Product Documentation

EnOcean GmbH

Table of contents

1. EnOcean SmartStudio

1.1 EnOcean SmartStudio Platform

EnOcean SmartStudio is an IoT device management platform designed to manage EnOcean devices efficiently. It enables companies to easily install these devices to gather data on space utilization, air quality, energy consumption, and more. Additionally, EnOcean SmartStudio supports seamless integration with partner applications for workplace management, data analytics, AI, and other services.

Key features of EnOcean SmartStudio include:

- **Device Management**: Simplifies device setup and maintenance, reducing installation time and future upkeep.
- **Data Connectivity**: Enables device data transfer through HPE Aruba Network, eliminating the need for additional IoT gateways.
- **Integration Management**: Offers easy integration with third-party applications via MQTT and APIs.

1.2 Getting Started

1.2.1 Account Setup

To get an account on EnOcean SmartStudio please contact EnOcean Sales

Once an account is created you will receive login information to access the EnOcean SmartStudio Interface for managing devices, gateways and adding integrations. For new users of EnOcean SmartStudio, it is recommended to follow below Quick Start Guide.

1.2.2 Quick Start Guide

This guide will walk you through the steps to add an EnOcean STM 550 to EnOcean SmartStudio using the HPE Aruba Network, and how to view the device data.

Step 1: Enter Topology

Before adding devices, you must first configure the building topology that represents where the devices will be installed. The topology follows a hierarchical structure:

World → Site → Building → Floor → Room

This structure allows you to define multiple sites (such as cities or campuses) and multiple buildings within each site. To create the topology:

1. In the **Topology** section, hover over **World** and select **Add Child**.
2. Enter a name for the site, then select **Create**.
3. Continue building the topology by selecting **Add Child** on the newly created site to add a **Building**.
4. Repeat this process to add **Floors** and **Rooms** as needed.

Step 2: Adding STM 550 to EnOcean SmartStudio

The following steps guide you through adding an STM 550 device to SmartStudio.

1. Navigate to the **Devices** section and click **Add Device**.
2. **Scan** the QR code being shown in SmartStudio to complete the commissioning on your phone.

On Your Phone

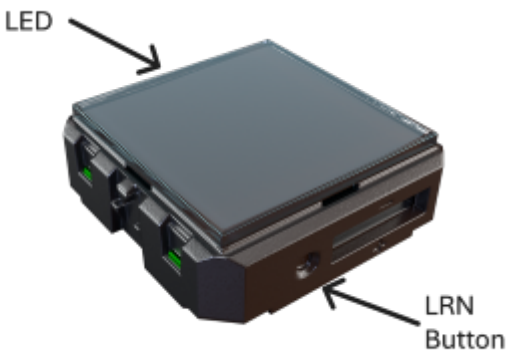
1. Select the desired **location** and tap **Next**.
2. Allow the app to access your camera, then **scan** the QR code on the device.
3. Change the device name if desired.
4. To add additional devices to the same location, tap the **QR icon** and scan the next device.
5. Tap **Save** to add the device(s) to SmartStudio.

Note: On some phones, you may need to adjust camera settings.

Step 3: Activating the STM 550

The STM 550 is delivered in flight mode. To activate the device, press the LRN button (see picture below) using the following sequence:

1. Single press the LRN button, and the LED blinks green two times.
2. Wait a few seconds.
3. Triple press the LRN button, and the LED blinks green three times.



Note: Before using the device, ensure it is adequately charged by placing it under ambient lighting for few minutes.

Step 4: Configuring the HPE Aruba Network

Configure your HPE Aruba Network to forward STM 550 data to EnOcean SmartStudio. The configuration process depends on your Aruba network setup:

- **Aruba Instant**
- **Aruba Controller**
- **Aruba Central (AOS 8)**
- **Aruba Central (AOS 10)**

Step 5: Verify Connectivity

In SmartStudio, you can verify the device connectivity:

1. Navigate to the **Devices** section to see the list of added devices.
2. Verify the device **Status** is either yellow or green, based on below description.
3. Check the **Last Seen** time to see when the last telegram was received.

Under Device Status, a colored indicator shows the device activity

Status	Description
Green	Indicates that the reporting interval has been successfully calculated and the device is online and reporting according to its configured interval.
Yellow	Indicates that activity has been received and the platform is calculating the reporting interval after collecting a sufficient number of telegrams.
Red	Indicates that no activity has been received from the device.

Step 6: Viewing Data on the MQTT Interface (Optional)

You can now connect to the MQTT interface to monitor data from the STM 550.

1. Download and install MQTT Explorer.
2. Navigate to the **Integrations** section and select **MQTT Client**.
3. Copy **Username**, **Password**, and **Broker URL** needed to setup the connection.
4. Use these credentials to connect to the EnOcean SmartStudio MQTT interface in MQTT Explorer.
5. Subscribe to the relevant topics like `<your-MQTT-username>/#`.

For more information about MQTT integration, please refer to the MQTT documentation.

2. Devices

2.1 Device Overview

EnOcean SmartStudio supports devices from both EnOcean GmbH and members of the EnOcean Alliance. The integration includes automatic data decoding and device recognition for seamless use with the EnOcean commissioning tool. Below is a list of devices fully integrated with EnOcean SmartStudio.

Device Name	Device Information	Device Data	Application
STM 550	Manufacturer: EnOcean GmbH Product Type: sensor Product EEP: D2-14-41	Desk Utilization Temperature Humidity Illumination	Space Utilization Air Quality
EMDC	Manufacturer: EnOcean GmbH Product Type: sensor Product EEP: A5-07-03 / A5-07-01	Occupancy Detection Activity Count	Space Utilization
EMCS	Manufacturer: EnOcean GmbH Product Type: sensor Product EEP: D5-00-01	Magnet Contact Sensor	Space Utilization
SD-ENO-CO2	Manufacturer: Jumitech Product Type: sensor Product EEP: D2-14-59	CO2 Temperature Humidity	Air Quality
1 Phase CT Clamp	Manufacturer: Pressac Product Type: sensor Product EEP: D2-32-00	Current	Energy Management
3 Phase CT Clamp	Manufacturer: Pressac Product Type: sensor Product EEP: D2-32-02	Current	Energy Management
OPUS TRV	Manufacturer: OPUS Product Type: bidirectional Product EEP: A5-20-06	TRV	Energy Management

In addition, EnOcean SmartStudio offers broad support for EnOcean Equipment Profiles (EEP), enabling generic device integration through data decoding based on the device's EEP. However, devices supported only by their EEP will not be automatically recognized by the commissioning tool and must be added manually via the API interface.

2.2 Commissioning

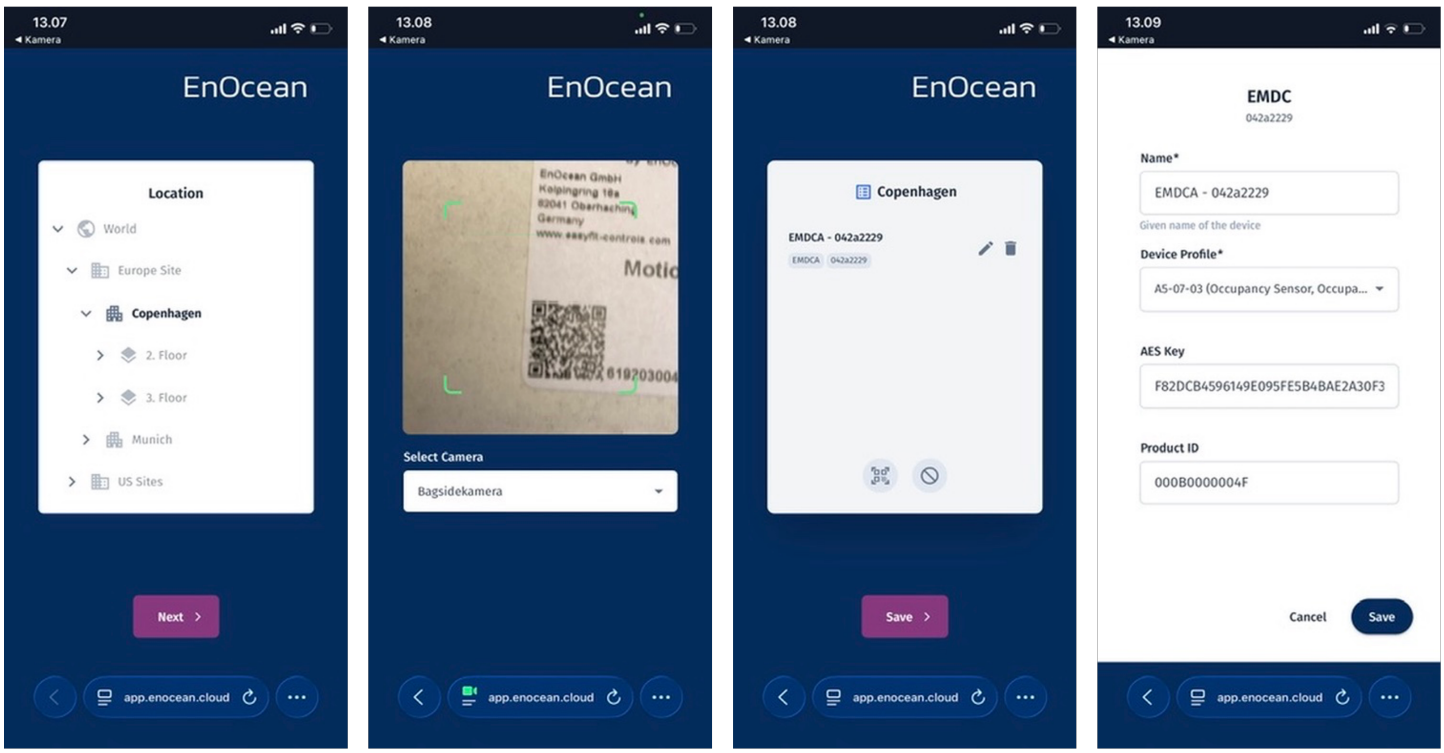
The following describes the device commissioning flow for adding devices to SmartStudio.

- Navigate to the **Devices** section and click **Add Device**.
- From the pop-up screen is it possible to complete the commissioning **Using Mobile** or by **Manual Entry**.

2.2.1 Using Mobile

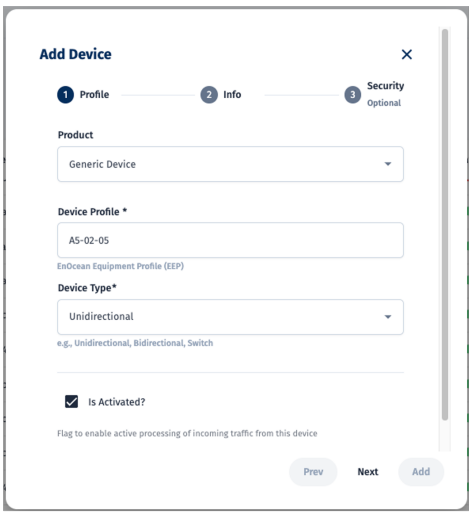
To complete the commissioning using mobile simply scan the on-screen QR code. The following commissioning flow will than open on the phone:

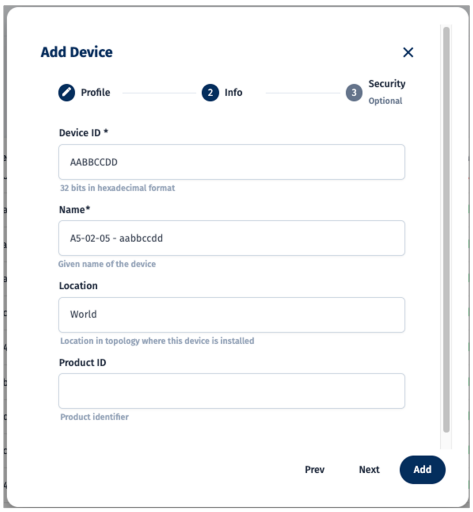
1. Select the desired location and tap Next.
2. Scan the QR code on the device. *Note: On some phones, you may need to adjust camera settings.*
3. Change the device name if desired.
4. To add additional devices to the same location, tap the QR icon and scan the next device.
5. Tap Save to add the device(s) to SmartStudio.

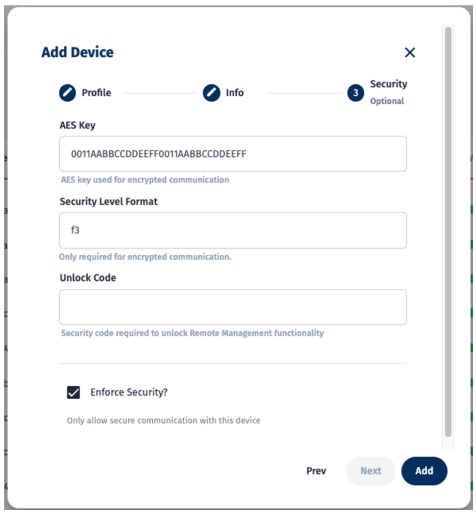


2.2.2 Manual Entry

Some devices may not be recognized by the mobile commissioning tool and in this case the device information can be entered manually. To do so, select **Manual Entry** to start the following commissioning flow:







Parameter	Description	Options
Product	Product to be added	Select from the list of known products to pre-populate device information Select Generic Device to manually enter device information
Device Profile	EnOcean EEP	Select the product EEP
Device Type	Sensor only or bidirectional communication	**Unidirectional** - Sensor only **Bidirectional** - Two way communication **Switch** - EnOcean push button
Is Activated	Enable processing of incomming data	-
Device ID	Device specific ID	-
Name	Device display name	-
Location	Device installation location	Select location from Topology
Product ID	Device family ID	-
AES Key	Device encryption key	-
Security Level Format	Device security Level	**f3** - 32 bit CMAC (Default) **ab** - 24 bit CMAC
Unlock Code	Device specific Unlock Code used for configuration	-
Enforce Security	Only allow encrypted telegrams will be processed	-

2.3 EnOcean Equipment Profiles

The EnOcean Equipment Profiles (EEP) is a standardized set of communication profiles used for interoperable communication between EnOcean-based devices. These profiles define how sensors and actuators communicate and ensure that devices from different manufacturers can work together seamlessly. The EEP covers various application areas such as space utilization, HVAC (heating, ventilation, air conditioning), energy management and more. The EEP's are standardized by the EnOcean Alliance and published here.

EnOcean SmartStudio currently support the following list of EEP's:

A5	D2	D5	F6
A5-02-05	D2-01-0B	D5-00-01	F6-02-04
A5-04-01	D2-01-0F		F6-05-01
A5-04-03	D2-14-40		
A5-06-02	D2-14-41		
A5-06-03	D2-14-52		
A5-07-01	D2-14-53		
A5-07-03	D2-14-58		
A5-08-01	D2-14-59		
A5-08-02	D2-14-5C		
A5-08-03	D2-14-5D		
A5-09-04	D2-15-00		
A5-09-09	D2-32-00		
A5-10-03	D2-32-01		
A5-12-00	D2-32-02		
A5-12-01	D2-B1-00		
A5-14-05			
A5-20-01			
A5-20-06			

More EEP's may be added on request by contacting EnOcean support.

3. Gateways

3.1 Aruba

3.1.1 Aruba Integration Overview

The EnOcean integration with Aruba allows seamless communication between EnOcean IoT devices and Aruba's network infrastructure, enabling businesses to manage IoT data using their existing Wi-Fi setup. Wireless, self-powered EnOcean sensors transmit data through an EnOcean USB stick connected to Aruba’s Wi-Fi Access Point, which forwards the data to the EnOcean SmartStudio. This integration lets companies leverage their current IT infrastructure to incorporate IoT sensors into their network, while maintaining full compliance with security standards, thanks to Aruba’s robust security features.

The EnOcean integration with Aruba supports the following Aruba network architectures:

- **Aruba Instant**
- **Aruba Controller**
- **Aruba Central (AOS 8)**
- **Aruba Central (AOS 10)**

Prerequisites

The following prerequisites are required for EnOcean integration with Aruba:

Aruba Access Point (AP) with USB port

Ensure the Aruba AP meets the necessary energy requirements to properly power the USB port.

AP model	USB port (5W)	IPM feature	802.3af (class 3)	802.3at (class 4)	802.3bt (class 5+)	DC power	AC power
AP-303	no	no	no USB port	no USB port	no USB port	no USB port	not supported
AP-303P	no	no	no USB port	no USB port	no USB port	no USB port	not supported
AP-304/305	yes	yes	disabled	OK	OK	OK	not supported
AP-314/315	yes	yes	disabled	OK	OK	OK	not supported
AP-324/325	yes	no	disabled	OK	OK	OK	not supported
AP-334/335	yes	yes	disabled	disabled	disabled	OK	not supported
AP-344/345	yes	yes	disabled	disabled	disabled	OK	not supported
AP-504/505	yes	yes	disabled	OK	OK	OK	not supported
AP-514/515	yes	yes	disabled	OK	OK	OK	not supported
AP-534/535	yes	yes	not supported	disabled	OK	OK	not supported
AP-555	yes	yes	not supported	disabled	OK	OK	not supported
AP-203H	no	no	no USB port	no USB port	no USB port	not supported	not supported
AP-303H	yes	yes	disabled	disabled when P	disabled when P	OK	not supported
AP-503H	no	no	no USB port	no USB port	no USB port	no USB port	not supported
AP-505H	yes	yes	disabled	disabled when P	OK	OK	not supported
AP-203R	yes	no	not supported	not supported	not supported	not supported	OK

EnOcean USB dongle

Current Versions:

USB revision	Aruba OS 8	Aruba OS 10
USB 300 DE-13	AOS 8.10.0.13 / AOS 8.12.0.1 +	Supported*
USB 500U DB-06	AOS 8.10.0.0 +	Supported*
USB 400J DD-11	AOS 8.10.0.13 / AOS 8.12.0.1 +	Supported*
USB 500J DB-02	AOS 8.10.0.0 +	Supported*

Previous Versions:

USB revision	Aruba OS 8	Aruba OS 10
USB 300 DE-12	AOS 8.10.0.0 +	Supported*
USB 300 DC	AOS 8.10.0.0 +	Supported*
USB 300 DD	AOS 8.10.0.0 +	Not supported
USB 500U DB-05	AOS 8.10.0.0 +	Supported*
USB 500U DA	AOS 8.10.0.0 +	Supported*
USB 500J DA-01	AOS 8.10.0.0 +	Supported*
USB 400J DD-09	AOS 8.10.0.0 +	Supported*
USB 400J DD-10	AOS 8.10.0.13 / AOS 8.12.0.1 +	Supported*

Note

- ◦ Minimum AOS 10.8 is required for running the EnOcean SmartStudio App in AP as connector mode.

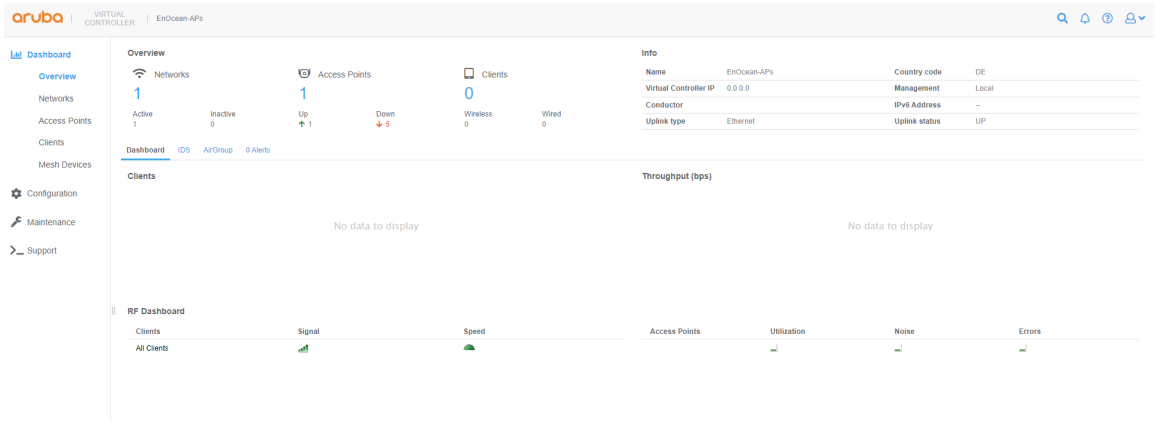
3.1.2 Configuration using Aruba Instant

Aruba Instant

Aruba Instant is a controllerless Wi-Fi solution designed for easy deployment and management, ideal for small to medium-sized businesses. Each Aruba Instant Access Point (AP) works independently, automatically creating a self-configuring, scalable network without the need for a physical controller. The network can be managed through a local web interface or via **Aruba Central** for cloud-based control.

Step 1: Connect to Instant

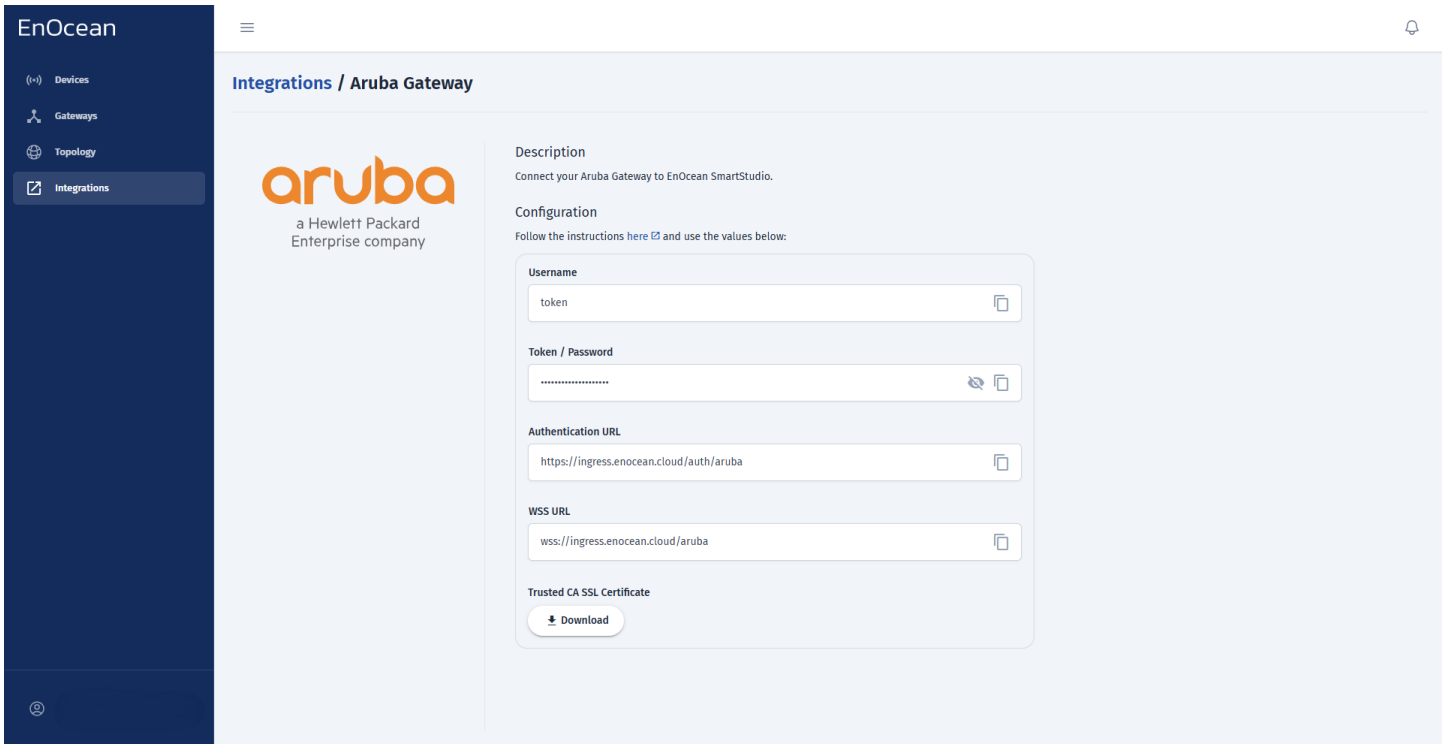
Log into the router’s web-based management page for Aruba instant.



Step 2: Installing Trusted CA Certificates

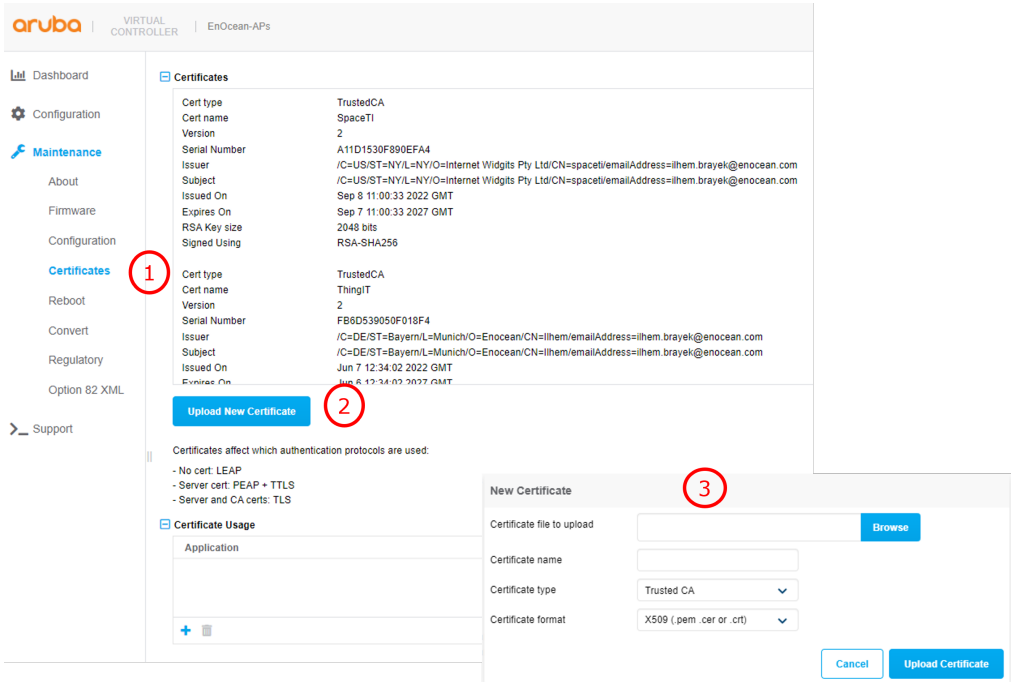
Note

The **Username**, **Password**, **Authentication URL**, **WSS URL** and the **Trusted CA Certificate** can be obtained from SmartStudio -> **Integrations** section -> **Aruba Gateway** like shown below. It is also recommended to generate a random 10-character string to use as your client ID.



- Upload your .pem certificate file From **Maintenance** -> **Certificates** -> **Upload new certificate**.

  select **Trusted CA** as **certificate type** and **x509(.pem .cer or .crt)** as **Certificate format**.

