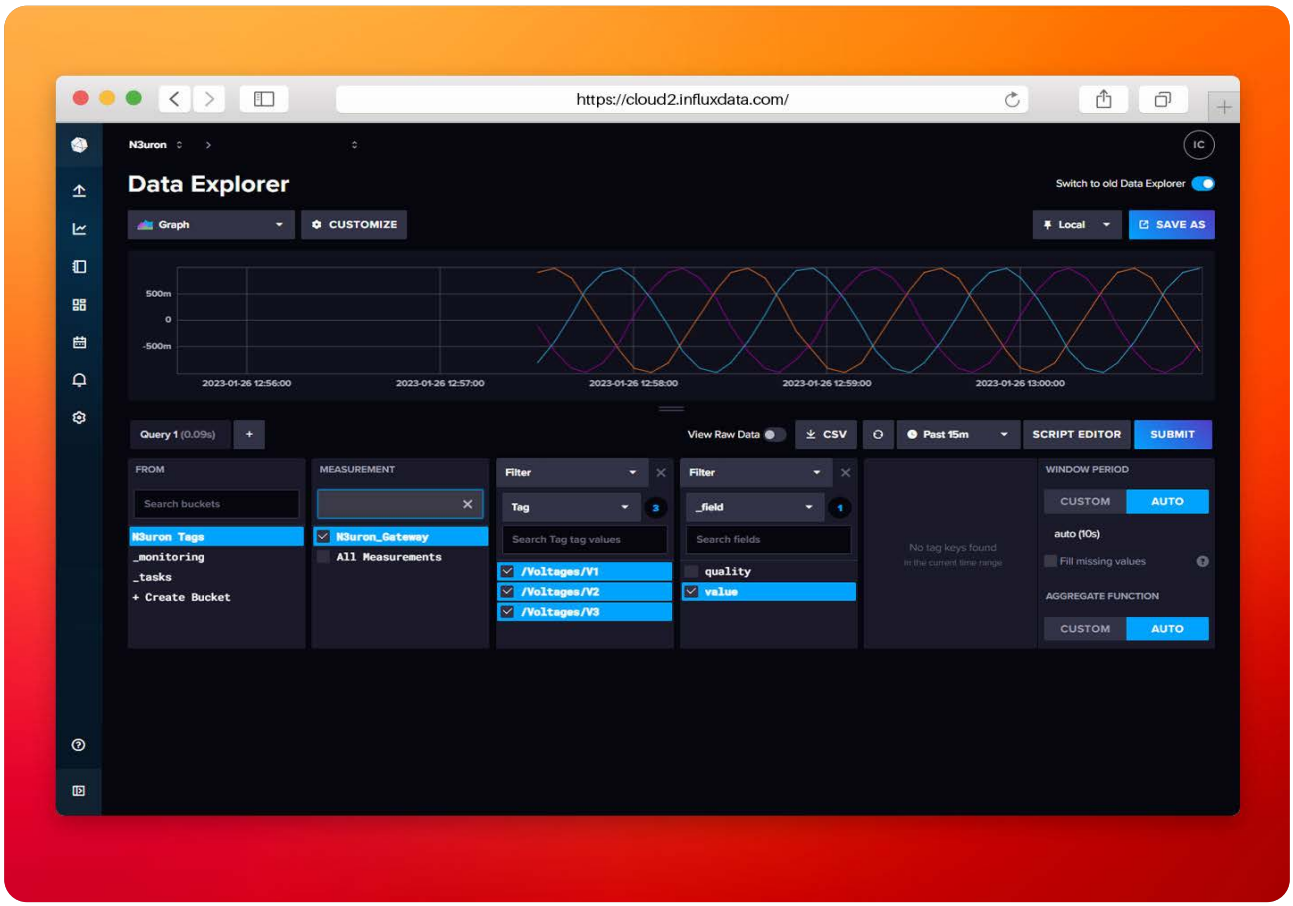


Image showing the InfluxDB data visualizing.

Conclusion: Seamlessly Push your Data from N3uron to InfluxDB Cloud using MQTT

In conclusion, we have seen how by combining **N3uron**, InfluxDB and MQTT, you can very quickly deploy a solution to start sending your IIoT data to a robust and fully scalable time-series database in order to gain insights from your industrial time-stamped data in record time.

If you are ready to start building your IIoT infrastructure, download the N3uron free trial version and read our MQTT Client Manual on how to implement and use N3uron's MQTT Client module in our IIoT platform.