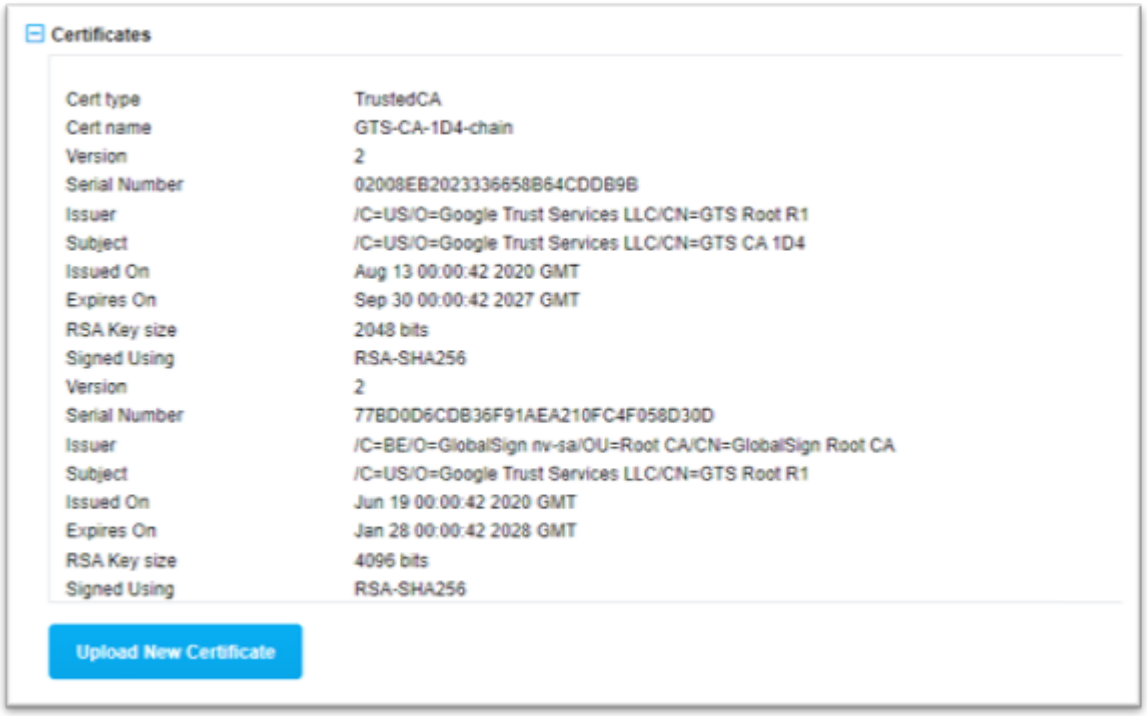


- Browse to certificate.
- Click **Upload Certificate** to save your settings.

Verify Certificate upload.

Verify the certificate is shown on the certificate list:

- Using the UI:

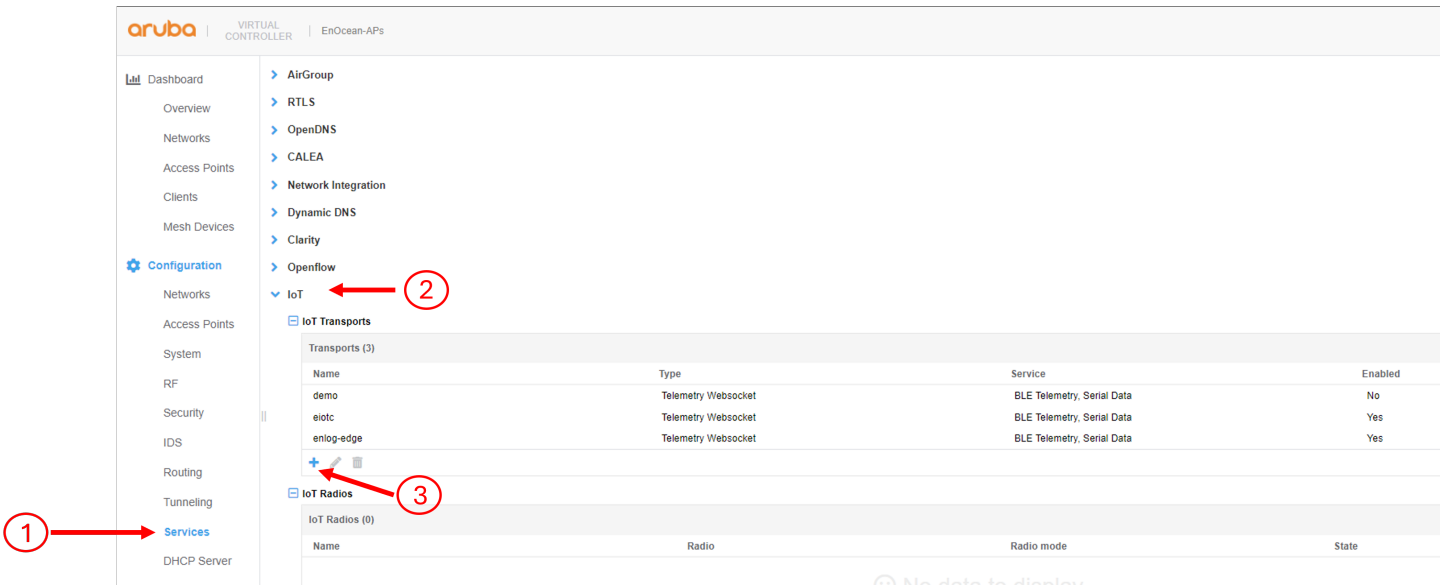


- Using the CLI:

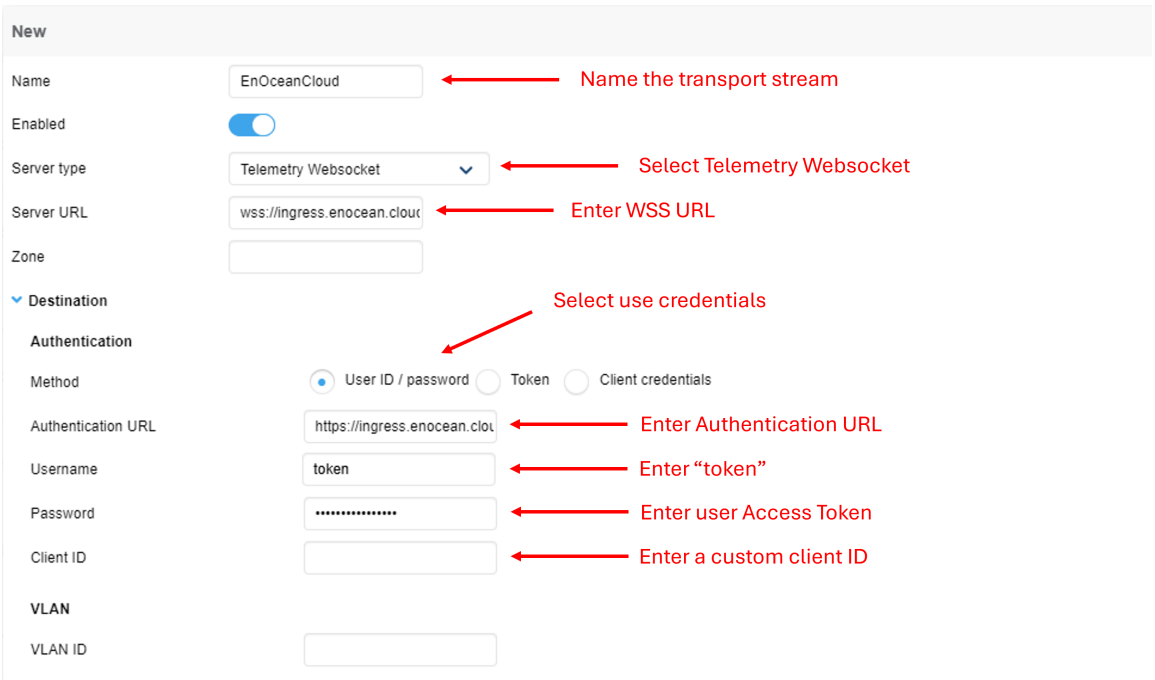
  ```powershell (IAP)# show cert all

  Cert type:TrustedCA Cert name:GTS-CA-1D4-chain Version :2 Serial Number :02008EB2023336658B64CDDB9B Issuer :/C=US/O=Google Trust Services LLC/CN=

  ```

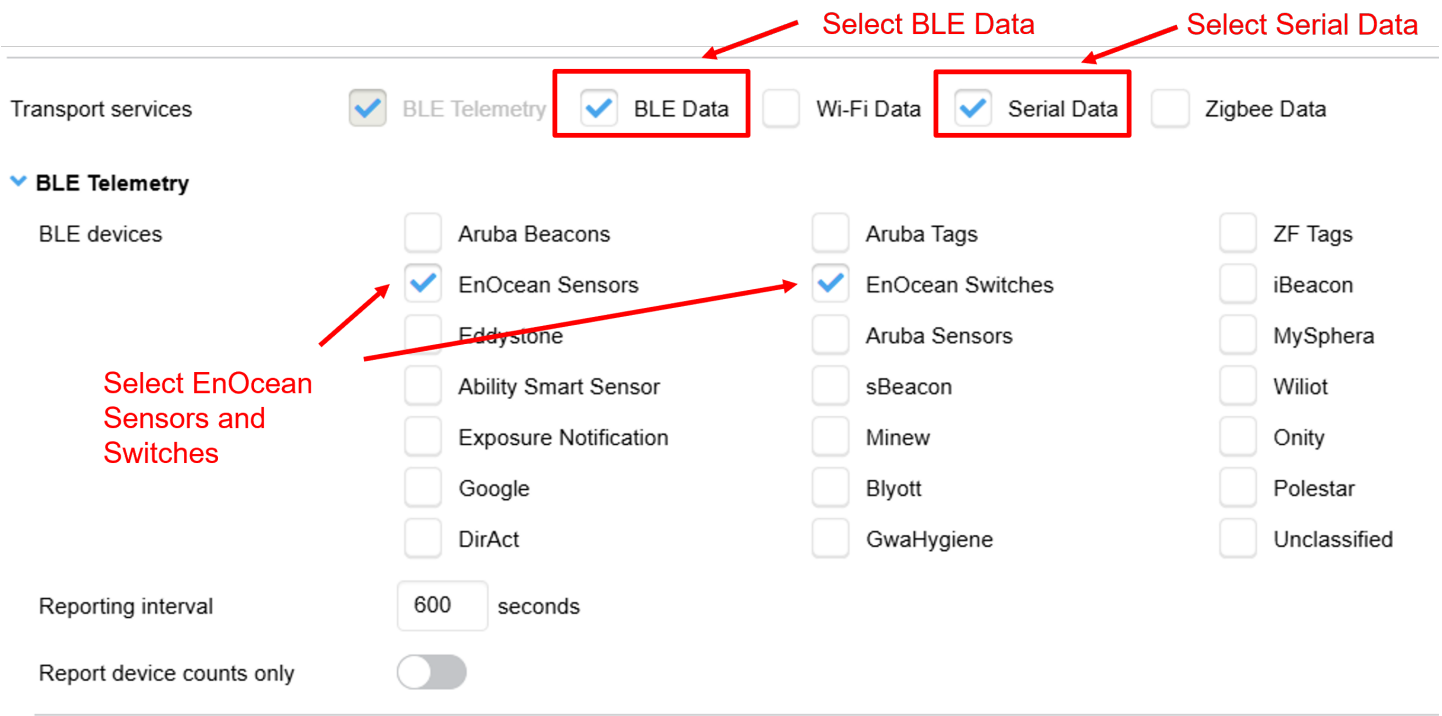
Step 3: IoT transport configuration

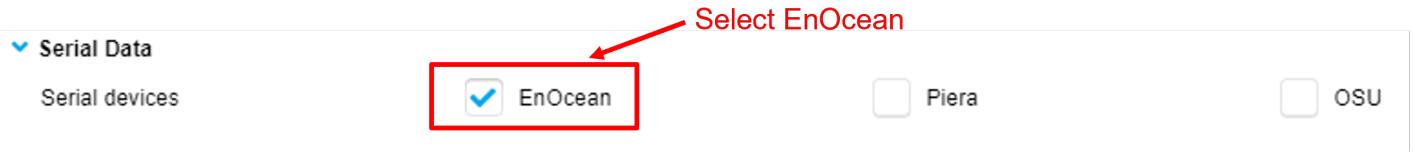
- In Aruba Instant select **Configuration** -> **Services** -> **IoT** -> **IoT transports** then add a new transport using the **+** icon:



- Enter the following information in the IoT transport popup:

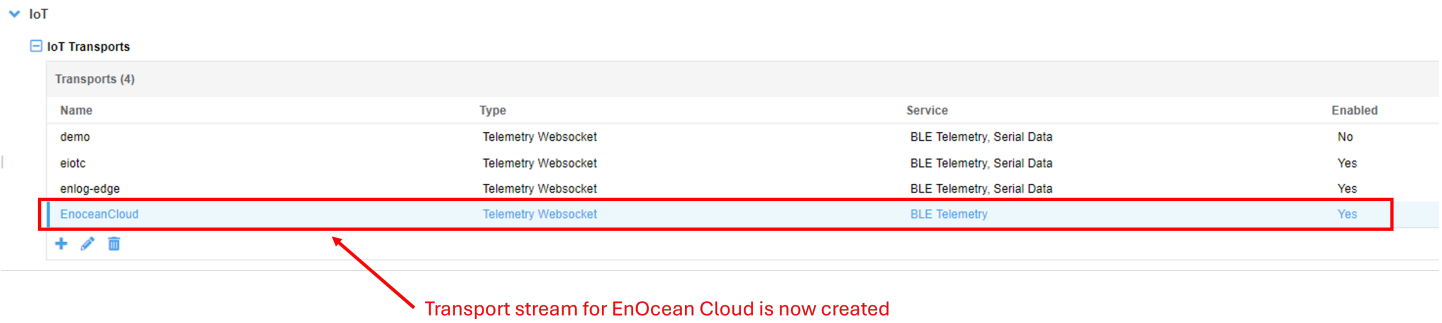






Click **OK** then **Save** to save you configuration.

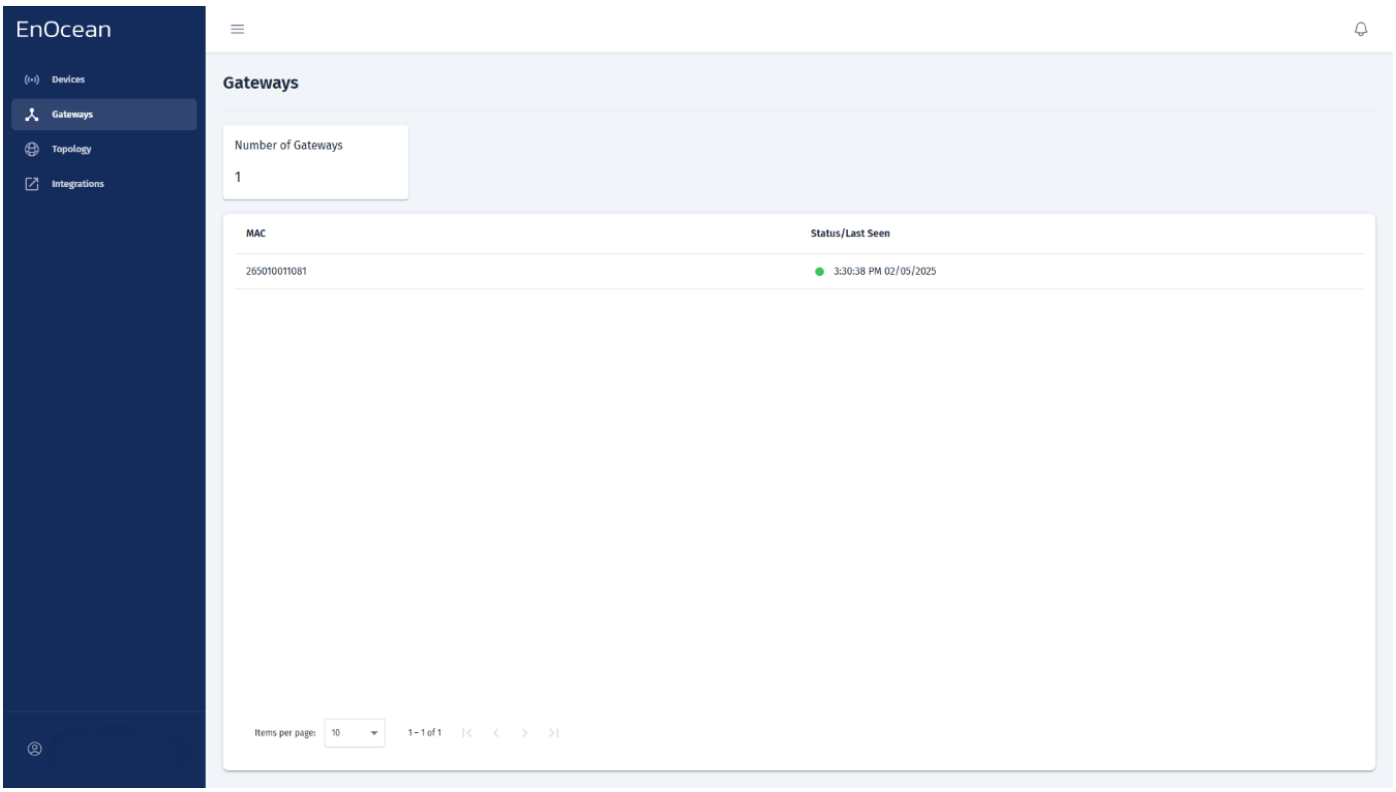
- Check that you transport stream is enabled:



Step 4: Verify that your Gateway is connected

You can check the gateway status directly from the Gateways tab in the EnOcean SmartStudio dashboard:

1. Log in to the EnOcean SmartStudio web interface.
2. Navigate to the Gateways tab.
3. Locate your gateway in the list and check its connection status.



Alternatively, you can verify the gateway status using the API:

1. Login to EnOcean SmartStudio API.
2. Use the GET /v0/gateways endpoint to check the connection status.

3.1.3 Configuration using Aruba Controller

Aruba Controller

An **Aruba Controller Network** is a centralized architecture that provides unified management and control over a network of Aruba Access Points (APs), ensuring seamless, secure, and scalable connectivity. In this setup, an Aruba controller acts as the aggregator of the network, managing traffic routing, user authentication, RF optimization, and security policies across all connected APs.

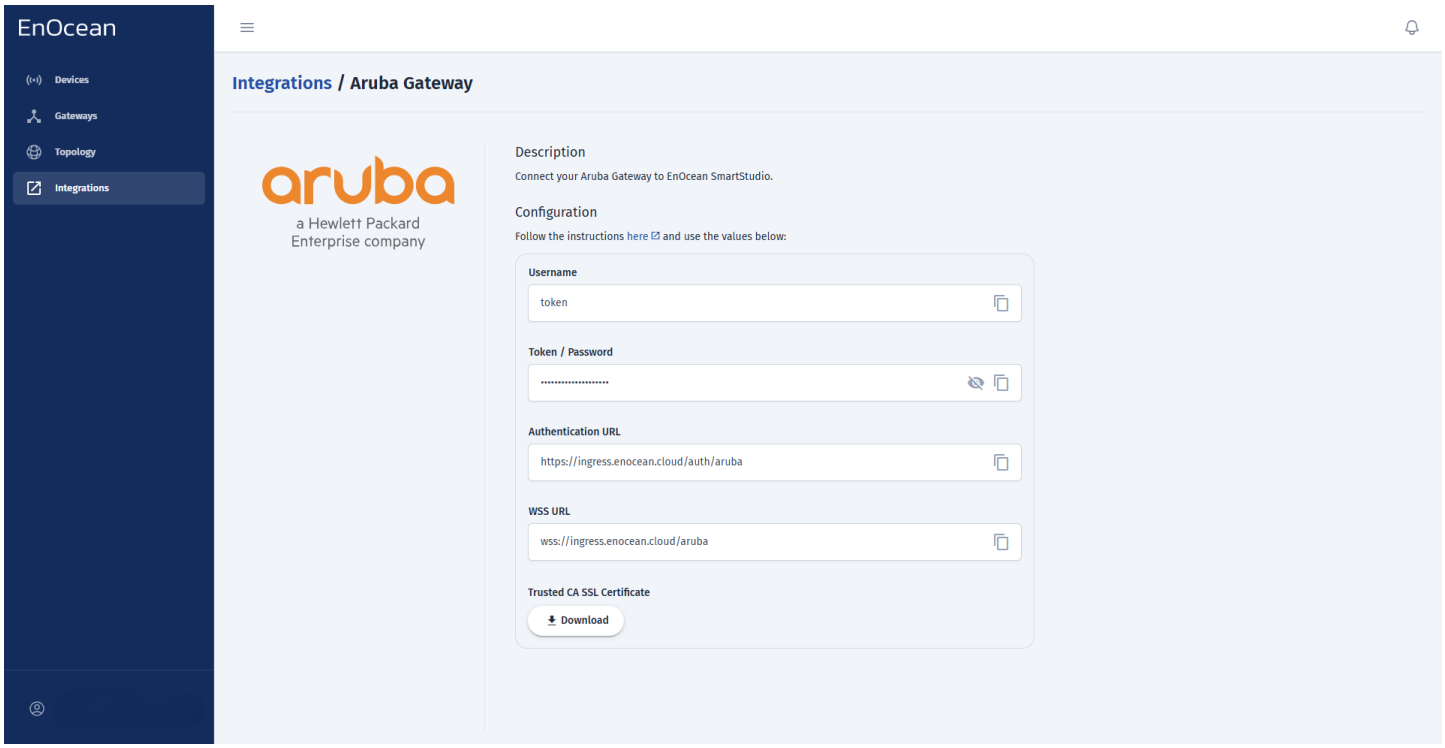
Step 1: Connect to ArubaOS

Log into the web-based management page for ArubaOS.

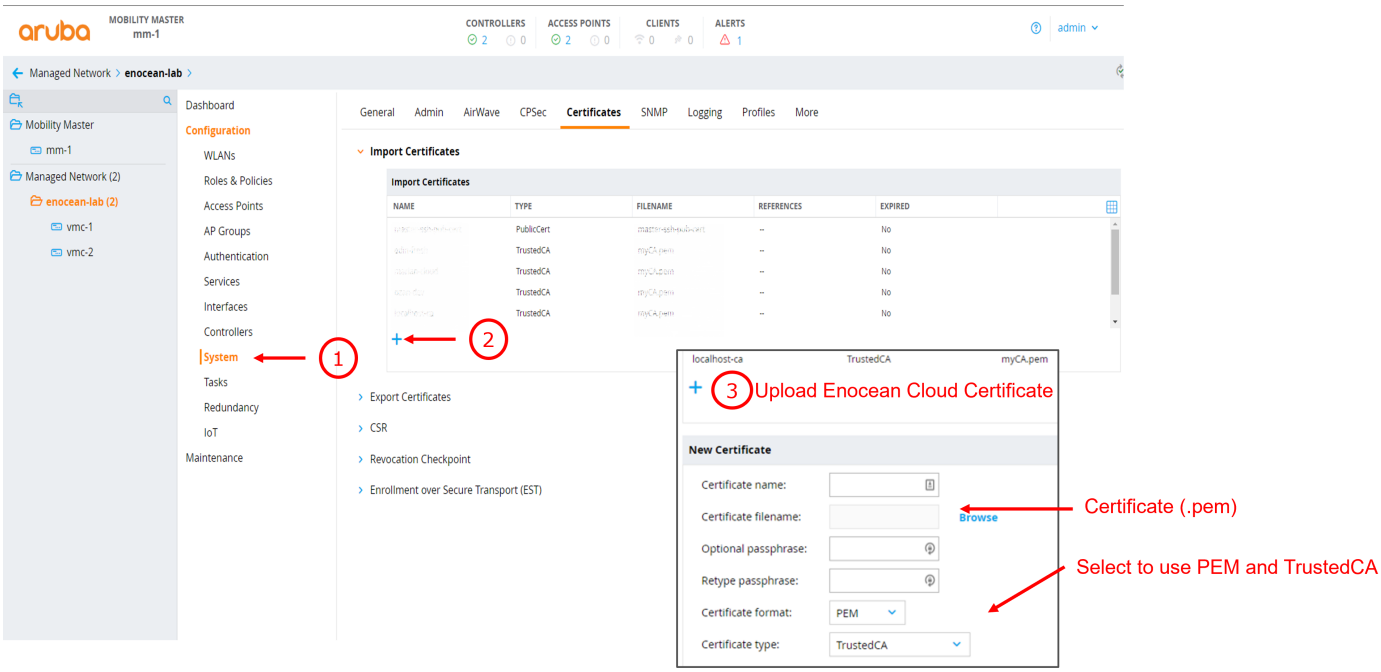
Step 2: Installing Trusted CA Certificates

Note

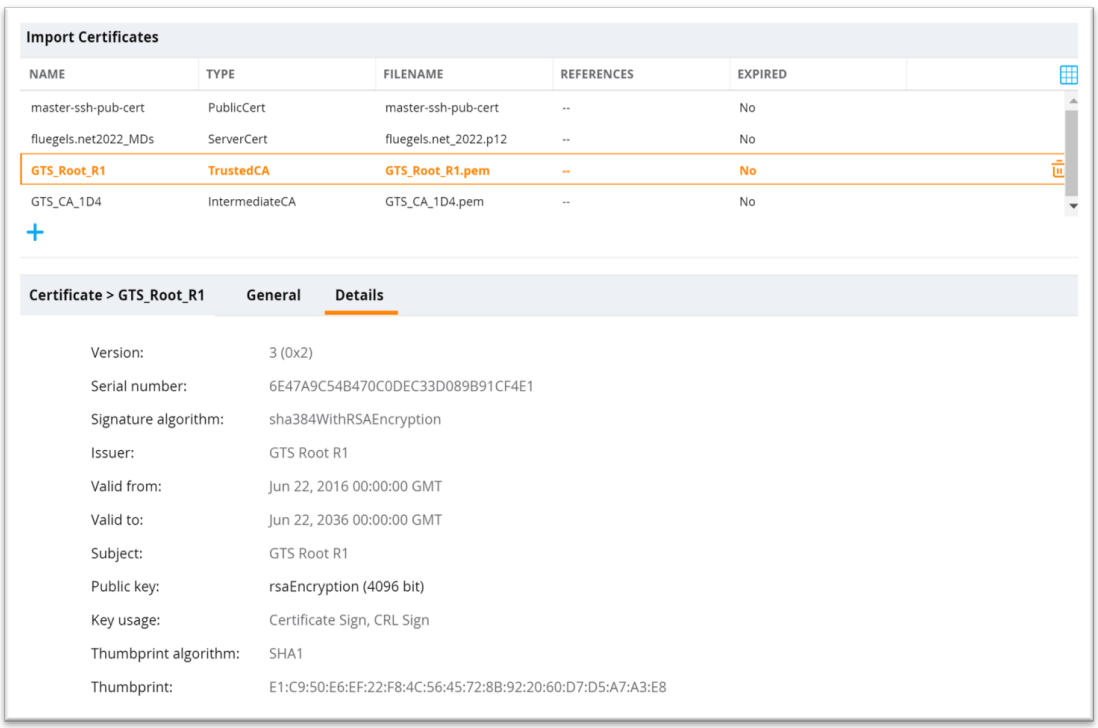
The **Username**, **Password**, **Authentication URL**, **WSS URL** and the **Trusted CA Certificate** can be obtained from SmartStudio -> **Integrations** section -> **Aruba Gateway** like shown below. It is also recommended to generate a random 10-character string to use as your client ID.



- In your managed network tab select **Configuration** -> **System** -> **Certificates** Click the **+** to upload a new certificate. select Trusted CA as **certificate type** and **x509(.pem .cer or .crt)** as Certificate format.



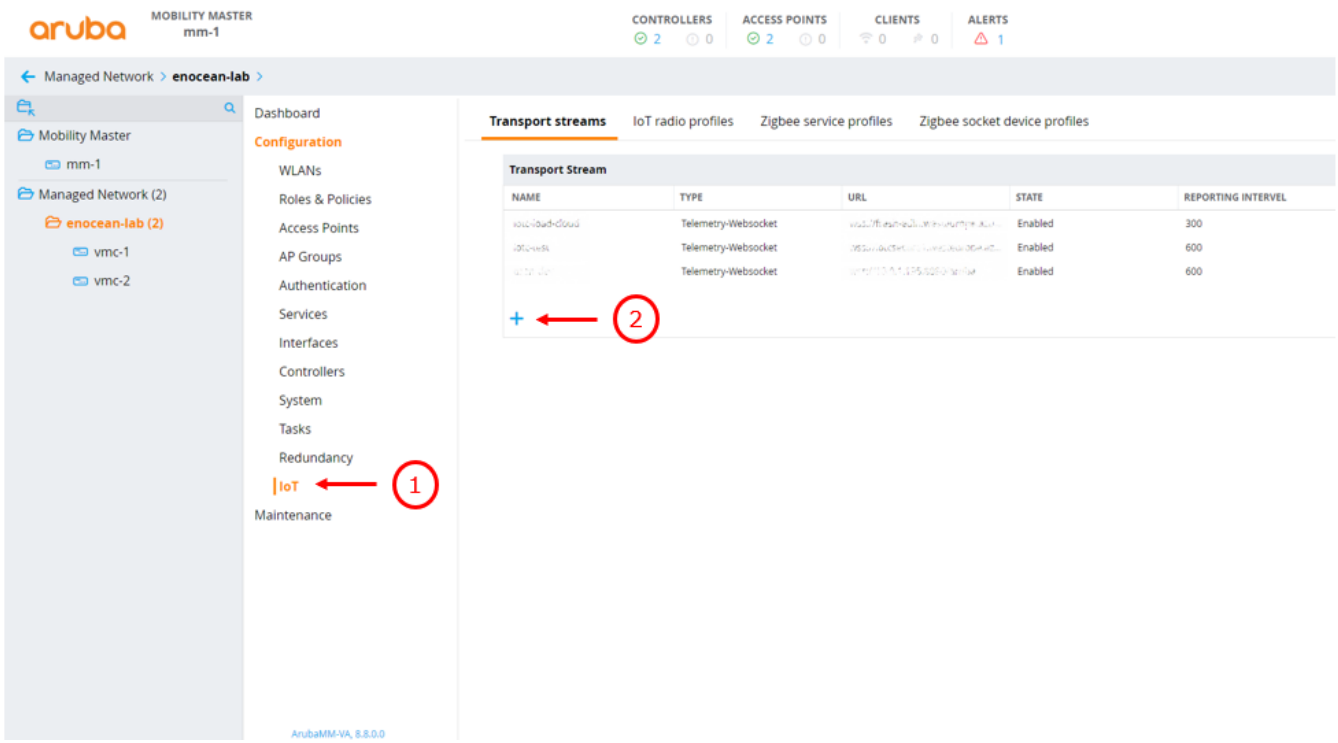
- Click **Upload Certificate** to save your settings.
- Verify that the certificate is shown on the certificates list.

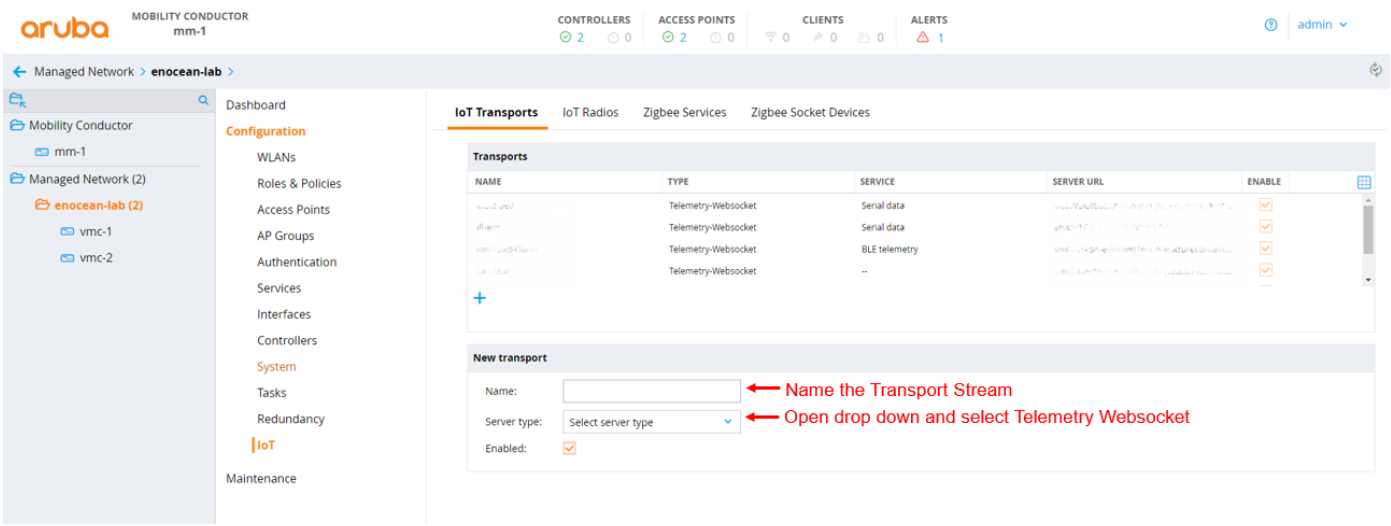


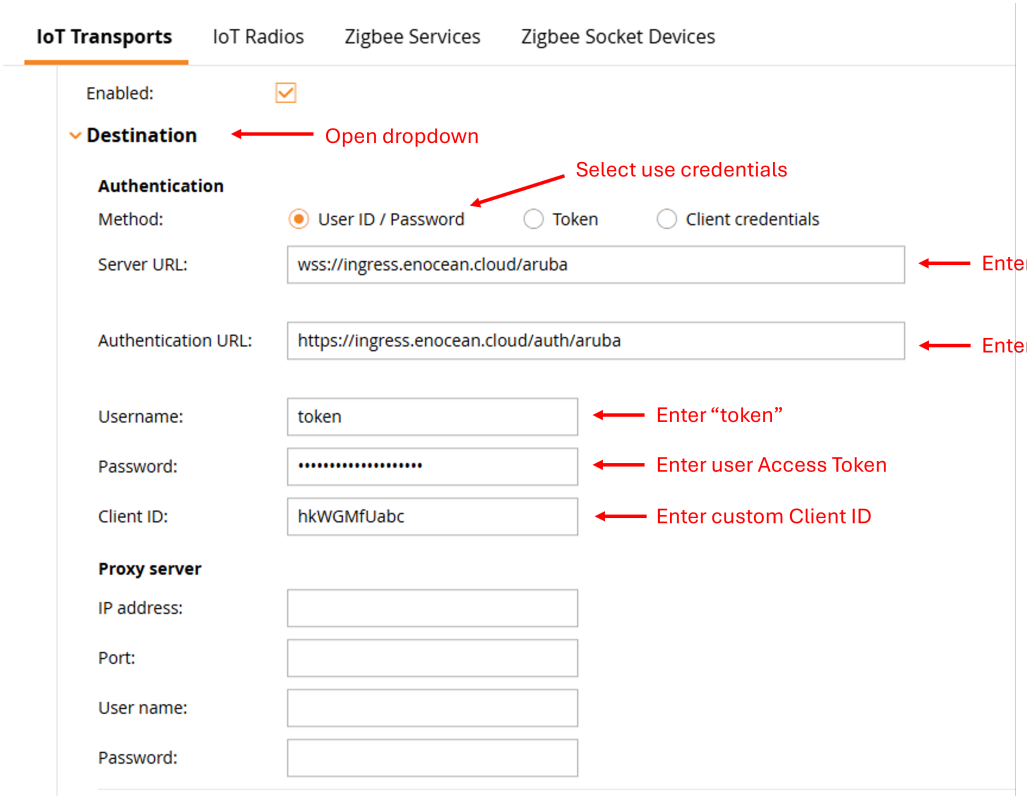
Step 3: IoT transport configuration

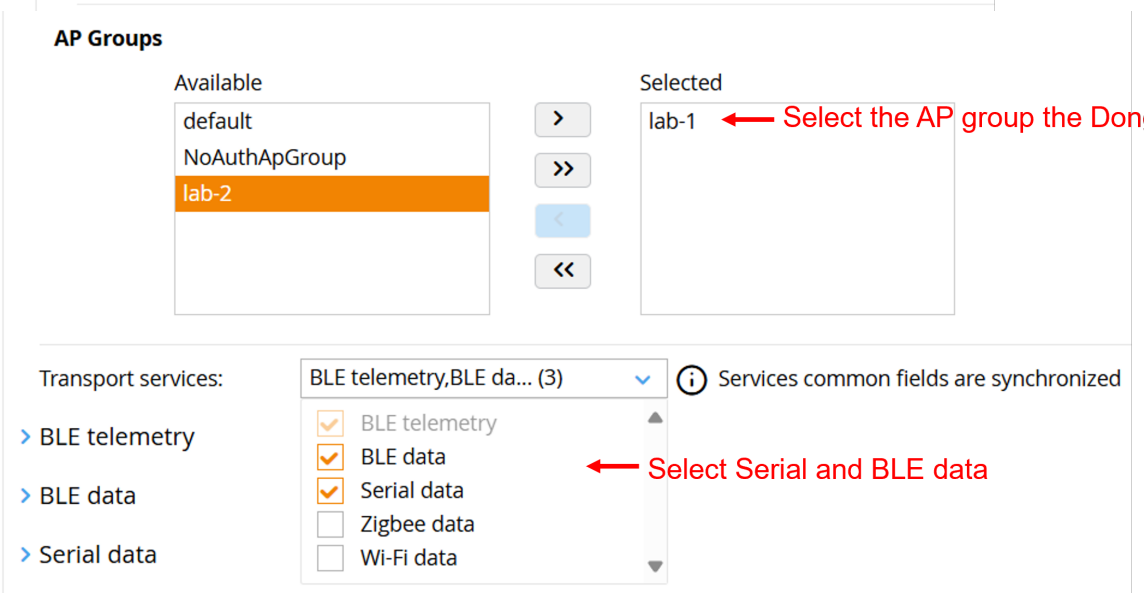
- In Aruba managed network select **Configuration -> IoT -> Transport streams** then add a new transport using the **+** icon.

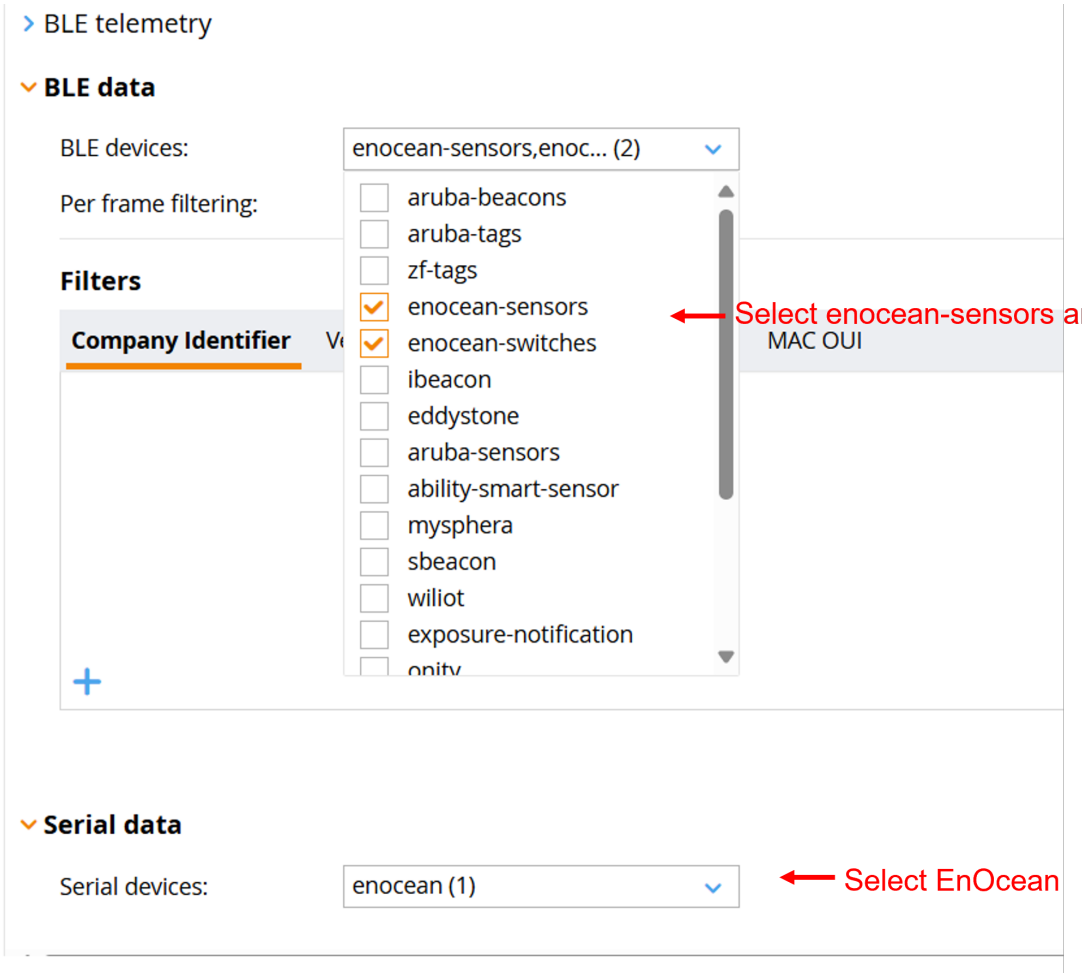
  Enter the following information in the IoT transport tab:



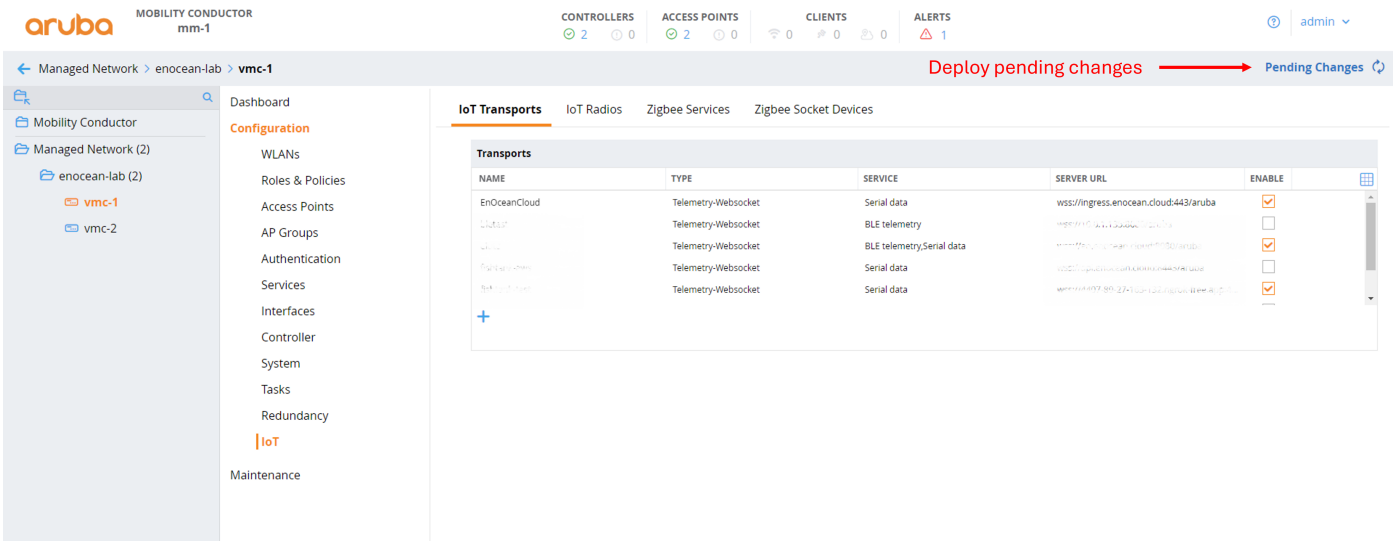








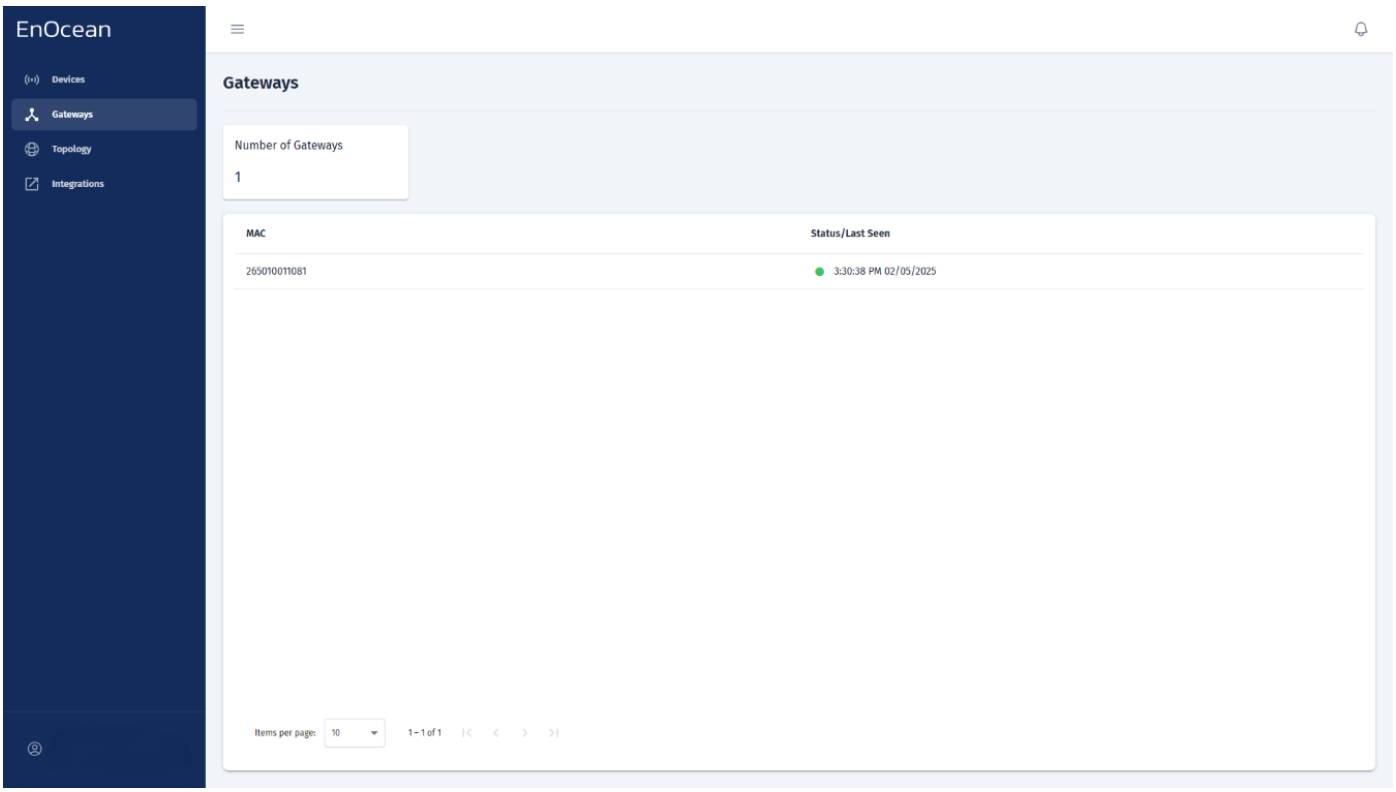
- Check that you transport stream is enabled:



Step 4: Verify that your Gateway is connected

You can check the gateway status directly from the Gateways tab in the EnOcean SmartStudio dashboard:

1. Log in to the EnOcean SmartStudio web interface.
2. Navigate to the Gateways tab.
3. Locate your gateway in the list and check its connection status.



Alternatively, you can verify the gateway status using the API:

1. Login to EnOcean SmartStudio API.
2. Use the GET /v0/gateways endpoint to check the connection status.

3.1.4 Configuration using Aruba Central

Aruba Central

Aruba Central is a cloud-based platform that provides unified management and control over a network of Aruba Access Points (APs), ensuring seamless, secure, and scalable connectivity. Designed for scalability and ease of use, Aruba Central allows IT teams to manage multiple locations and thousands of devices from a single interface.

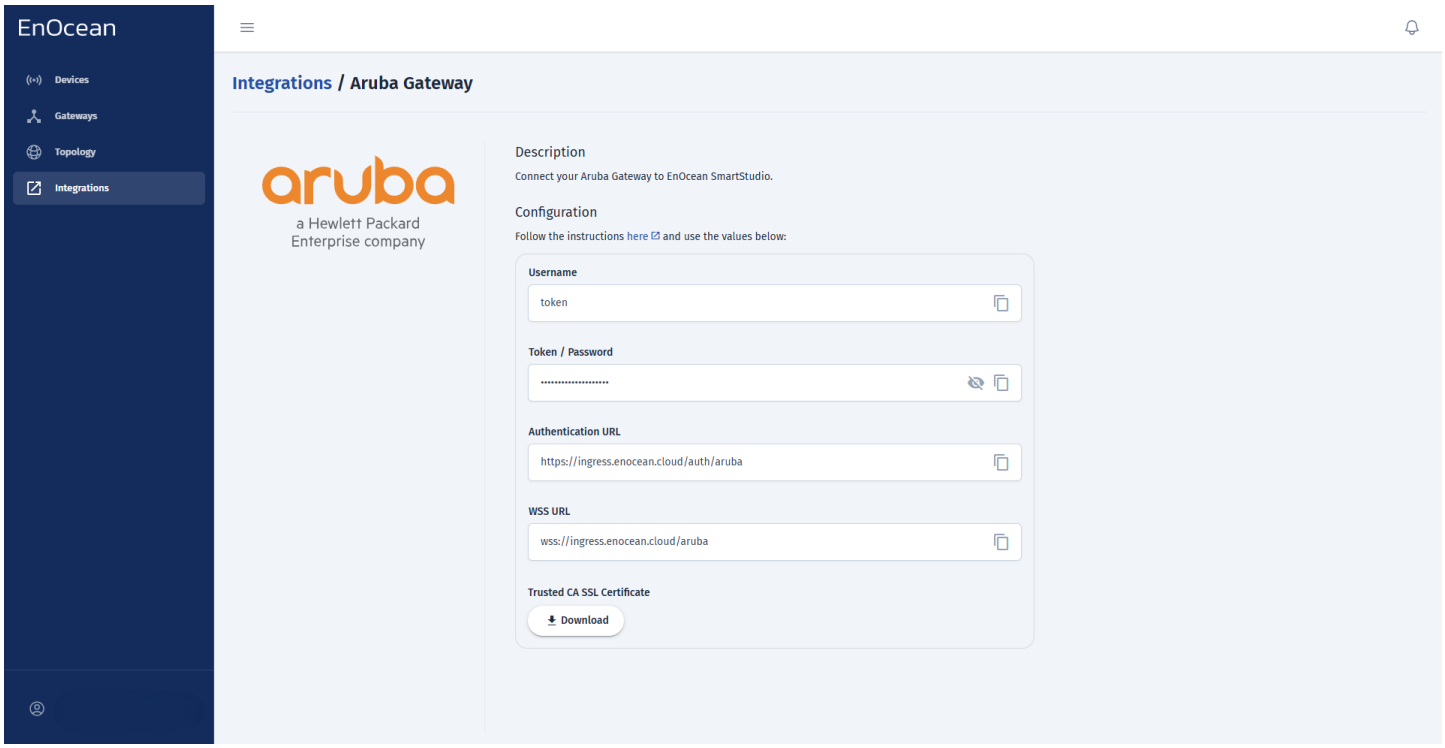
Step 1: Connect to Aruba Central

Log into the web-based management page for Aruba Central.

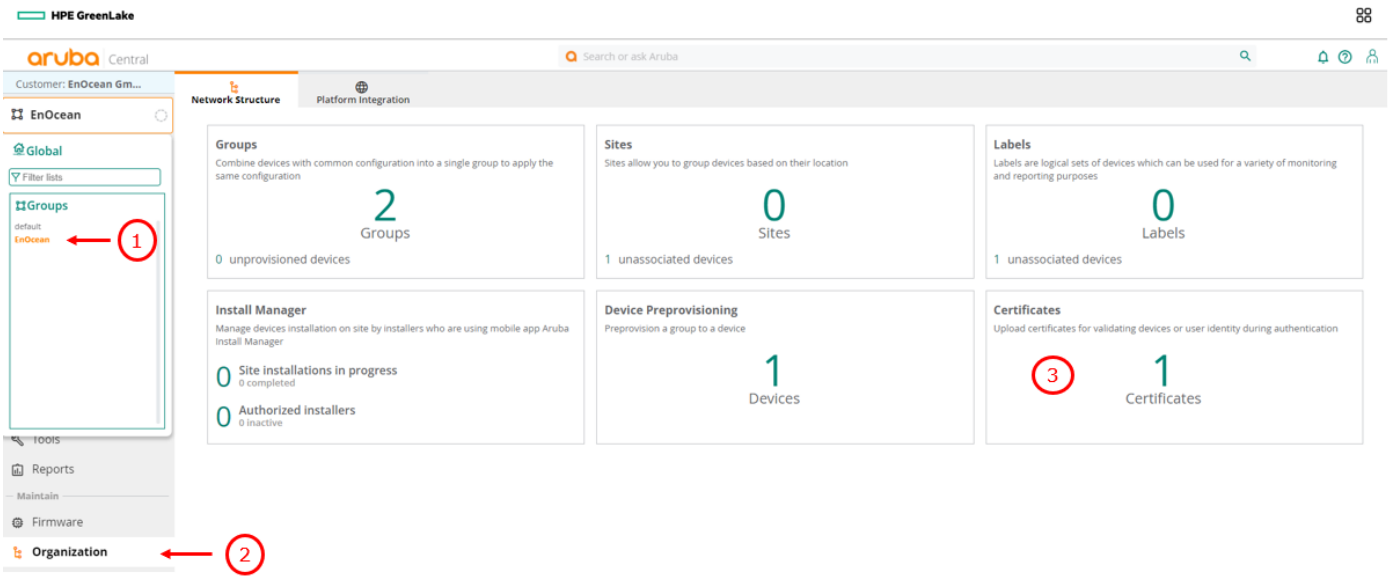
Step 2: Installing Trusted CA Certificates

Note

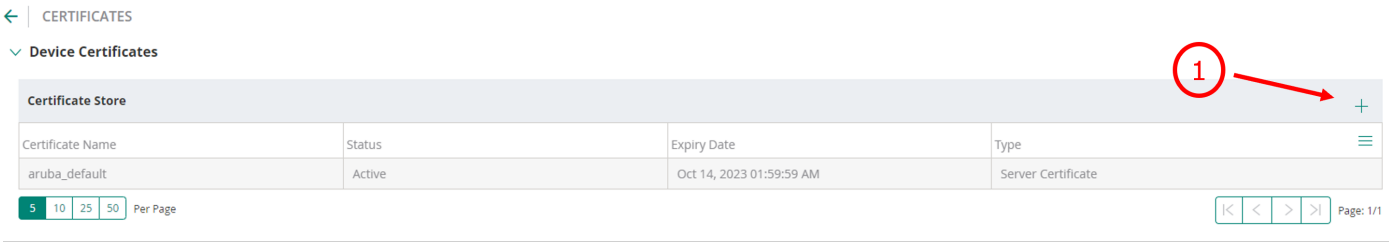
The **Username**, **Password**, **Authentication URL**, **WSS URL** and the **Trusted CA Certificate** can be obtained from SmartStudio -> **Integrations** section -> **Aruba Gateway** like shown below. It is also recommended to generate a random 10-character string to use as your client ID.

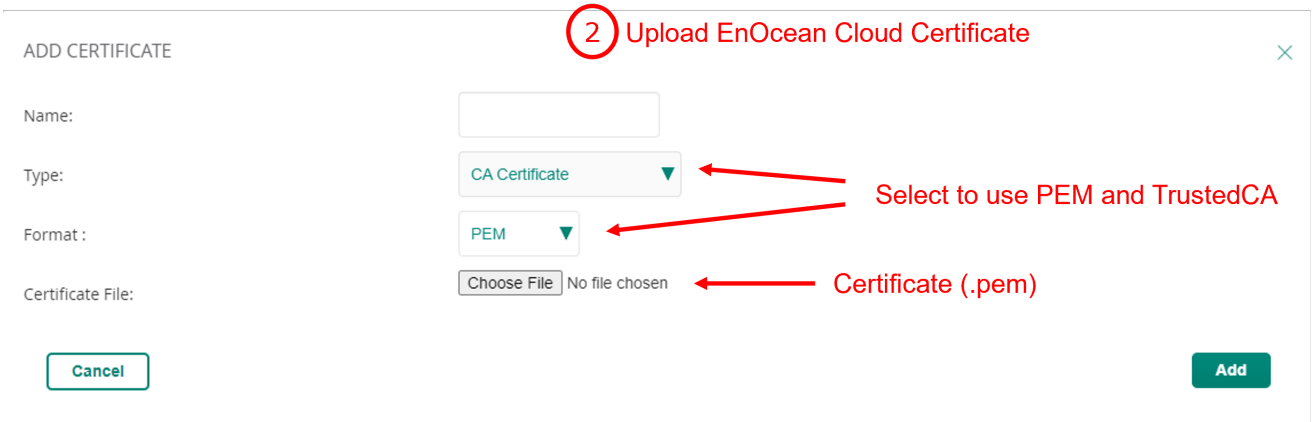


- Upload your .pem certificate file From **Your group -> Organization -> Certificates**.

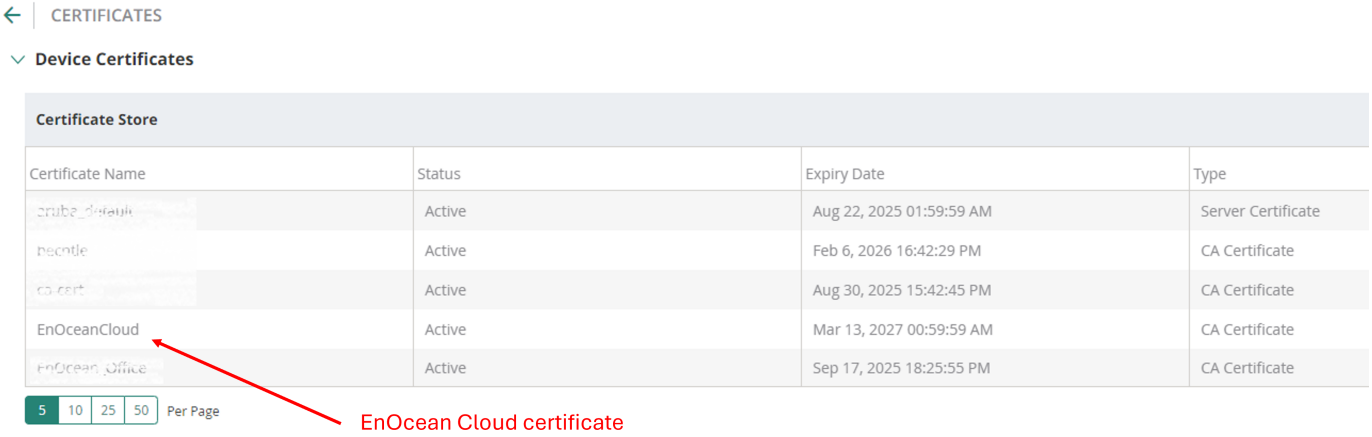


- Select **CA certificate** as **certificate type** and **PEM** as **Certificate format** then choose your certificate .pem file.



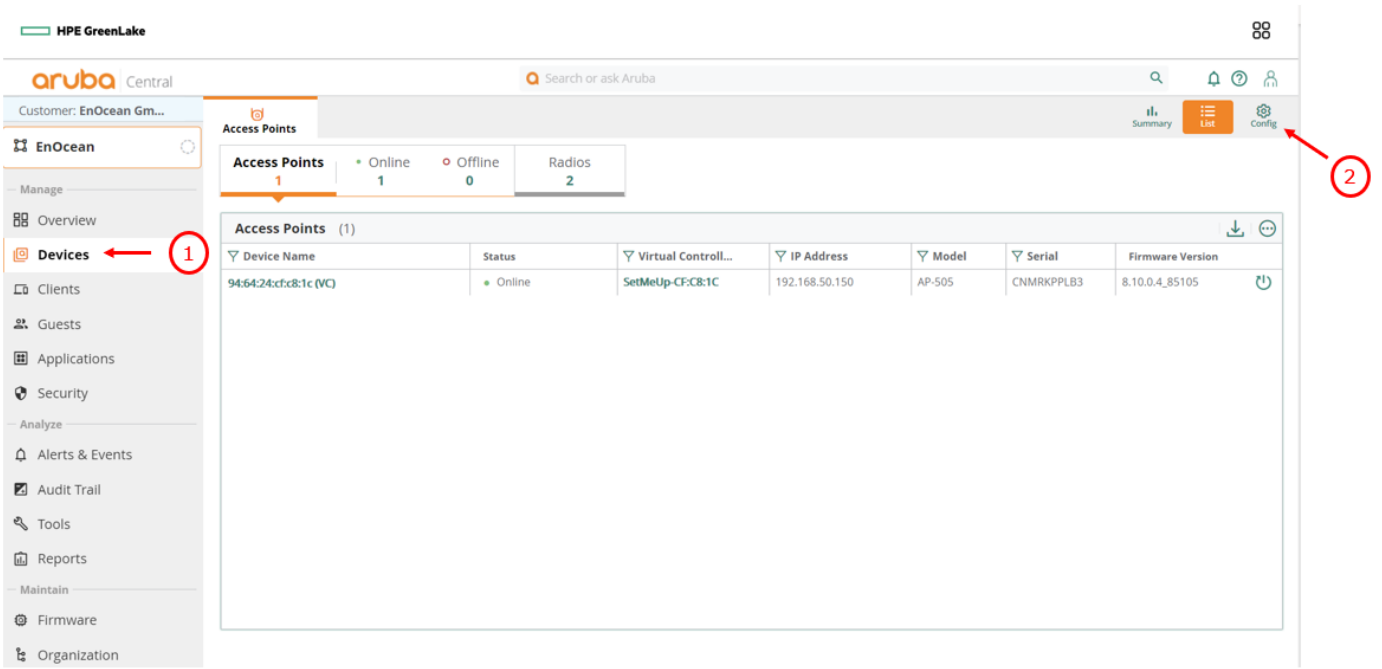


- Click **Add** to save your settings.
- Verify the certificate is shown on the certificate list.

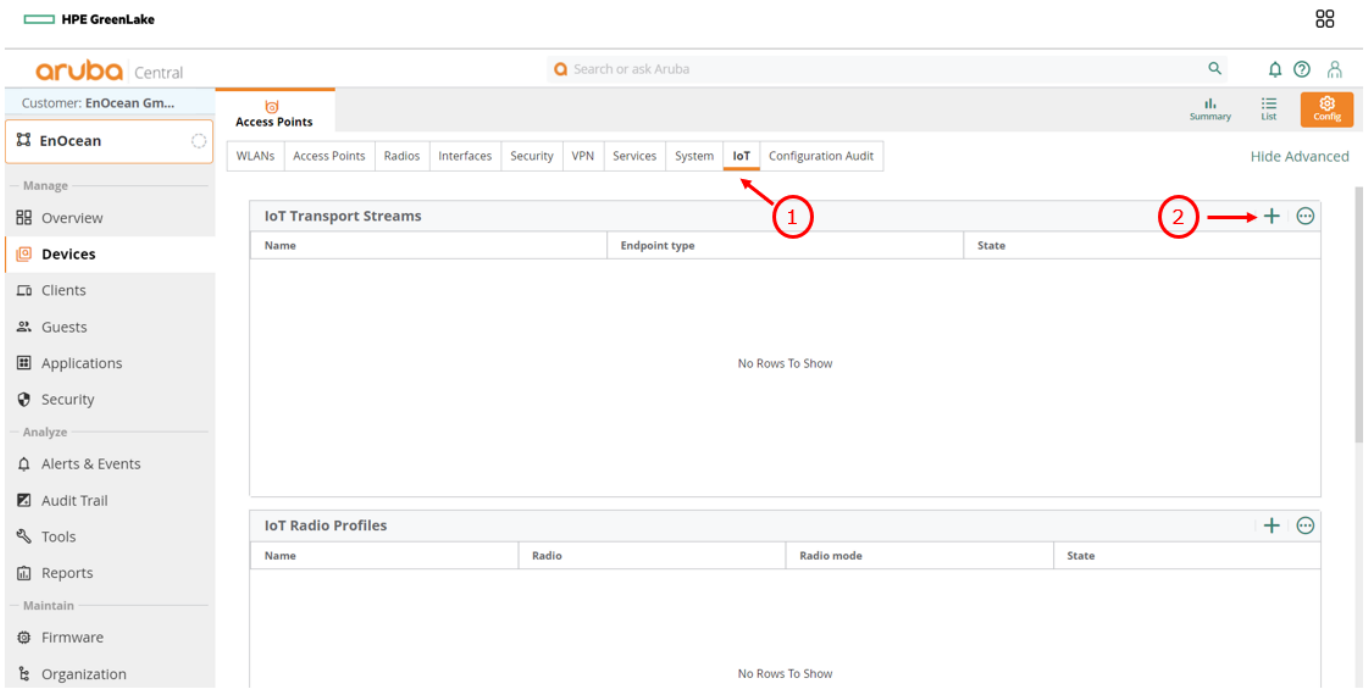


Step 3: IoT transport configuration

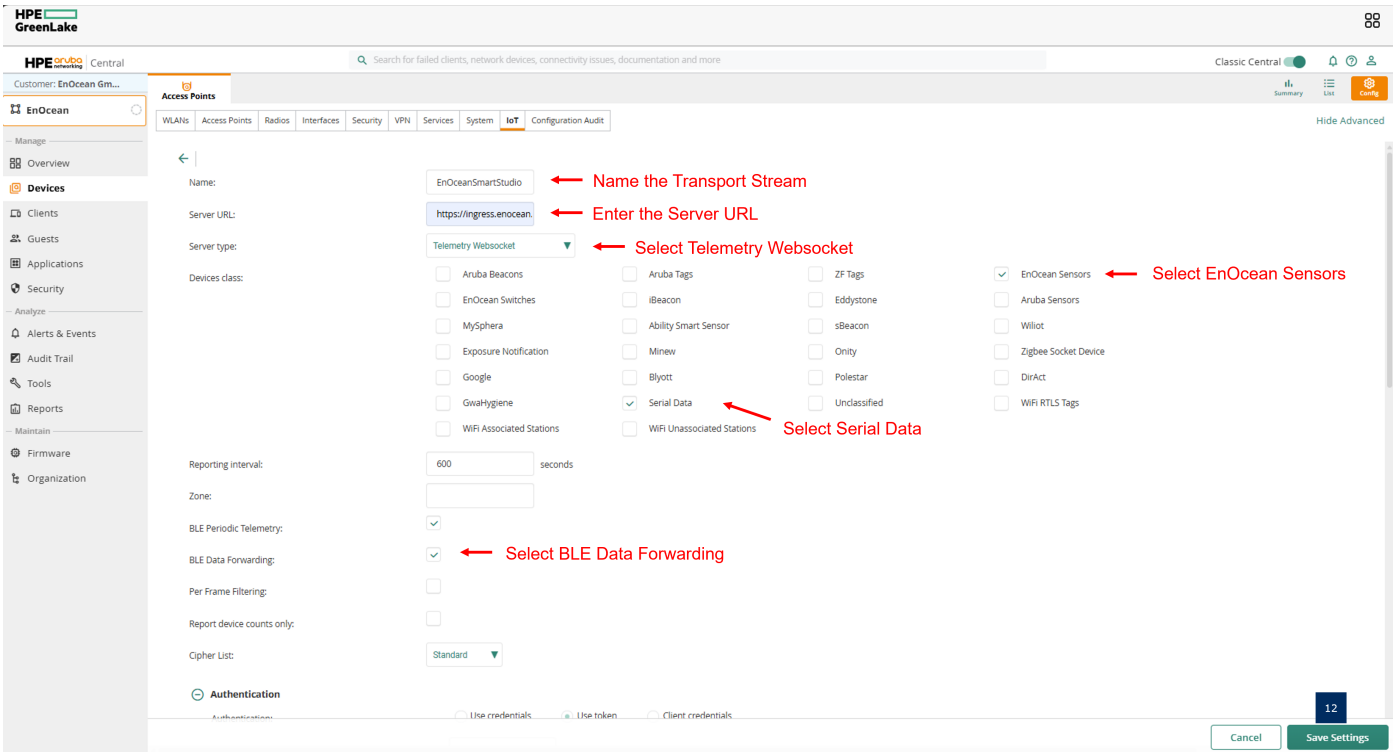
- In your Aruba Central group select **Devices** -> **Config**:

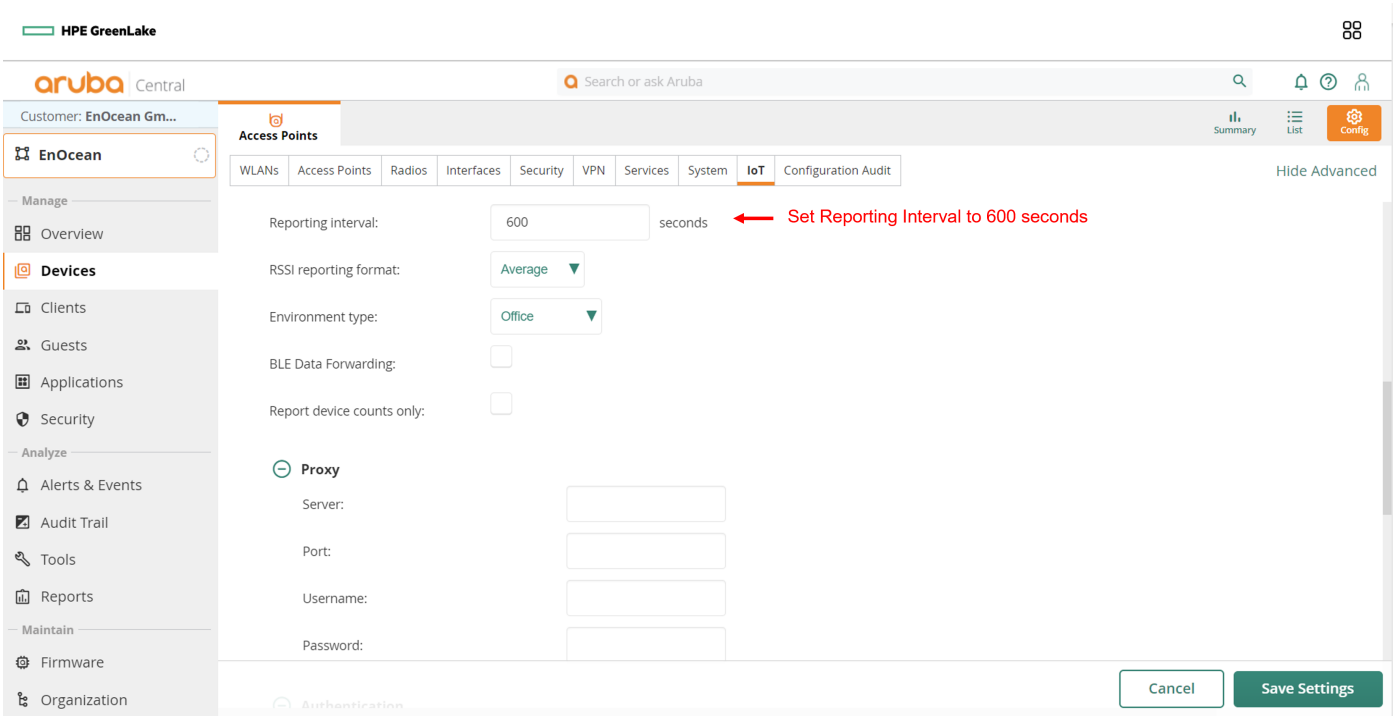


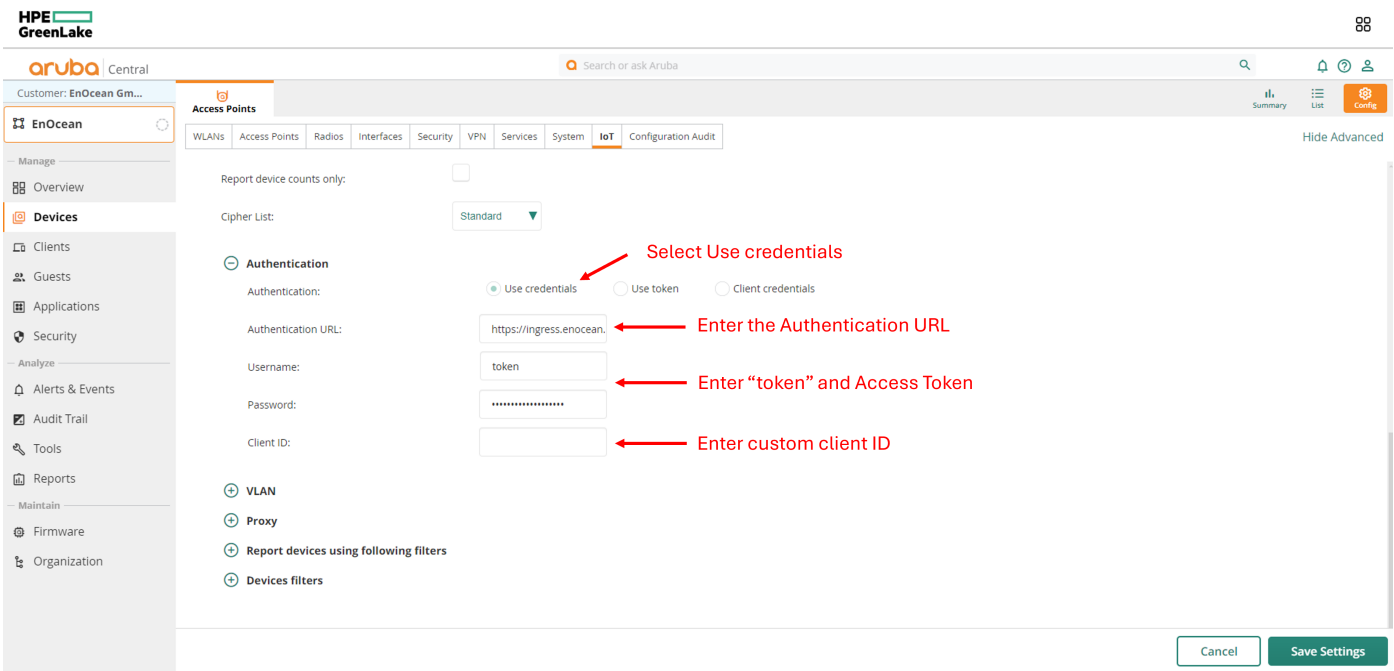
- select **IoT** then add a new IoT transport stream using the **+** icon:



- Enter the following information in the IoT transport tab:

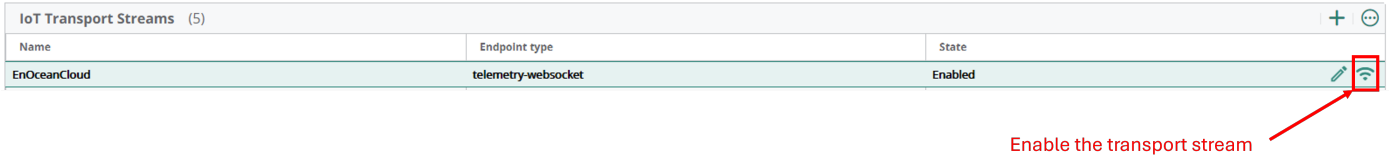






Click **Save settings** to save you configuration.

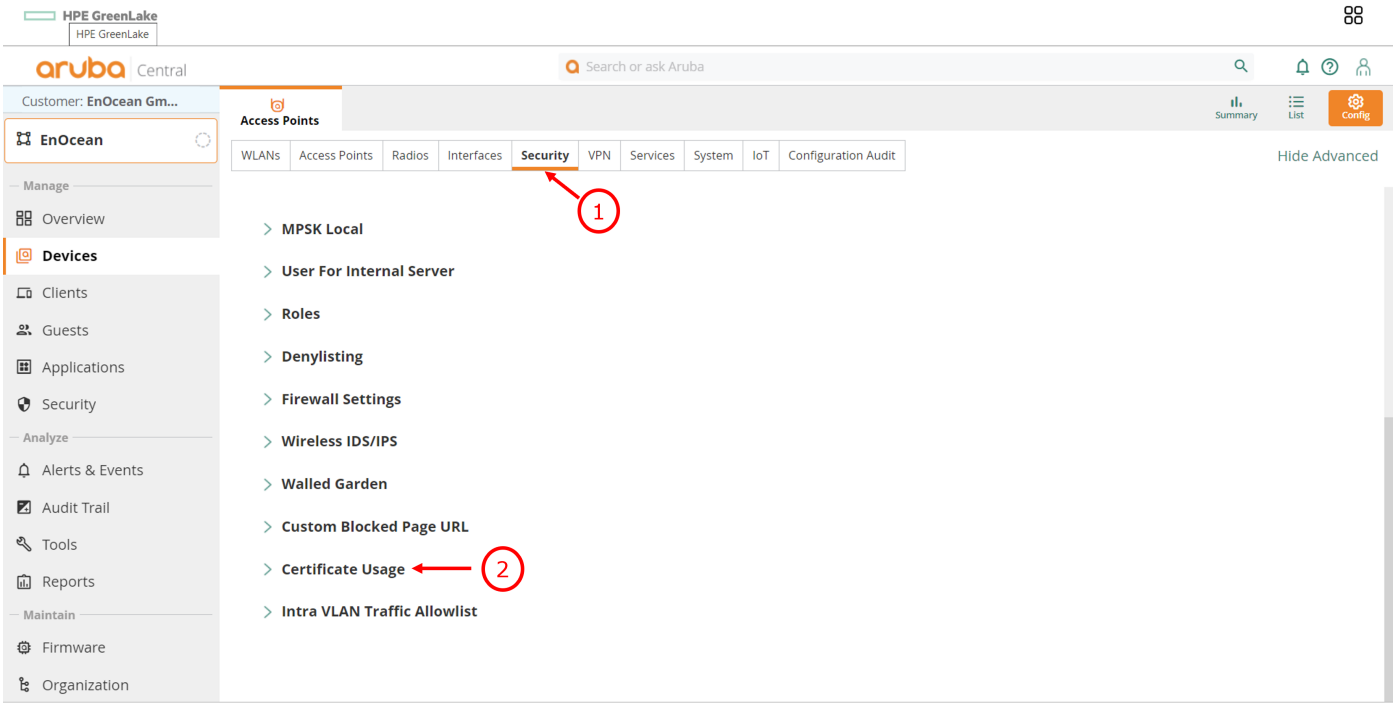
- Check that you transport stream is enabled:



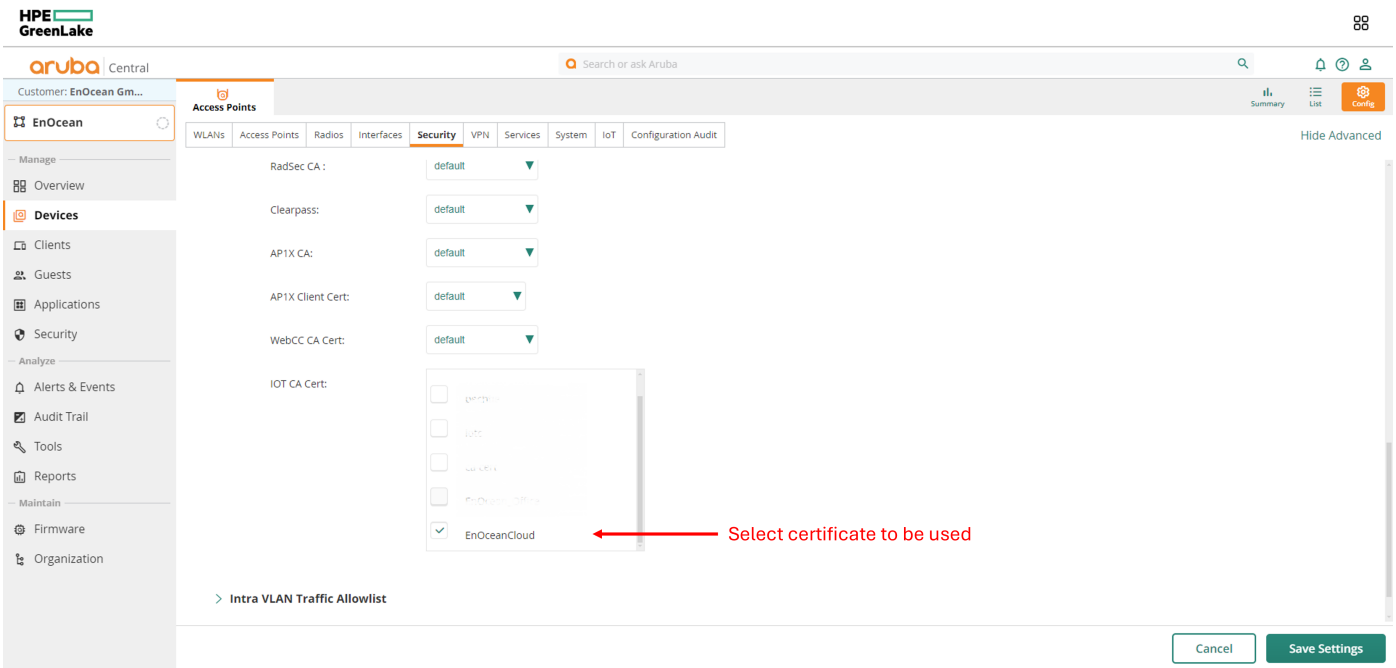
Enable the transport stream

Step 4: Activating the certificate

- Under **Access Points** click **Security** then **Certificate Usage**.



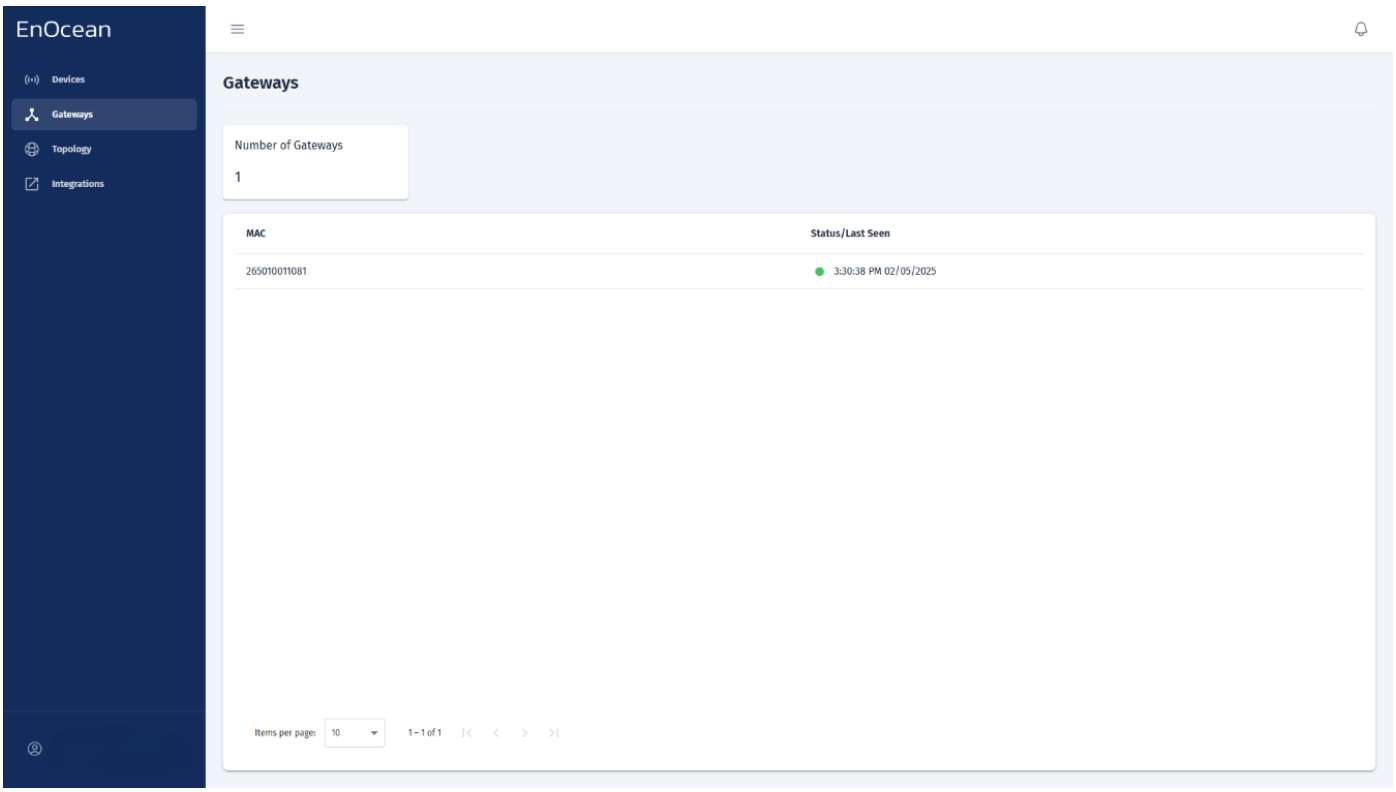
- Tick your certificate then click **Save Settings**.



Step 5: Verify that your Gateway is connected

You can check the gateway status directly from the Gateways tab in the EnOcean SmartStudio dashboard:

1. Log in to the EnOcean SmartStudio web interface.
2. Navigate to the Gateways tab.
3. Locate your gateway in the list and check its connection status.



Alternatively, you can verify the gateway status using the API:

1. Login to EnOcean SmartStudio API.
2. Use the GET /v0/gateways endpoint to check the connection status.

3.1.5 Configuration using Aruba Central

Aruba Central

Aruba Central is a cloud-based platform that provides unified management and control over a network of Aruba Access Points (APs), ensuring seamless, secure, and scalable connectivity. Designed for scalability and ease of use, Aruba Central allows IT teams to manage multiple locations and thousands of devices from a single interface.

This document describes how to integrate Aruba Central (AOS 10) with EnOcean SmartStudio using one of the following supported deployment options:

- **Option A**: VM-based Aruba IoT Collector
- **Option B**: Aruba Access Point acting as Connector (AOS 10.8+, AP 6xx and above)

Note

EnOcean configuration is currently supported only in Aruba Central (Classic). All EnOcean-related configuration must therefore be performed in Classic Central. The New Central interface can still be used for monitoring, including visibility of Access Points that were configured in Classic Central. However, no EnOcean configuration changes can be made in New Central.

Deployment Options Summary

Option	Description	Requirements
Option A – VM Collector	Dedicated Collector virtual appliance	Any supported AP, Collector VM
Option B – AP as Connector	AP runs the connector directly	**AOS ≥ 10.8**, **AP6xx and above**

Network Requirements

All communication between the Aruba Gateway or the Aruba Connector and EnOcean SmartStudio is strictly outbound over the default HTTPS port (TCP 443).
Ensure your network/firewall allows outbound connections to the following endpoint:

- **Hostname:** ingress.enocean.cloud
- **Port:** 443 (HTTPS)

No inbound connections are required.

If your organization uses strict egress filtering, please update your policies to permit outbound HTTPS traffic to the above endpoint.

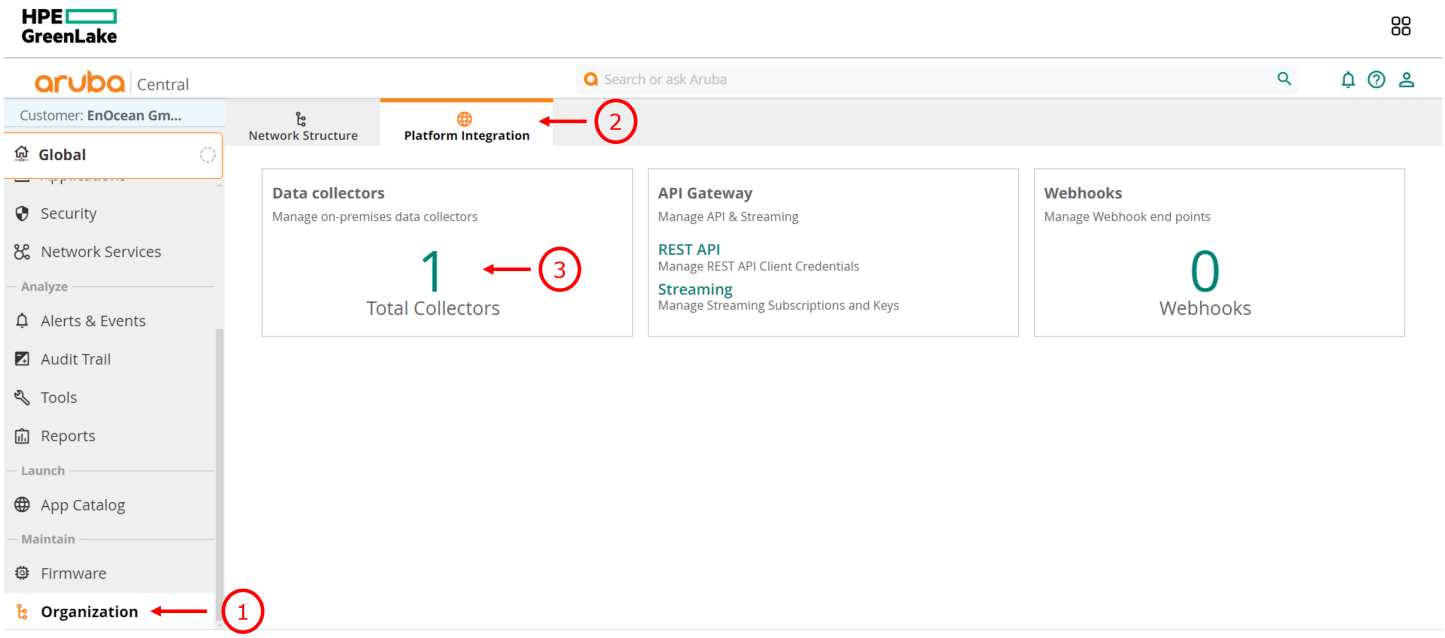
Step 1: Connect to Aruba Central

Log into the web-based management page for Aruba Central.

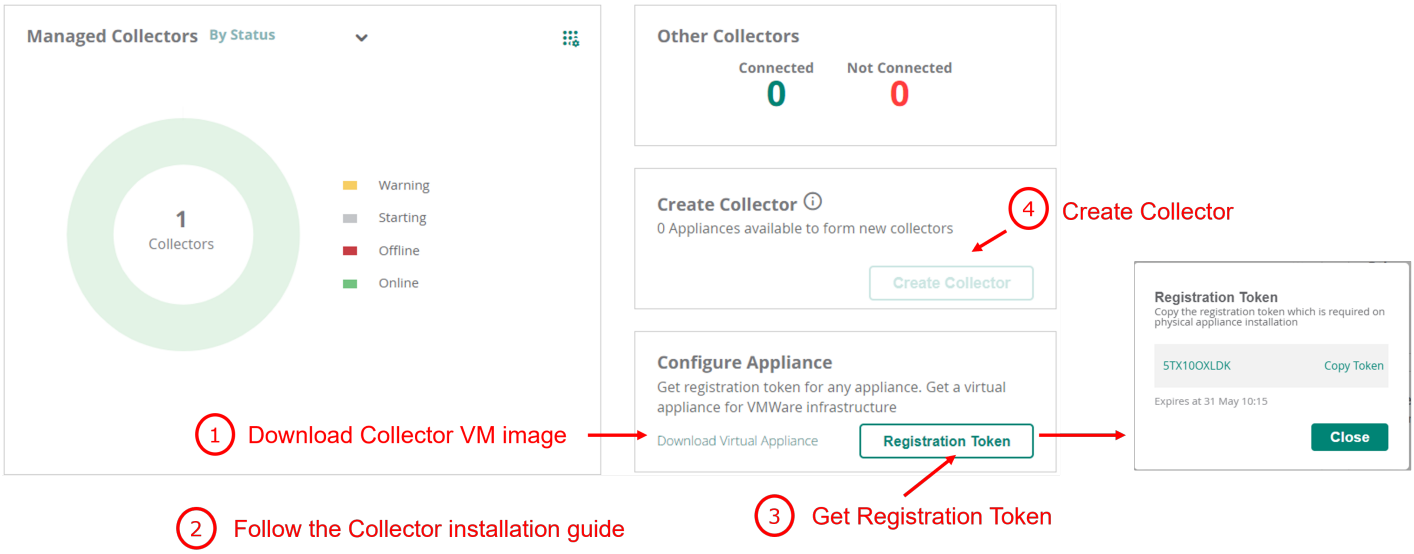
Option A: VM-Based Collector

Step A1: Install Collector

- Under **Organization** click **Platform Integration** then **Data Collectors**.

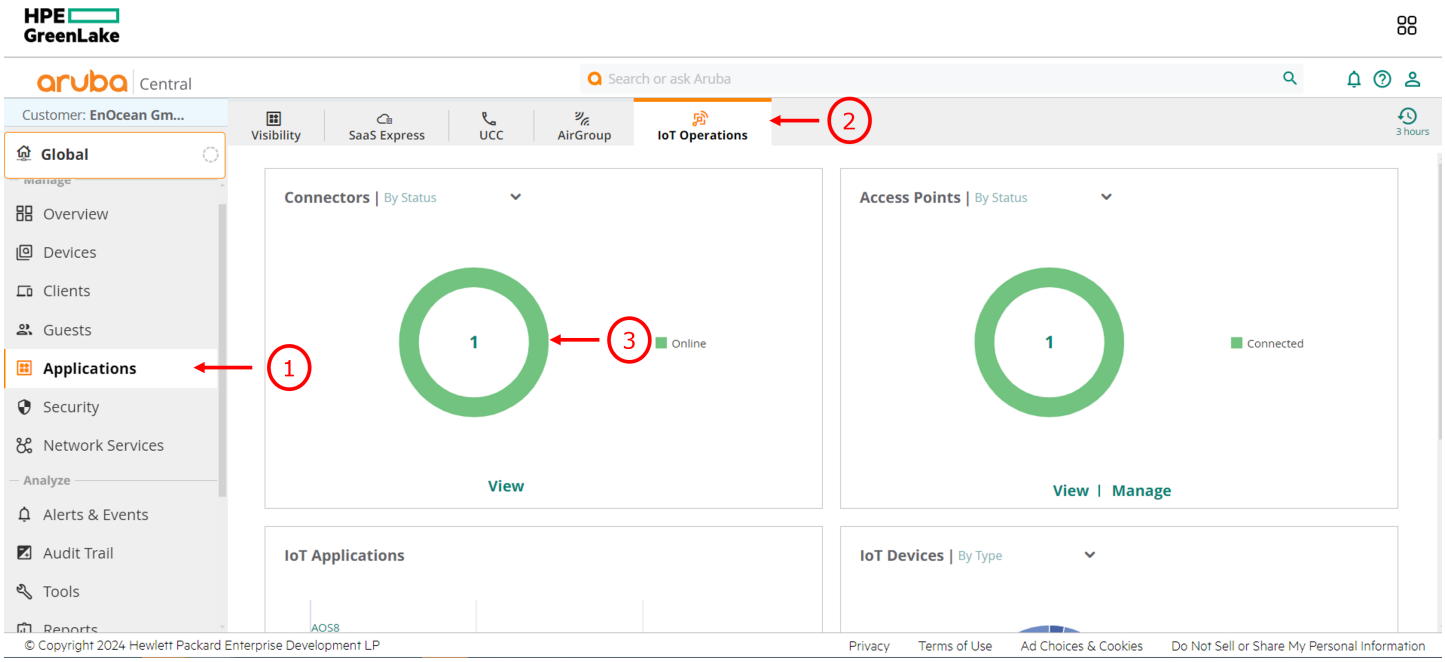


- Under the section **Configure Appliance** click **Download Virtual Appliance** (Small or Medium appliance recommended).
- Follow the collector installation guide from Aruba.
- Get registration token under **Registration Token**.
- Click on **Create collector** and follow the steps required for the connection.

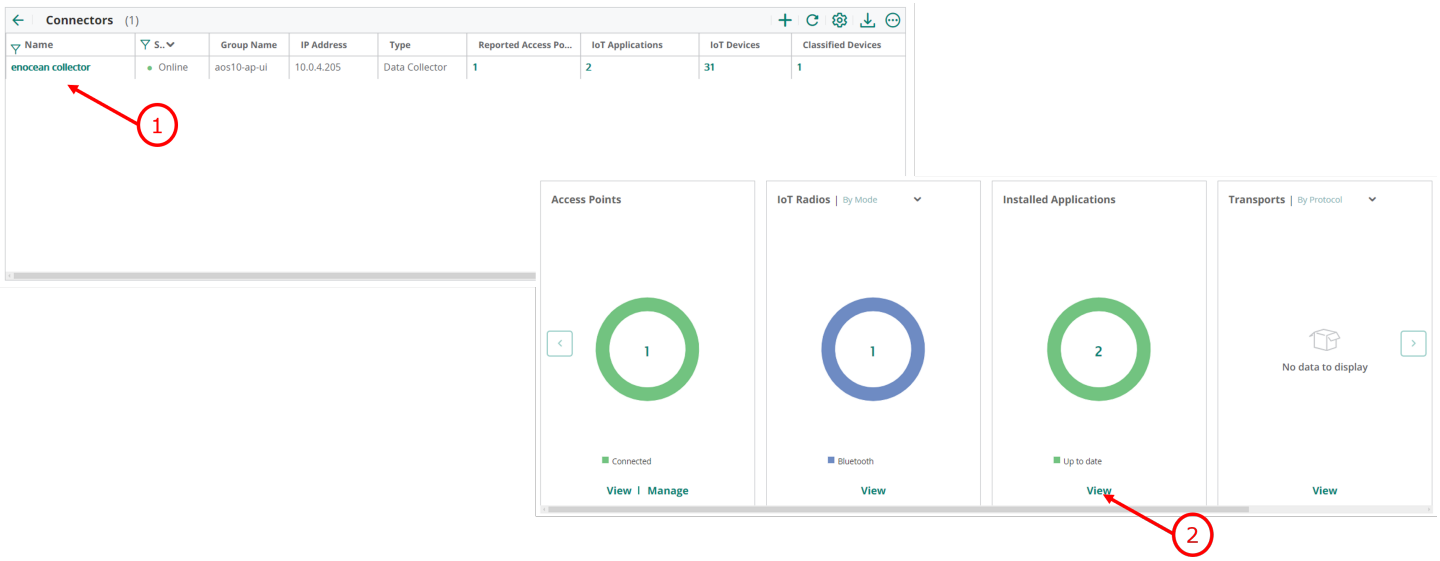


Step A2: Configure Collector

- Under **Applications** click on **IoT Operations** then **Connectors**.



- Select your connector then click on **View** under **Installed Applications**



Option B: AP as Connector (AOS 10.8+)

With this option, the Access Point itself acts as the IoT connector. No Collector VM or virtual appliance is required.

Step B1: Verify AP Eligibility

Ensure that:

- The AP is AP6xx or above
- The AP is running AOS 10.8 or later
- IoT Operations is enabled in the AP’s group

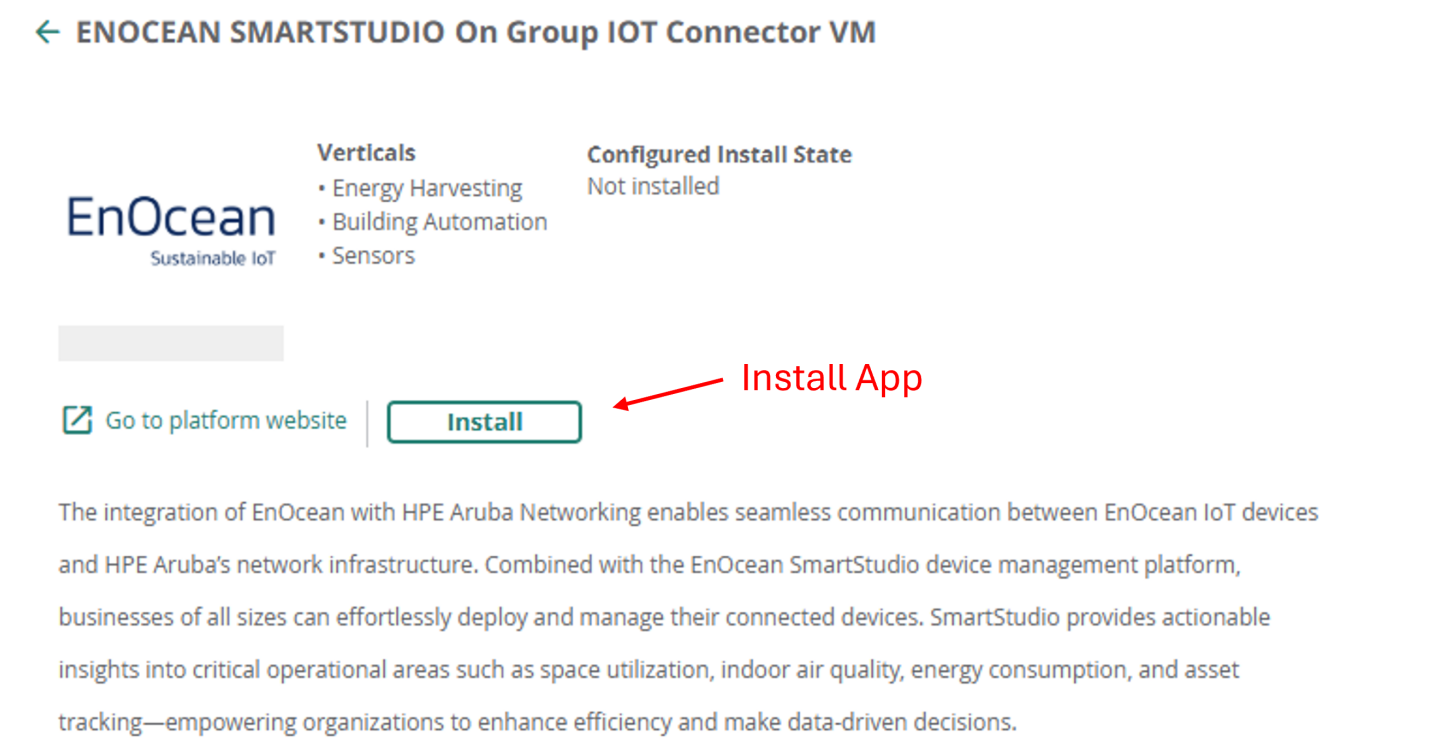
Step B2: Enable IoT Operations

1. In Aruba Central, navigate to: Applications → IoT Operations
2. Confirm the AP is available to run connectors.

Steps A1 and A2 are skipped when using this option.

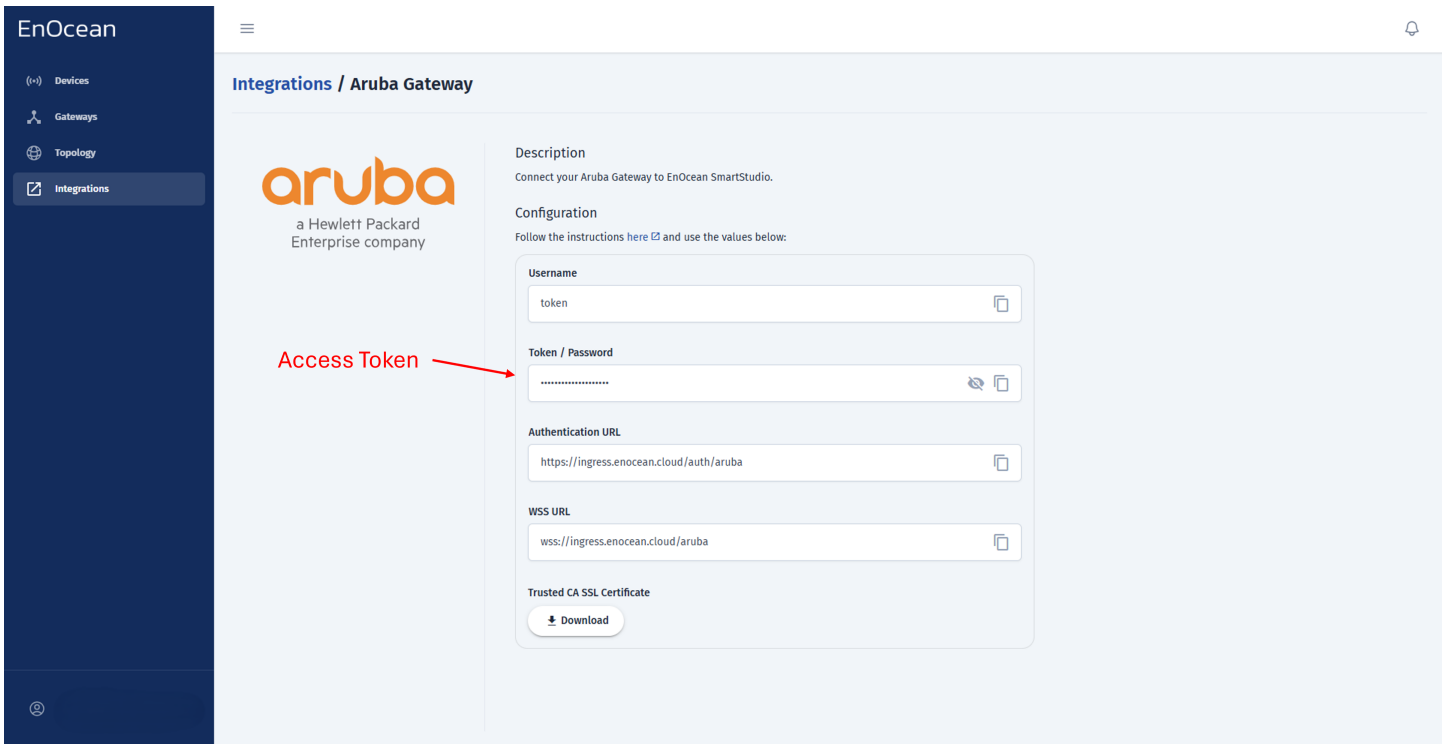
Step 2: Install EnOcean SmartStudio App

- To enable the connection from your APs to your EnOcean SmartStudio account, you will need to install the EnOcean SmartStudio App.

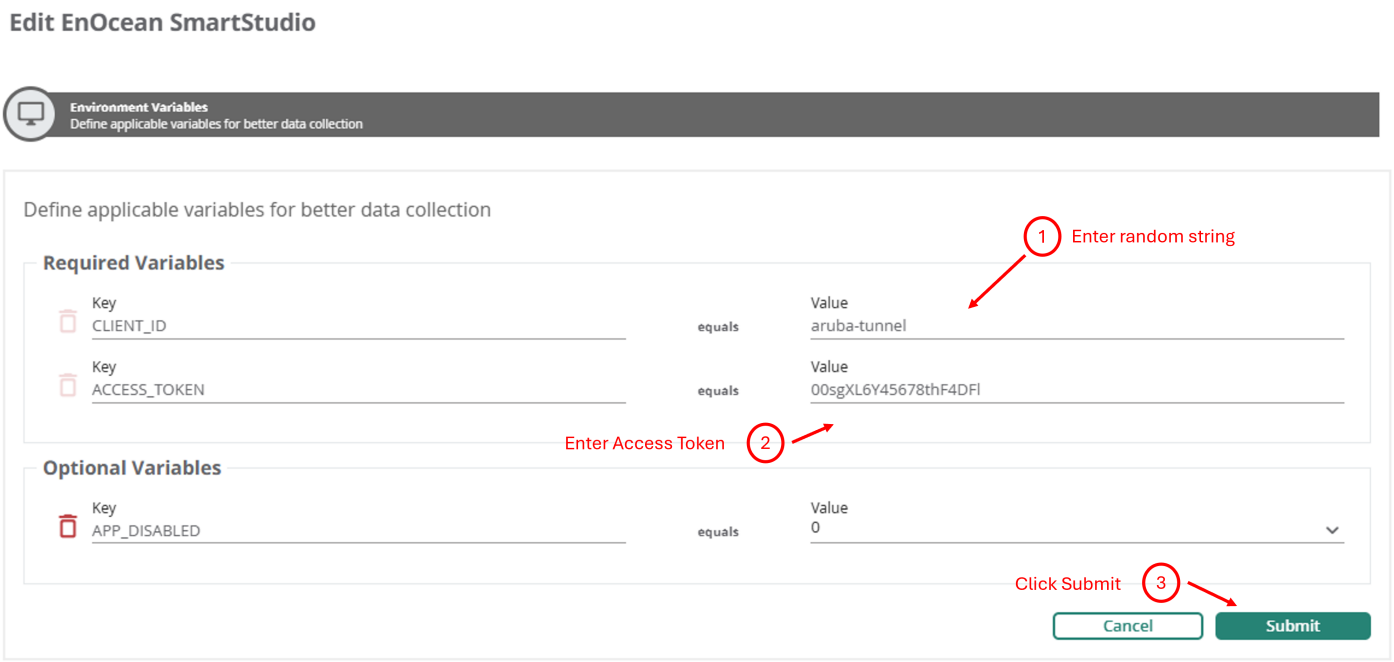


Step 3: Configure EnOcean SmartStudio App

Retrieve the **Access Token** from SmartStudio -> **Integrations** -> **Aruba Gateway** like shown below:



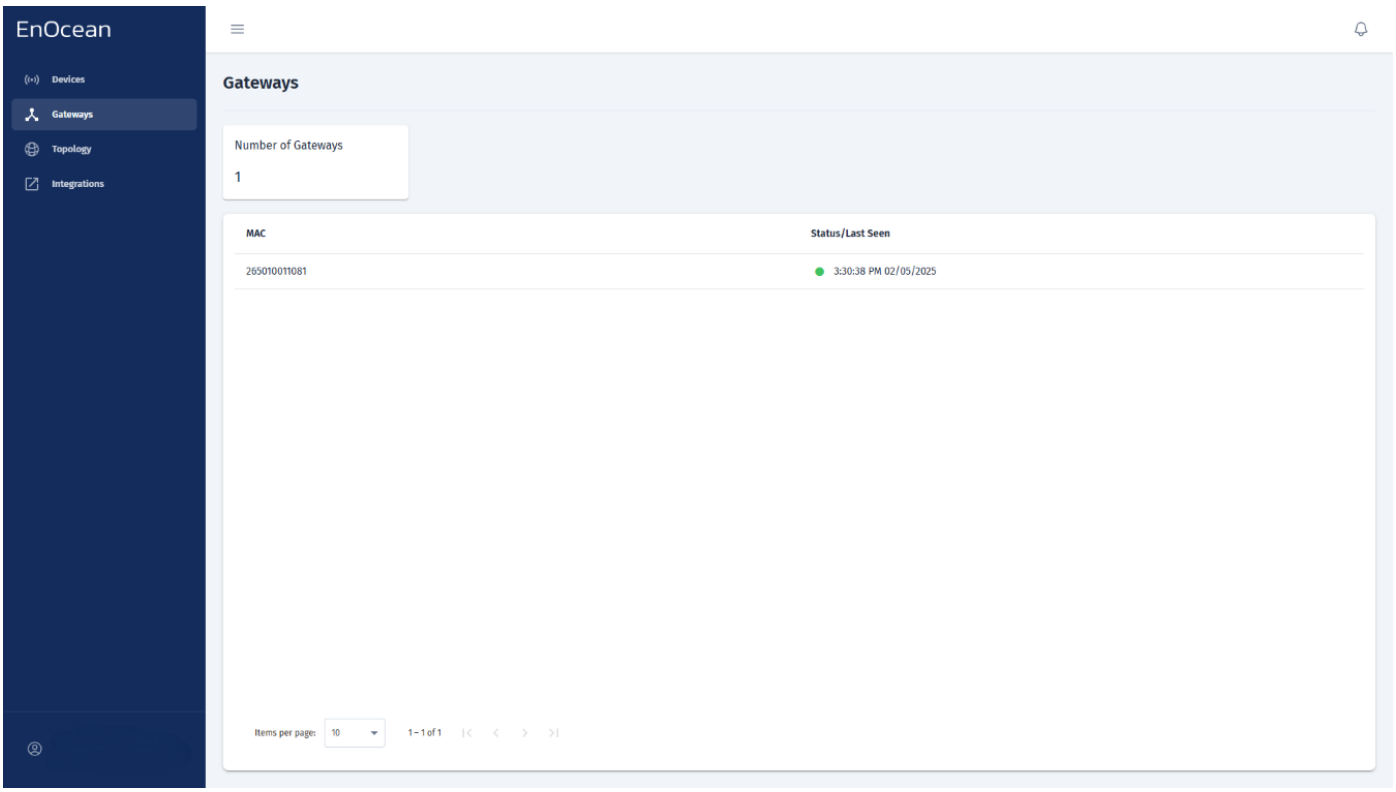
Enter the **Access Token** in the App configuration page and submit:



Step 4: Verify that your Gateway is connected

You can check the gateway status directly from the Gateways tab in the EnOcean SmartStudio dashboard:

1. Log in to the EnOcean SmartStudio web interface.
2. Navigate to the Gateways tab.
3. Locate your gateway in the list and check its connection status.



Alternatively, you can verify the gateway status using the API:

1. Login to EnOcean SmartStudio API.
2. Use the GET /v0/gateways endpoint to check the connection status.

3.1.6 Aruba APs Debugging & Troubleshooting

In case the Aruba AP is not connected to EnOcean SmartStudio i.e. the device is not listed in the gateway list or no EnOcean telegrams are visible on the egress of EnOcean SmartStudio. Try the following steps.

This guide is divided into two sections:
- **Section 1:** Debugging the IoT Transport Profile deployed on the gateway (AP or Controller).
- **Section 2:** Debugging the EnOcean SmartStudio App deployed from Aruba Central, including log retrieval.

---

1. Debugging the IoT Transport Profile on the Gateway

Note: Command syntax may change with Aruba OS releases. Tested with Aruba OS 8.10.x.

1.1 Show IoT Configuration

```
powershell aa:bb:cc:dd:ee:ff # show iot transportProfile myProfile
```

1.2 Show & Check Connected USB Devices

```powershell aa

cc:dd:ee:ff # show usb devices

USB Device Info --------------- DeviceID APMac Vendor ID Product ID Manufacturer Product Version Serial Class Device Driver Uptime -------- ----------

```

Ensure an `EnOcean USB` device is connected.

1.3 Check IoT Configuration Status

```powershell aa
```

cc:dd:ee:ff # show ap debug ble-relay iot-profile

```
ConfigID : xx --------------------------Profile[myProfile]-------------------------- authenticationURL : ... serverURL : ... ... --------------------
```

```

Look for `TransportContext: Connection Established`.
```

1.4 Check EnOcean Telegram Forwarding

```powershell aa

cc:dd:ee:ff # show ap debug ble-relay report

----------Profile[myProfile]---------- WebSocket Connect Status : Connection Established WebSocket Connection Establ

```

Monitor `Websocket Write Stats` and `Last Send Time` for activity.

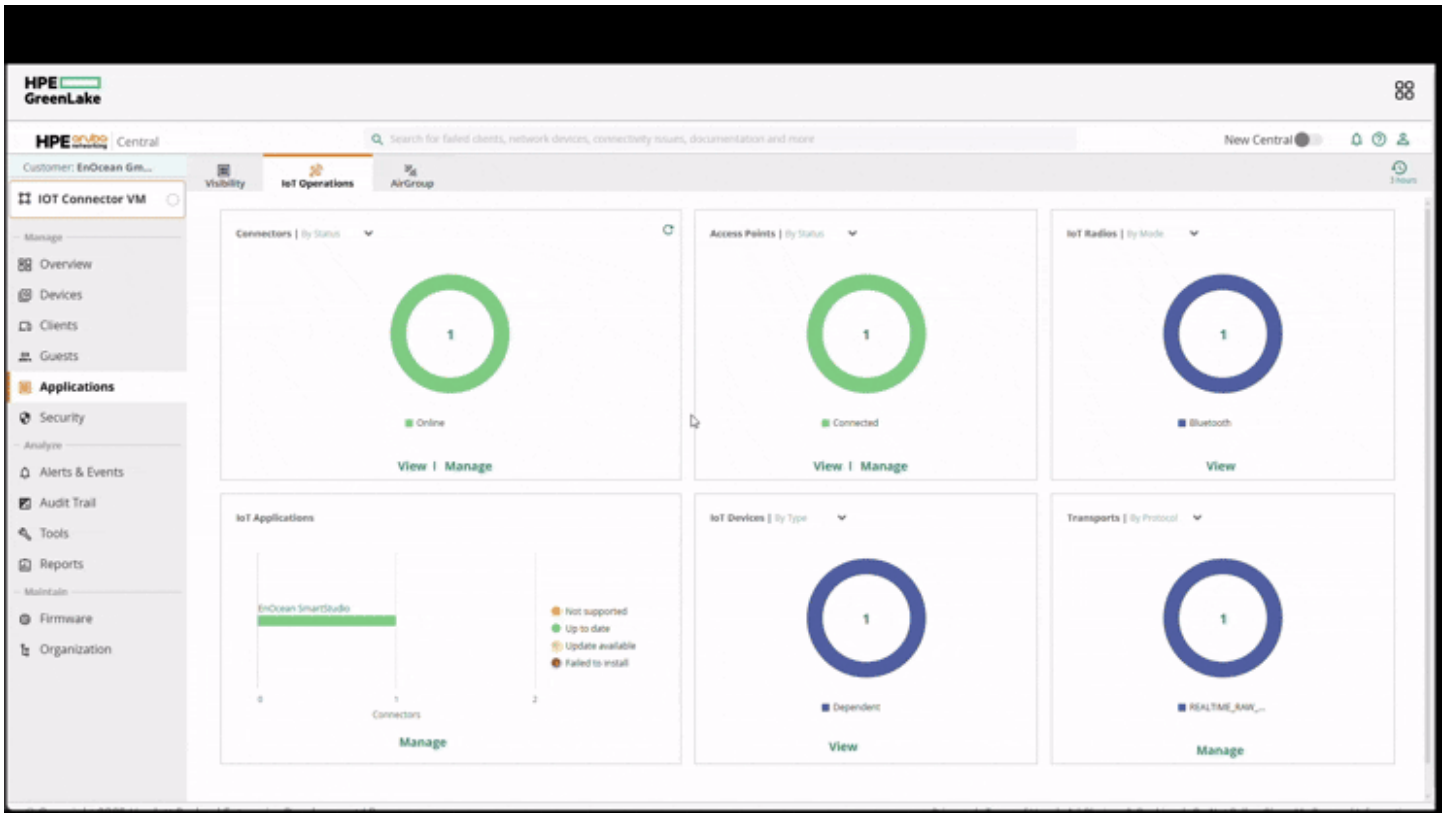
1.5 Retrieve Additional Logs

`powershell aa:bb:cc:dd:ee:ff # show ap debug ble-relay ws-log myProfile`

For Enterprise APs (Controller-managed):

```powershell

### 3.1.7 Show profiles

show iot transportProfile myProfile

### 3.1.8 Show USB devices

show ap usb-device-mgmt all

### 3.1.9 Show status and report

show ble_relay iot-profile show ble_relay report

### 3.1.10 Show Log

show ble_relay ws-log ```

---

#### 2. Debugging the EnOcean SmartStudio App from Aruba Central

If you have deployed the EnOcean SmartStudio app via Aruba Central, you can retrieve and analyze logs directly from Central for deeper troubleshooting.

#### 2.1 Accessing App Logs in Aruba Central

1. **Log in to Aruba Central** and navigate to your group or site.
2. Go to **Applications** -> **IoT Operations** and select **Manage** Under **IoT Applications**.
3. Find the **EnOcean SmartStudio** app and click on it.
4. Under the **App Status across Connectors** click on the number displaying the number of apps deployed then click **on go to app details**
5. Look for a **Logs** tab within the app details.
6. Download or view the latest logs.
```

Logs typically include connection attempts, errors, and data transmission details.



2.2 Navigating Logs

- Search for keywords such as `error`, `connection`, or `EnOcean`.
- Check timestamps to correlate with observed issues.
- Review any warnings or failures related to WebSocket connections or USB device detection.

If you continue to experience issues, contact EnOcean technical support and provide the relevant logs from both the gateway and Aruba Central.

3.2 SmartServer IoT:

3.2.1 Overview

The EnOcean SmartServer IoT is an edge controller providing connectivity for EnOcean radio devices and integration with Building Management Systems (BMS) and HVAC systems via BACnet, Modbus, and LON protocols. Installed on-premises, the SmartServer IoT enables secure connections between Operational Technology (OT) systems, IT systems, and the EnOcean SmartStudio platform.

The following describes how to configure the SmartServer IoT as a gateway for connecting EnOcean radio devices with SmartStudio. How to configure the SmartServer IoT to pull data from SmartStudio and integrate it into Building Management Systems (BMS) and HVAC systems is described here.

3.2.2 Prerequisites

Before deployment, ensure the following prerequisites are met:

1. The SmartServer is mounted and powered up.
2. The SmartServer is connected to the internet.
3. The EnOcean USB dongle must be connected.

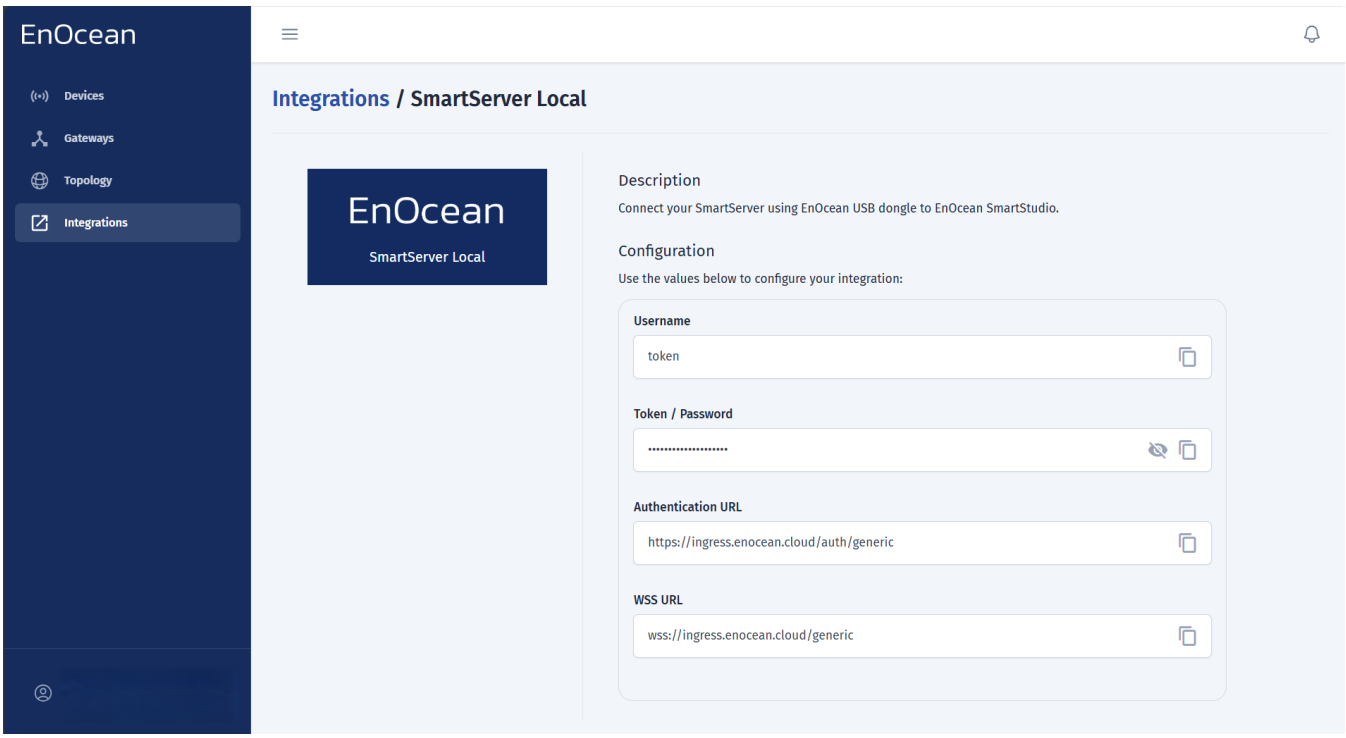
Note

Refer to the link below for the detailed steps on completing the prerequisites.
Set up SmartServer IoT

3.2.3 Configuration

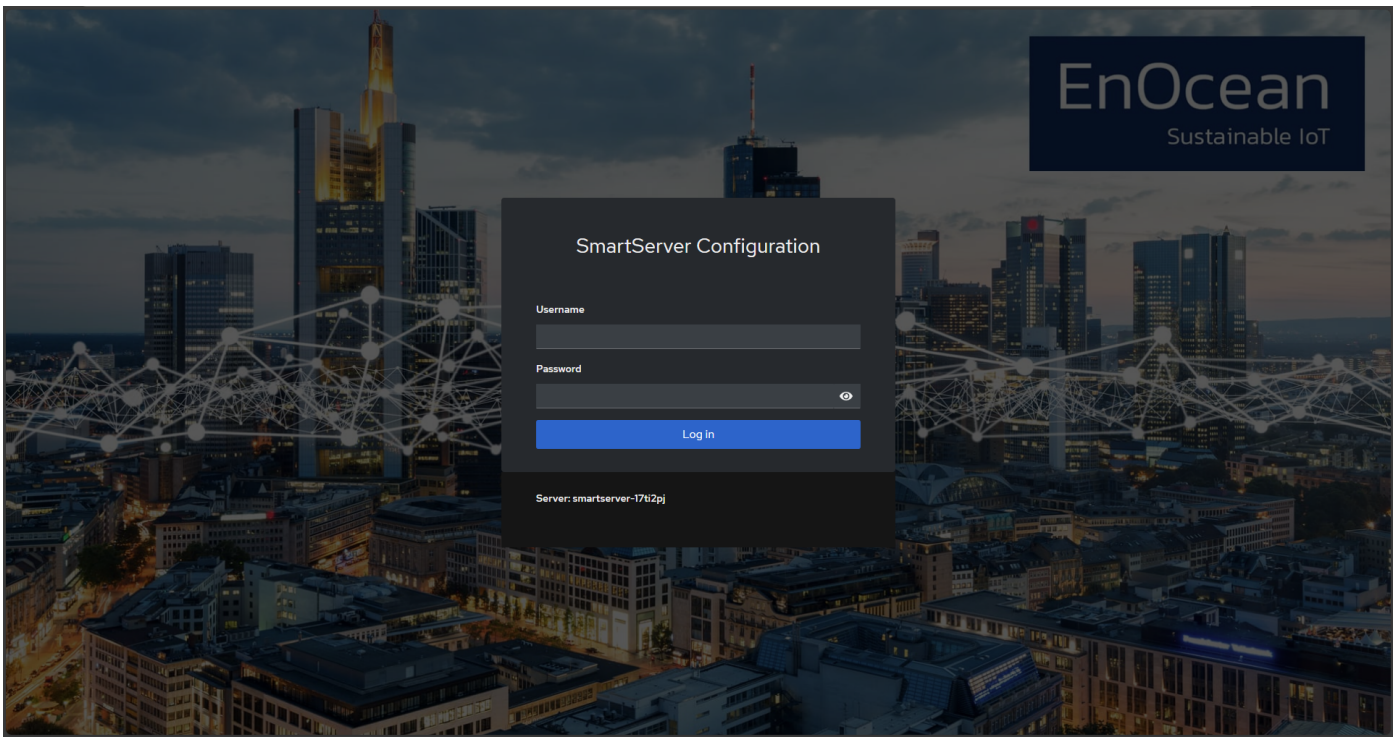
Step 1: Retrieve SmartStudio Connection Information

1. Retrieve the the **Username**, **Password**, and **Authentication URL** from SmartStudio:
   - Go to the **Integrations** section.
   - Select **SmartServer Local**.
   - Copy the information:



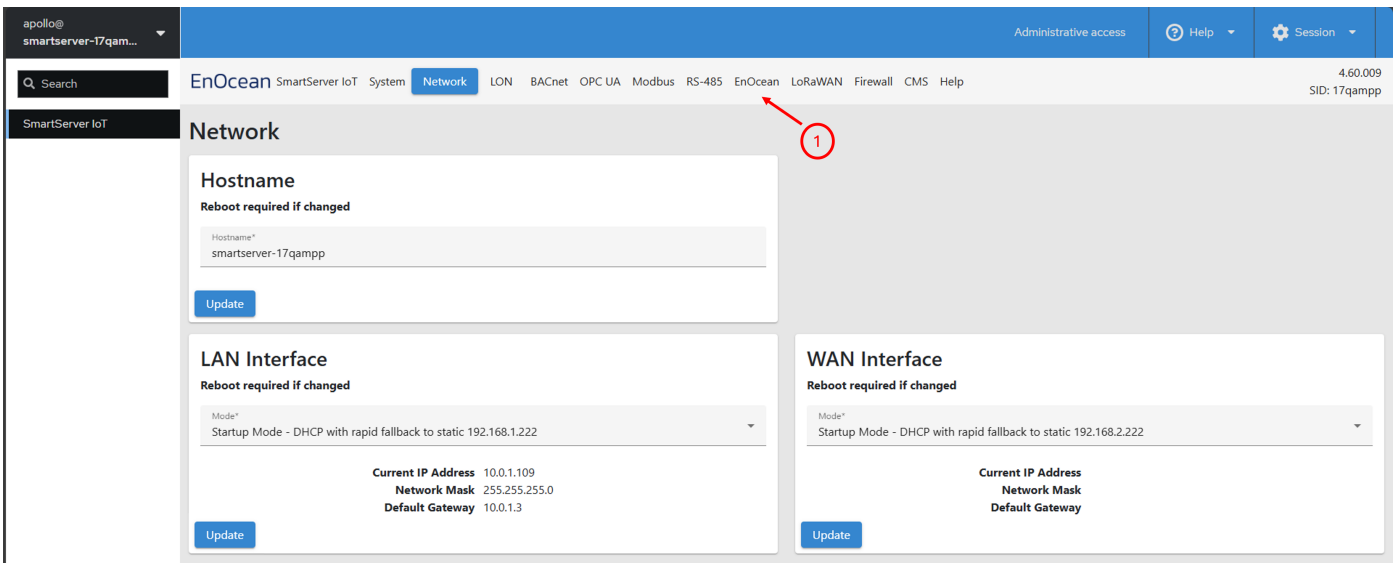
Step 2: Log in to the SmartServer Web Management UI

1. Open a web browser and navigate to the SmartServer's Configuration Page.
2. Log in using your apollo user and password.



Step 3: Navigate to the EnOcean Tab

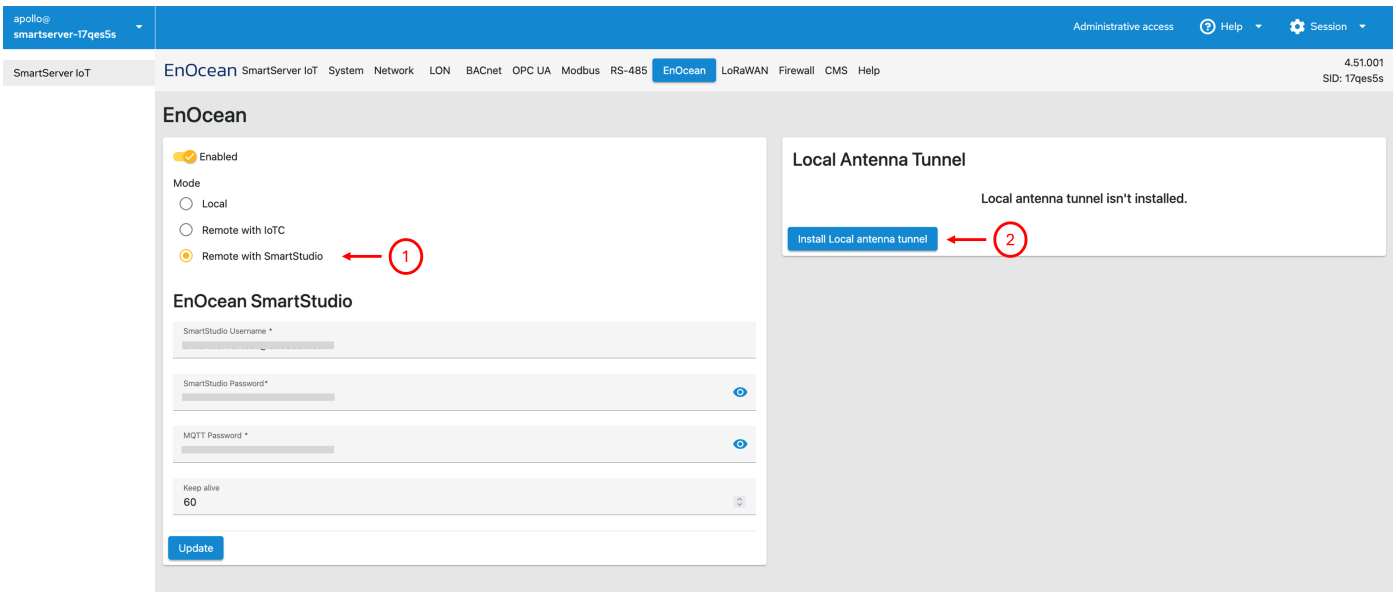
1. Once logged in, go to the **EnOcean** tab in the SmartServer Configuration Page as shown below.



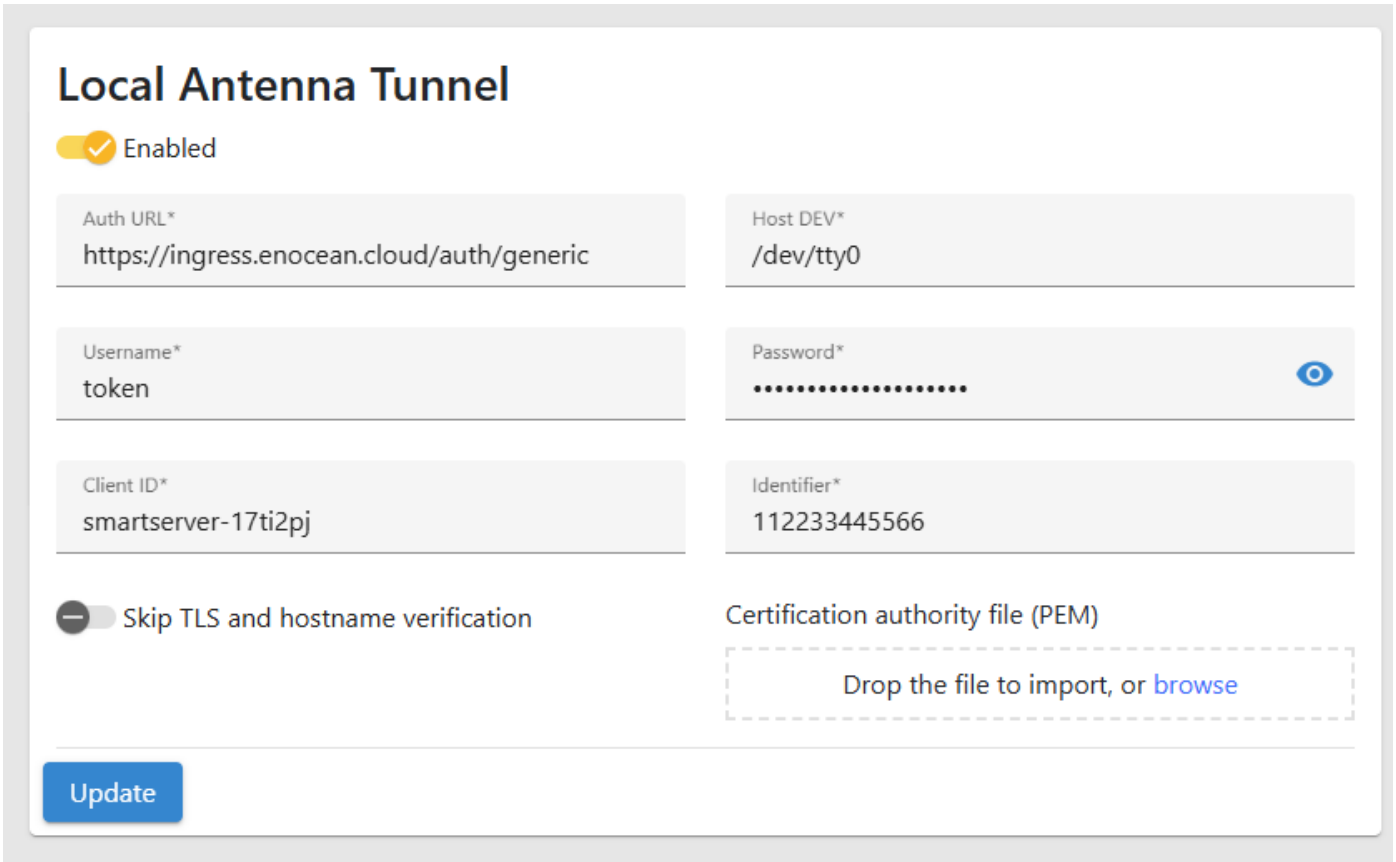
Step 4: Authenticate with EnOcean SmartStudio

1. In the left section of the **EnOcean** tab, enable the switch and select **Remote with SmartStudio**.
2. Leave the Username and Password blank, unless the SmartServer should pull information from SmartStudio as described here.

3. Click **Update**.
4. In the right section of the **EnOcean** tab, Install Local antenna tunnel.



5. Enter the information to connect with SmartStudio as shown below, then click **Update**:



Note

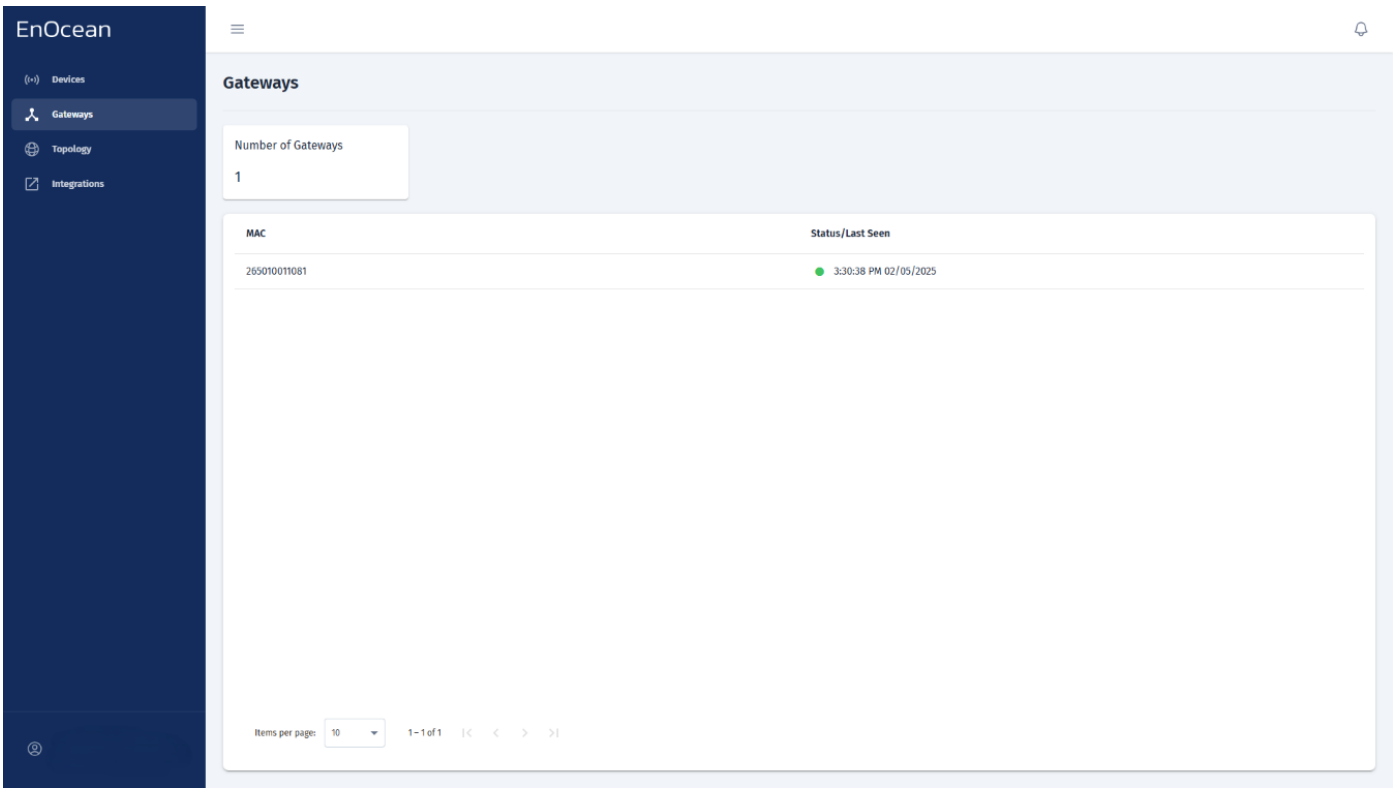
- **Auth URL**: Retrieved from SmartStudio
- **Host DEV**: Select the USB dev port of the EnOcean USB dongle
- **Username**: Retrieved from SmartStudio
- **Password**: Retrieved from SmartStudio
- **Client ID**: Enter the SmartServer hostname obtained from product label.
- **Identifier**: Enter the 12 Digit hex code of your MAC ID 1 obtained from product label.

More detailed information on the setup can be found here.

Step 5: Verify the Connection

You can check the SmartServer IoT status directly from the **Gateways** tab in the EnOcean SmartStudio dashboard:

1. Log in to the EnOcean SmartStudio web interface.
2. Navigate to the **Gateways** tab.
3. Locate your SmartServer IoT in the list and check its connection status.



Alternatively, you can verify the gateway status using the API:

1. Log in to the EnOcean SmartStudio API.
2. Use the `GET /v0/gateways` endpoint to check the connection status.

3.3 OPUS IQ DOT

3.3.1 Overview

The OPUS IQ DOT from DC Next can be used as a gateway to connect EnOcean devices with SmartStudio.

3.3.2 Prerequisites

Before deployment, ensure the following prerequisites are met:

1. The OPUS IQ DOT is mounted and powered up.
2. The OPUS IQ DOT is connected to the internet.

Note

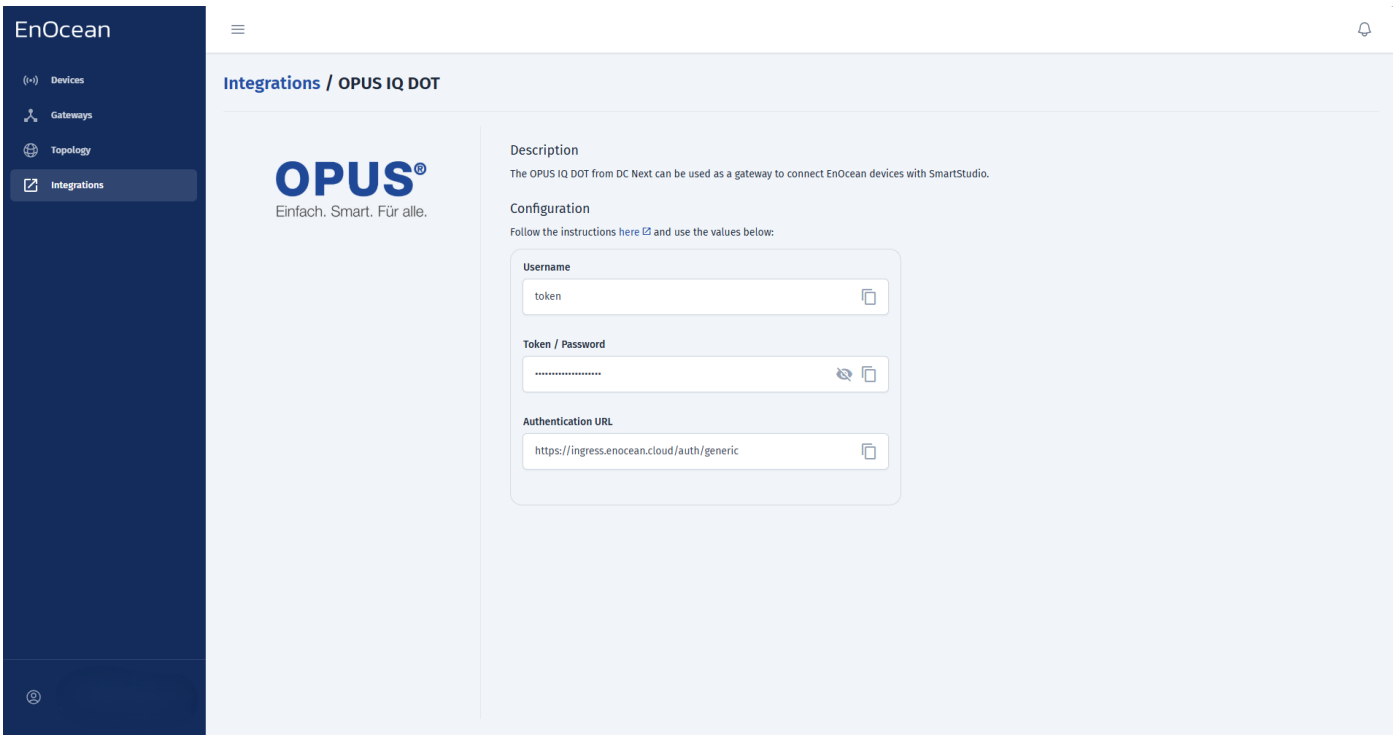
For detailed instructions on mounting and connecting the OPUS IQ DOT, refer to the guides provided by DC Next.

3.3.3 Configuration

Step 1: Retrieve SmartStudio Connection Information

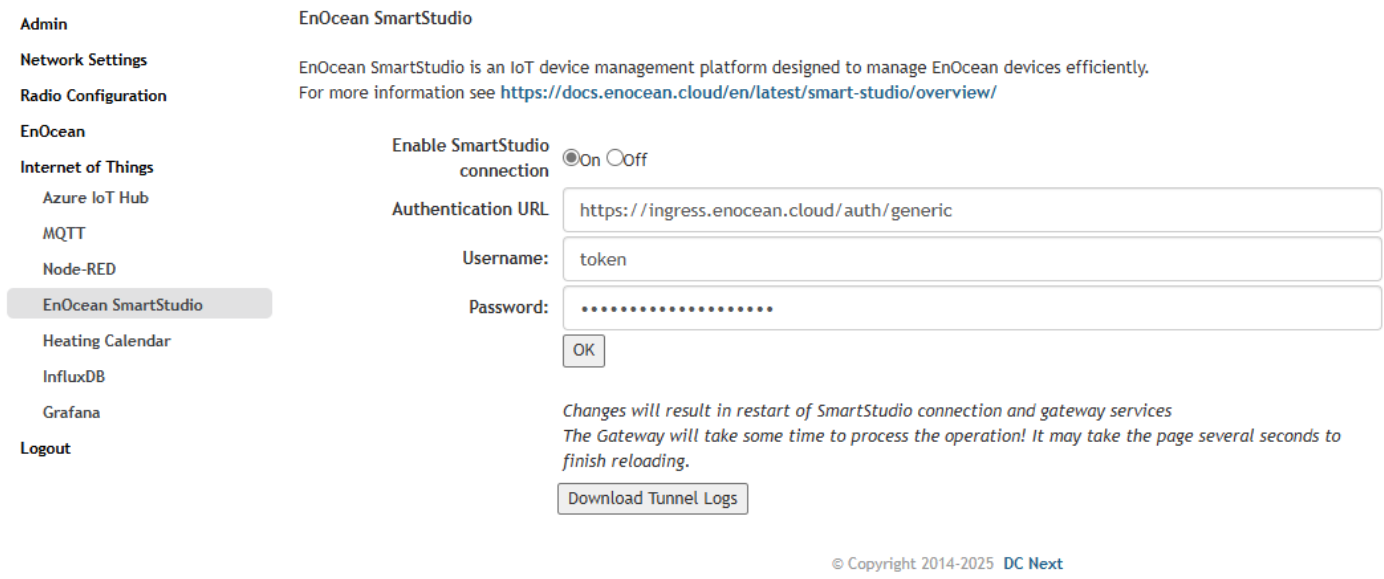
1. Retrieve the **Username**, **Password**, and **Authentication URL** from SmartStudio:

    - Go to the **Integrations** section.
    - Select **OPUS** under Gateways.
    - Copy the information:



Step 2: Configure SmartStudio Connection

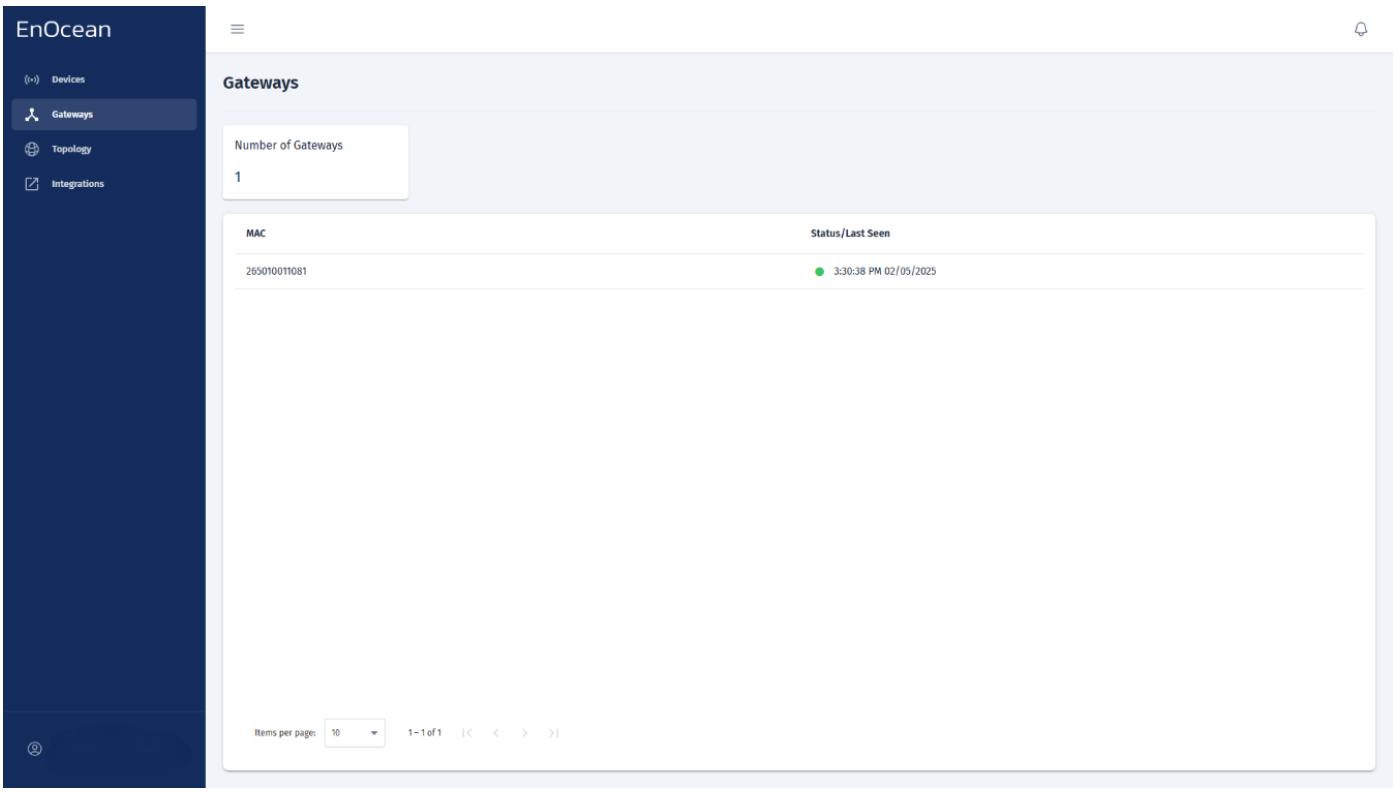
1. In the Management UI of your gateway, navigate to **Internet of Things → EnOcean SmartStudio**.
2. Fill in the required fields with the information obtained above.
3. Set **Enable SmartStudio Connection** to **On**.
4. Click **OK** to save your settings.



Step 3: Verify Gateway Connection

You can check the gateway status directly from the **Gateways** tab in the EnOcean SmartStudio dashboard:

1. Log in to the EnOcean SmartStudio web interface.
2. Navigate to the **Gateways** tab.
3. Locate your gateway in the list and check its connection status.



Alternatively, you can verify the gateway status using the API:

1. Log in to the EnOcean SmartStudio API.
2. Use the `GET /v0/gateways` endpoint to check the connection status.

4. Integrations

4.1 REST API

4.1.1 Swagger API Overview

This API allows EnOcean SmartStudio users to interact with and manage your devices, gateways, and system resources within the platform. It provides a robust set of endpoints to authenticate users, manage devices, retrieve system data, and gateways.

The API is structured to support the automation of device onboarding, real-time telemetry collection, system health checks, and much more. Whether you're building applications for device control, monitoring, or reporting, this API provides the essential tools to integrate seamlessly with the platform.

EnOcean SmartStudio API is accessible from the link below:

EnOcean SmartStudio API

Key Features

- **Authentication**: Secure access to the platform using token-based authentication.
- **Device Management**: Comprehensive device lifecycle management including adding, updating, and deleting IoT devices.
- **System Operations**: Manage and retrieve system backups, restore instances, and check the overall health of the platform.
- **Gateway Monitoring**: Access detailed information about gateways, including health, performance, and data statistics.

Sections

This API is divided into four major sections based on functionality:

1. **Auth**: Handles user authentication and token management, ensuring secure access to the platform.
   - Endpoints: `/auth/login`, `/auth/logout`, `/auth/accessToken`
2. **System**: Allows for system-level operations such as backups, restores, and health checks.
   - Endpoints: `/system/backup`, `/system/restore`
3. **Device**: Manages IoT devices, including adding, updating, retrieving, and removing devices. It also provides access to device-specific telemetry and signal data.
   - Endpoints: `/devices`, `/devices/{deviceId}`, `/devices/{deviceId}/telegram`, `/devices/{deviceId}/telemetry`, `/devices/{deviceId}/signal`
4. **Gateway**: Provides gateway management and monitoring capabilities, offering insight into gateway status, health, and performance.
   - Endpoints: `/gateways`, `/gateways/{mac}`, `/gateways/{mac}/stats`, `/gateways/{mac}/health`

Authentication and Authorization

To interact with most endpoints, users must authenticate using the **Auth** endpoints.

How to Use the API

1. **Authenticate**: Use the `/auth/login` endpoint to login to the platform.
2. **Perform Operations**: Once authenticated, you can interact with devices, gateways, or system operations depending on your needs.
3. **Monitor and Manage**: Use device telemetry, signal, and system health endpoints to monitor the health and performance of your platform.

Response Format

All API responses are returned in JSON format. For successful requests, a status code of `200 OK` or `201 Created` is returned, along with the relevant data. In the event of an error, the response will include an appropriate HTTP status code (e.g., `400 Bad Request`, `401 Unauthorized`, `404 Not Found`) and a descriptive error message.

Status Codes

- **200 OK**: The request was successful.
- **201 Created**: The resource was successfully created.
- **400 Bad Request**: The request contains invalid parameters.
- **401 Unauthorized**: Authentication failed or is required.

- **404 Not Found**: The requested resource could not be found.
- **500 Internal Server Error**: The server encountered an unexpected condition.

Conclusion

This API is designed to provide a flexible and secure way to manage your IoT devices and gateways while monitoring system performance. With a rich set of endpoints, it supports both operational needs and real-time telemetry data.

For detailed information on each endpoint, refer to the specific sections for **Auth**, **System**, **Device**, and **Gateway**.

Auth Endpoints

POST /auth/login

Login to the platform.

- **Request Body:**

```
json { "username": "example@example.com", "password": "example" }
```

- **Response:**

```
json { "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." }
```

POST /auth/logout

Logout of the platform, invalidating the current session.

- **Request:** No request body required. Uses the access token in headers.
- **Response:**
  - 200 OK on successful logout.

GET /auth/accessToken

Retrieve a new access token using the refresh token.

- **Request:** Uses the refresh token in headers.
- **Response:**

```
json { "credentialsType": "ACCESS_TOKEN", "credentialsId": "bmemh9OHFEBLGc...", "credentialsValue": null }
```

System Endpoints

GET /system/backup

Retrieve the backup of your instance.

- **Request:** No request body required.
- **Response:**

```
json { "apiVersion": "1.0", "exportedAt": "2024-10-15T08:54:14.491Z", "devices": [ { ... }, { ... }, ... ], "gateways": [ { .
```

POST /system/restore

Restore your instance to a previous state.

- **Request Body:**

```
json { "apiVersion": "1.0", "exportedAt": "2024-10-15T08:54:14.491Z", "devices": [ { ... }, { ... }, ... ], "gateways": [ { .
```

- **Response:**
  - 200 OK on successful restore.

Device Endpoints

GET /devices

Retrieve a list of devices.

- **Response:**

```
json [ { "tenantId": "abcdbe80-8ac6-11ef-a33b-...", "eurid": "05066ca8", "destinationEurid": "ffffffff", "friendlyid": "noCor
```

POST /devices

Add new devices to the system.

- **Request Body:**

```
json { "friendlyid": "Room Panel 02", "eep": "A5-04-05", "deviceType": "sensor", "eurid": "a1b2c3d4", "destinationEurid": "ff
```

- **Response:**
  - 201 Created on successful addition.

DELETE /devices

Remove all devices from the system.

- **Response:**
  - 200 OK on successful deletion.

GET /devices/{deviceId}

Retrieve a specific device by its ID.

- **Parameters:**
  - DeviceId: Identifier of the device.
- **Response:**

```
json { "tenantId": "abcdbe80-8ac6-11ef-a33b-2900...", "eurid": "05066ca8", "destinationEurid": "ffffffff", "friendlyid": "noC
```

PUT /devices/{deviceId}

Update details of a device by its ID.

- **Parameters:**
  - DeviceId: Identifier of the device.
- **Request Body:**

json { "friendlyid": "Room Panel 02", "eep": "A5-04-05", "deviceType": "sensor", "eurid": "a1b2c3d4", "destinationEurid": "ff

- **Response:**
  - 200 OK on successful update.

DELETE /devices/{deviceId}

Delete a device by its ID.

- **Parameters:**
  - DeviceId: Identifier of the device.
- **Response:**
  - 200 OK on successful deletion.

GET /devices/{deviceId}/telegram

Retrieve the telegram (communication packets) information of a device.

- **Parameters:**
  - DeviceId: Identifier of the device.
- **Response:**

json [{ "uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "data": "string", "sender": "string", "status": "string", "sub_teleg

GET /devices/{deviceId}/telemetry

Retrieve telemetry information (e.g., sensor data) for a device.

- **Parameters:**
  - DeviceId: Identifier of the device.
- **Response:**

```
json [ { "uuid": "582e79a9-ec42-4472-9f4c-2a4b400c326a", "timestamp": 1649200131, "raw_data": "d2aa0d002563f2fbd7a0", "data":
```

GET /devices/{deviceId}/signal

Retrieve signal strength or other signal-related metrics for a device.

- **Parameters:**
  - DeviceId: Identifier of the device.
- **Response:**

```
json [ { "raw": { "uuid": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "data": "string", "sender": "string", "status": "string", "
```

Gateway Endpoints

GET /gateways

Retrieve all gateways in the system.

- **Response:**

```
json [ { "mac": "1c28afc29...", "hardwareDescriptor": "AP-505", "softwareVersion": "8.10.0.2" }, { "mac": "1c28afc29...", "ha
```

GET /gateways/{mac}

Retrieve details for a gateway by its MAC address.

- **Parameters:**

mac: Identifier of the gateway.

- **Response:**

```
json { "hardwareDescriptor": "hardwareDescriptor", "mac": "mac", "softwareVersion": "softwareVersion", "lastSeen": "lastSeen"
```

GET /gateways/{mac}/stats

Retrieve statistics (e.g., data throughput, error rates) for a gateway.

- **Parameters:**

mac: Identifier of the gateway.

- **Response:**

```
json { "lastSeen": "12/02/2022 14:32:12 GMT+3", "succesfullyProcessed": 6, "notProcessed": 1, "totalTelegramCount": 0 }
```

GET /gateways/{mac}/health

Retrieve the health status of a gateway, including uptime and performance metrics.

- **Parameters:**

mac: Identifier of the gateway.

- **Response:**

```
json { "timestamp": 123575746883, "usbHealth": "healthy", "usbIdentifier": "0xff33a5" }
```

4.2 MQTT

4.2.1 MQTT API Overview

The **EnOcean SmartStudio MQTT Interface** provides a powerful and flexible way for applications to integrate with devices using the lightweight **MQTT protocol**. This interface enables seamless integration of EnOcean IoT devices with various third-party systems by allowing device data to be published and subscribed to in real-time.

Overview

The MQTT API supports two modes of operation:

1. **MQTT Client**: EnOcean SmartStudio acts as an MQTT client, publishing data to an MQTT broker and allowing clients to subscribe to predefined topics for telemetry, events, stats, and RPC commands/results.
2. **External MQTT Broker**: EnOcean SmartStudio can connect to an external MQTT broker, enabling the use of custom topics for telemetry, events, stats, RPC commands, and RPC results.

This flexibility allows seamless integration with both internal and external systems, depending on your requirements.

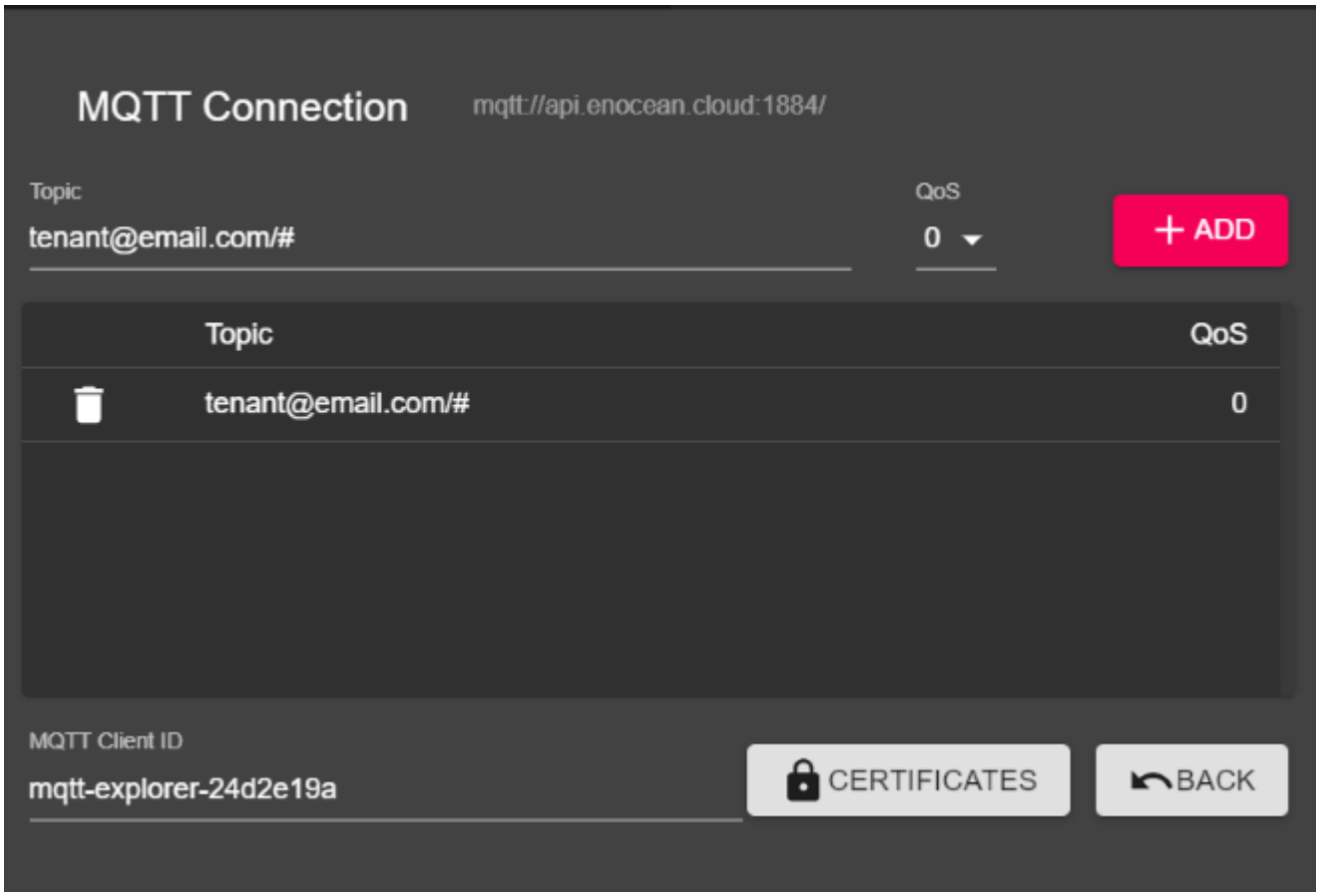
MQTT Client

The MQTT Client mode allows EnOcean SmartStudio to act as a client, publishing data to an MQTT broker. Clients can subscribe to predefined topics to receive updates.

MQTT Topics

Important

To start receiving data, you must subscribe to the relevant topics under `<your-MQTT-username>/#`. For example:



EnOcean SmartStudio specifies the topics below for MQTT:

PATH	Description
`tenant@email.com/v0/sensor/[ID]/telemetry`	EnOcean device telemetry of a specific [ID]. Publishing is done every time a valid telegram is processed. Payload consists of a JSON file described here.
`tenant@email.com/v0/sensor/[ID]/meta/event/`	Event information of a specific [ID]. Publishing is done with a specific event. Reference of possible events and content of JSON files can be found here.
`tenant@email.com/v0/sensor/[ID]/meta/stats/`	Statistical information about traffic of a specific [ID]. Publishing is done in predefined time intervals of 10 minutes. Published JSON Payload can be reviewed here.
`tenant@email.com/v0/device/[ID]/rpc/command`	Topic for queuing new commands.
`tenant@email.com/v0/device/[ID]/rpc/cancel`	Topic for deleting waiting commands.
`tenant@email.com/v0/device/[ID]/rpc/result`	Topic for results of requests.

Note

All timestamps in EnOcean SmartStudio are in Unix epoch time. It can be converted into a human-readable format using tools like this online converter.

External MQTT Broker

The External MQTT Broker mode allows EnOcean SmartStudio to connect to an external broker, enabling the use of custom topics for telemetry, events, stats, RPC commands, and RPC results.

Key Features

- **Custom Topics**: Define your own topic structure for telemetry, events, stats, and RPC communication.
- **Broker Flexibility**: Integrate with any external MQTT broker that supports standard MQTT protocols.
- **TLS Support**: Secure communication with the broker using TLS authentication and certificate validation.

Configuration

1. Navigate to the **MQTT Settings** in the SmartStudio Web Management UI.
2. Enable the **External Broker** option.
3. Enter the following details:
   - **Broker URL**: The URL of your external MQTT broker.
   - **Port**: The port used by the broker (default: `1883` or `8883` for TLS).
   - **Username**: Your MQTT broker username.
   - **Password**: Your MQTT broker password.
   - **Custom Topics**: Define the topics for telemetry, events, stats, RPC commands, and RPC results.

   Note

   Ensure that the external broker is configured to accept connections from EnOcean SmartStudio.
4. Save the configuration and verify the connection status.

Summary

The MQTT API provides flexibility for integrating EnOcean SmartStudio with third-party systems. Whether you use the built-in MQTT client or connect to an external broker, the interface ensures seamless data exchange for telemetry, events, stats, and RPC communication.

4.2.2 Sensor telemetry

Each output JSON consist of these sections:

- `sensor` - stored information about the sensor provided at onboarding via the API
- `telemetry` - information interpreted by the engine
  - `data` - sensor data included in the message and encoded via the EEP
  - `signal` - meta information about the sensor and encoded as signal telegram
  - `meta/stats` - meta information about the message added by the engine
- `raw` - raw message information
  - `rssi` - radio signal strength information. Important to track radio quality

telemetry -> data

The data is included in a JSON file as `key-value` pairs following the EnOcean Alliance IP Specification. Example JSON outputs from selected devices are available below.

Multisensor

CO2 sensor

Switch Module

EnOcean IoT Multisensor

```
json { "sensor": { "friendlyId": "Multisensor 1", "id": "04138bb4", "location": "Cloud center" }, "telemetry": { "data": [{ "
```

```
json { "sensor": { "friendlyId": "co2_Hardware2", "id": "051b03c9", "location": "Hardware 2" }, "telemetry": { "data": [{ "ke
```

PTM215 battery-less switch module

```
json { "sensor": { "friendlyId": "switch1", "id": "feee14ab", "location": "Entrance" }, "telemetry": { "data": [{ "key": "ene
```

telemetry -> signal

Selected devices from EnOcean transmit additionally to their data messages also messages about their internal states or events. This messages are known as signal telegrams. Signal telegrams include information about the:

- percentage of remaining energy available in the energy storage
- how much energy is provided via the energy harvester
- availability and status of a back up energy store
- for additional information see the signal telegrams specification and data sheet of your EnOcean product

Example of an energy `MID: 6` signal telegram is below:

`json { "sensor": { "friendlyID": "0413D759 D2-14-41 SIMU Multisensor", "id": "0413d759", "location": "Office 265", "eep": "d2`

telemetry -> meta

The `meta` section is complementary to `data` and `signal`. The meta section includes the `stats` section as provided by the API for the referenced device. Additionally the egress timestamp is included.

Examples are visible with the above examples with `data` and `signal`.

raw -> rssi

The `raw` element includes the radio telegram Information as received by the EnOcean SmartStudio. They are mostly included for tracking and debug purposes. The `rssi` is the only one of interest.

The `rssi` radio signal strength information provides important information about connectivity. We recommend to track it and raise and alarm if the level drops or changes significantly.

4.2.3 Sensor Meta

event

EnOcean SmartStudio provides important information about events that were detected in regard to the sensor status, data transmission or behavior.

There are these types of events:

Type	Event	Description
Security	`MAC_VALIDATION_ERROR`	A received message could not be authentificated with the included CMAC. This could be an indication for a security attack.
	`RLC_REPLAY`	A received message has a lower message sequence counter then the previous. This could e an indication for an replay attack.
	`DEVICE_SEND_NOW_UNSECURE`	A device which was onboarded as secure is now transmitting as non secure. This is an indication of compromision the set security level, possible attack.
Health	`FIRST_TIME_SEND`	An onboarded device transmitted for the first time.
Processing	`EEP_DECODE_ERROR`	The receive message could not be decoded with the specified EEP. This is an indication for an corrupted radio message (if occuring on limited basis) or wrong specified EEP (if occuring pernament).
	`EEP_NOT_FOUND_ERROR`	The specified EEP of an device is not known to EnOcean SmartStudio. Please contact our support in such case.

Example of an Health `FIRST_TIME_SEND` message is below:

```
json { "sensor": { "friendlyID": "Multisensor 1", "id": "04138d23", "location": "Cloud center", "eep": "d2-14-41", "customTag
```

stats

The telegrams `stats` of individual EnOcean devices are posted periodically. This should indicate their operational status and additionally provide operational updates.

Example of an stats message is listed below:

```
json {   "sensor": {     "friendlyID": "Multisensor 1",     "id": "04138d23",     "location": "Cloud center",     "eep": "d2-
```

4.2.4 Sensor RPC

A request is made using the `tenant@email.com/v0/device/{device_id}/rpc/command` topic with command schema as shown below. The command payload is using keys from the JSON representation of the EEP. More information about the payload is available in the description of the devices.

```
json { "request_id": 1, "eep": "A5-20-01", "payload": { "summerMode": 0, "temperature": 0, "temperatureSetpoint": 0, ... } }
```

key	type and meaning
request_id	integer, unique identifier generated by application and used for tracking requests
eep	string, EnOcean Equipment Profile. This may be different from device's reporting EEP
payload	object, command payload using keys from JSON representation of EEP

Request results

After making a request the application should examine the `tenant@email.com/v0/device/{device_id}/rpc/result` topic from EnOcean SmartStudio for result of the request.

```json

{ "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id}, "message": "Command successfully queued", "success": true } ```
```

key	type and meaning
request_id	integer, unique identifier of the RPC request
success	boolean, status of request
message	string, explanation of request's current state

If EnOcean SmartStudio can properly transcode the request and put the command in queue, the application will receive below result:

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id}, "success": true, "message": "RPC Command successf
```

Once data is sent from the queue to the device, the application will receive below result:

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id}, "success": true, "message": "RPC request sent to
```

Possible error messages

If the device type is not set bidirectional, or if the device is not paired

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id},   "success": false,   "message": "Device is not c
```

No gateway is available for the device or the gateway did not report USB information

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id},   "success": false,   "message": "Device is not y
```

The gateway websocket connection was closed before the southbound command could be sent to the device

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id},   "success": false,   "message": "WebSocket conne
```

RPC command validation failed

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id},   "success": false,   "message": "TypeError -> RP
```

RPC command contains invalid values

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id},   "success": false,   "message": "ValueError -> R
```

Equivalent to 500 error in HTTP

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id},    "success": false,   "message": "Unknown error
```

```
json { "request_id": 1, "device_id": {device_id}, "tenant_id": {tenant_id},    "success": false,   "message": "<any exception
```

4.3 Webhooks

4.3.1 Overview

EnOcean SmartStudio webhooks enable real-time data delivery from your EnOcean sensors to your HTTP server. When a sensor transmits data, SmartStudio automatically forwards the telegram data to your configured endpoint as an HTTP POST request.

This integration is ideal for:
- Real-time data processing and analytics
- Integration with existing business systems
- Custom application development
- Third-party platform connectivity

4.3.2 Network Requirements

All webhook communication from SmartStudio to your server is outbound over standard HTTP/HTTPS ports.
Ensure your server is accessible from the following endpoint:

- **Source:** ingress.enocean.cloud
- **Ports:** 80 (HTTP) or 443 (HTTPS)
- **Protocol:** HTTP POST requests

4.3.3 Configuration

Step 1: Access Webhook Settings

1. Log into EnOcean SmartStudio
2. Navigate to **Integrations**
3. Select **Webhook**

Step 2: Configure Connection Settings

1. **Server URL**: Enter your HTTP/HTTPS endpoint URL where you want to receive the data

2. **Authentication**: Choose the appropriate authentication method:

   - **None**: No authentication required
   - **Basic**: Username and password authentication
   - **Bearer**: Token-based authentication
   - **Custom Header**: Custom authentication header

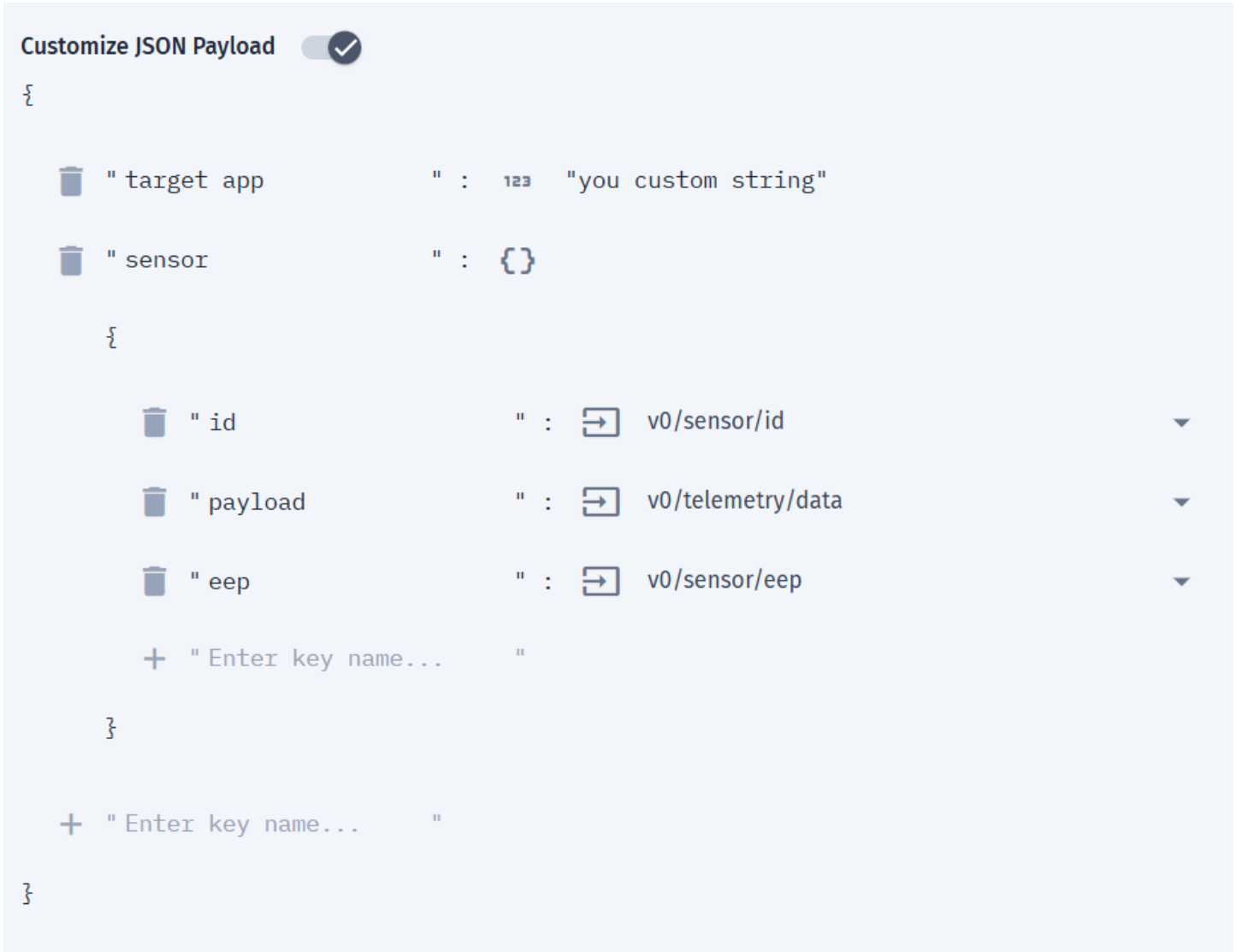
Step 3: Customize JSON Payload (Optional)

You can customize the JSON structure sent to your server:

1. Enable **Customize JSON payload** option
2. Configure key-value pairs where:
   - **Keys**: Custom string names for your data fields
   - **Values**: Can be any of the following:
     - SmartStudio telemetry objects (sensor ID, EEP profile, signal strength, etc.)
     - Static primitive values (strings, numbers, booleans)
     - Nested JSON objects

This allows you to tailor the webhook payload to match your server's expected data format.

Example Configuration:



Note

Customizations to the telemetry payload data values from the sensors is currently not possible.

Step 4: Save and Enable

1. Click **Update** to save your configuration changes
2. **Enable the integration** using the toggle switch at the top of the page

Important

The webhook integration will only start sending data after you enable it using the toggle switch. Don't forget this final step!

4.4 SmartServer IoT

4.4.1 Overview

The EnOcean SmartServer IoT is an edge controller providing connectivity for EnOcean radio devices and integration with Building Management Systems (BMS) and HVAC systems via BACnet, Modbus, and LON protocols. Installed on-premises, the SmartServer IoT enables secure connections between Operational Technology (OT) systems, IT systems, and the EnOcean SmartStudio platform.

The following describes how to configure the SmartServer IoT to pull data from SmartStudio and integrate it into Building Management Systems (BMS) and HVAC systems. How to configure the SmartServer IoT as a gateway for connecting EnOcean radio devices with SmartStudio is described here.

4.4.2 Prerequisites

Before deployment, ensure the following prerequisites are met:

1. The SmartServer is mounted and powered up.
2. The SmartServer is connected to the internet.
3. The SmartServer BACnet driver is enabled.

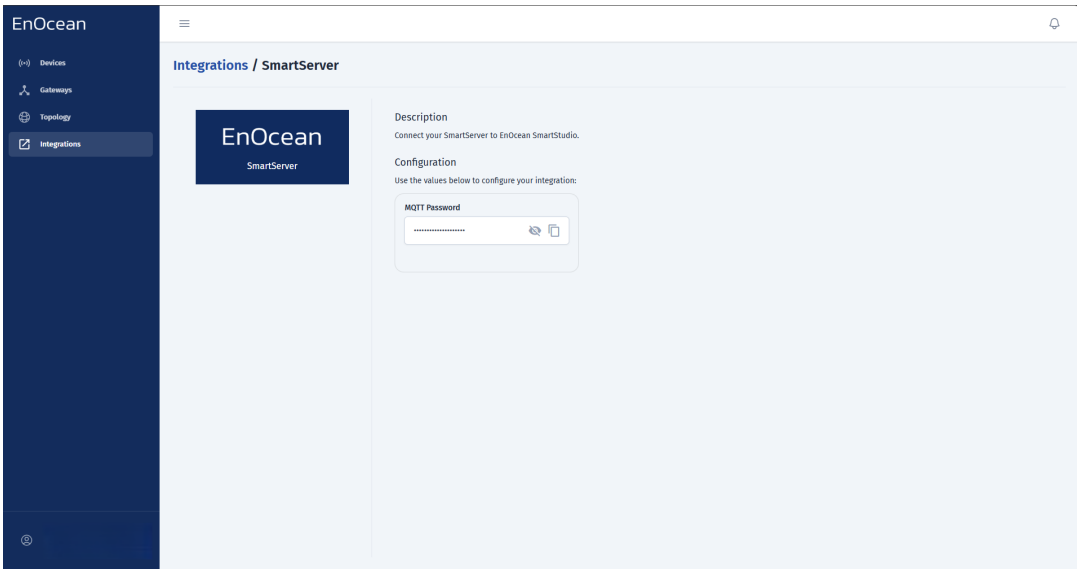
Note

Refer to the links below for the detailed steps on completing the prerequisites.
Set up SmartServer IoT
Setting up BACnet server on SmartServer IoT

4.4.3 Configuration

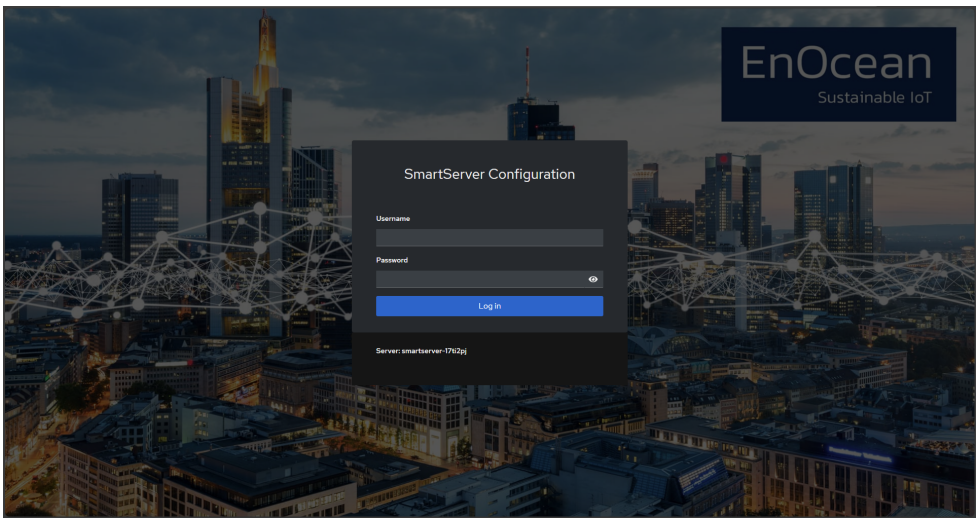
Step 1: Retrieve SmartStudio Connection Information

1. Retrieve the **MQTT Password** from SmartStudio:
   - Go to the **Integrations** section.
   - Select **SmartServer**.
   - Copy the information:



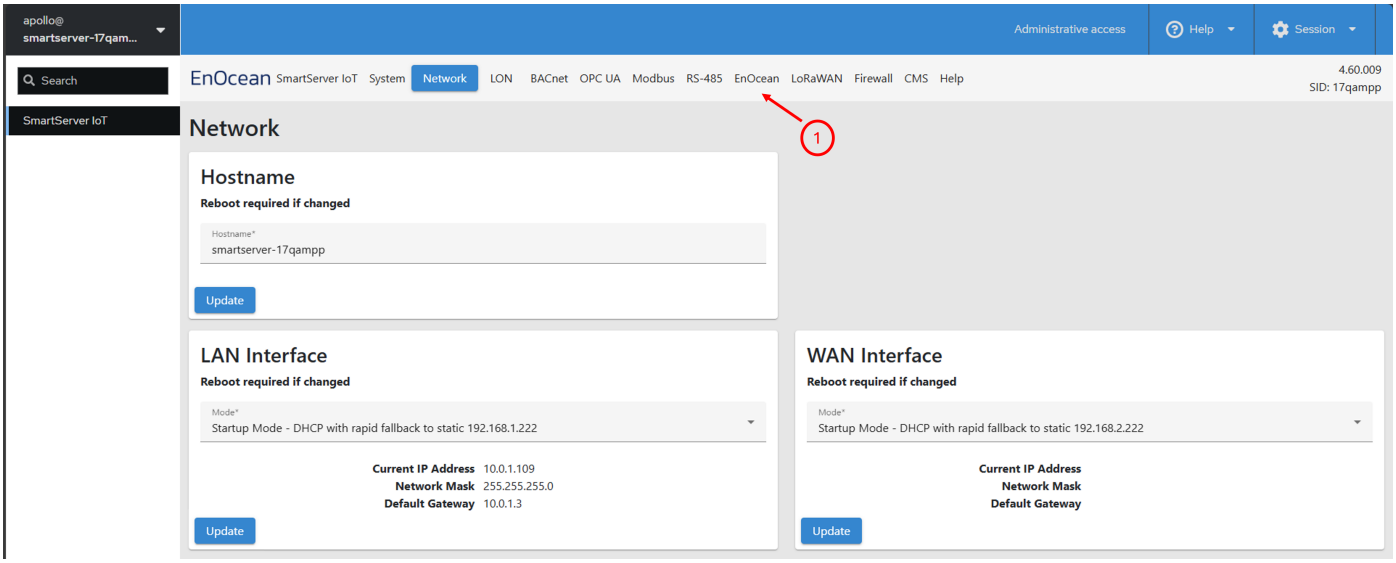
Step 2: Log in to the SmartServer configuration page

1. Open a web browser and navigate to the SmartServer's configuration page.
2. Log in using your apollo user credentials.



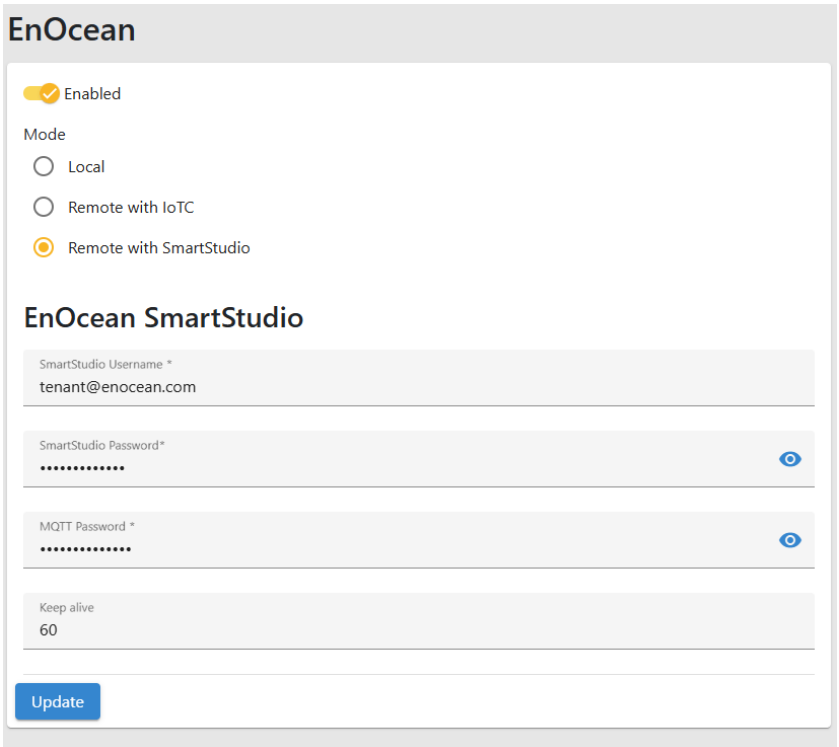
Step 3: Navigate to the EnOcean Tab

1. Once logged in, go to the **EnOcean** tab in the SmartServer configuration page.



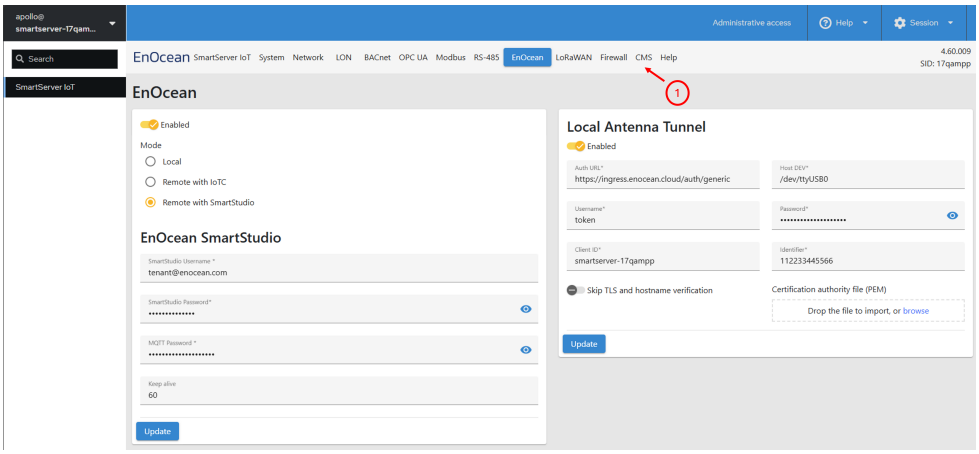
Step 4: Configure the SmartStudio Connection

1. Select the **Remote With SmartStudio** mode.
2. In the **Enocean SmartStudio** section, enter the information retrieved from SmartStudio above.
3. Click **Update** to enable remote mode with SmartStudio on your SmartServer IoT.

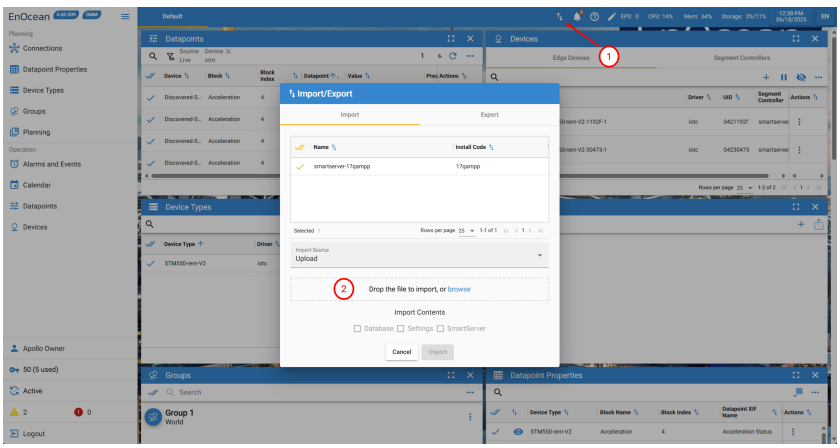


Step 5: Downloading EnOcean Device Interface (XIF) and Device Type Definitions

1. Log in to SmartServer CMS:



2. Visit the SmartServer GitHub repository.
3. Navigate to the appropriate region folder based on your deployment:
   - `EU/Remote` for European region
   - `US/Remote` for United States region
   - `JP/Remote` for Japanese region
4. Download the required .dtp (Device Type Package) and .btm (BACnet Type Mapping) files from the selected region remote folder.
5. Import the downloaded files in the step above using the Import / Export button on the SmartServer app bar.

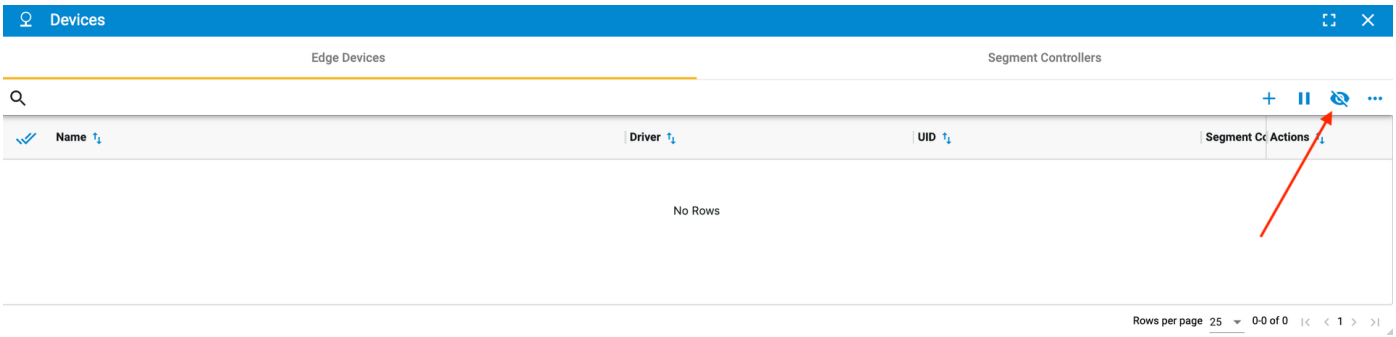


Once the EnOcean XIF and device type definitions are loaded, the device type definitions will appear in the Device Types widget.

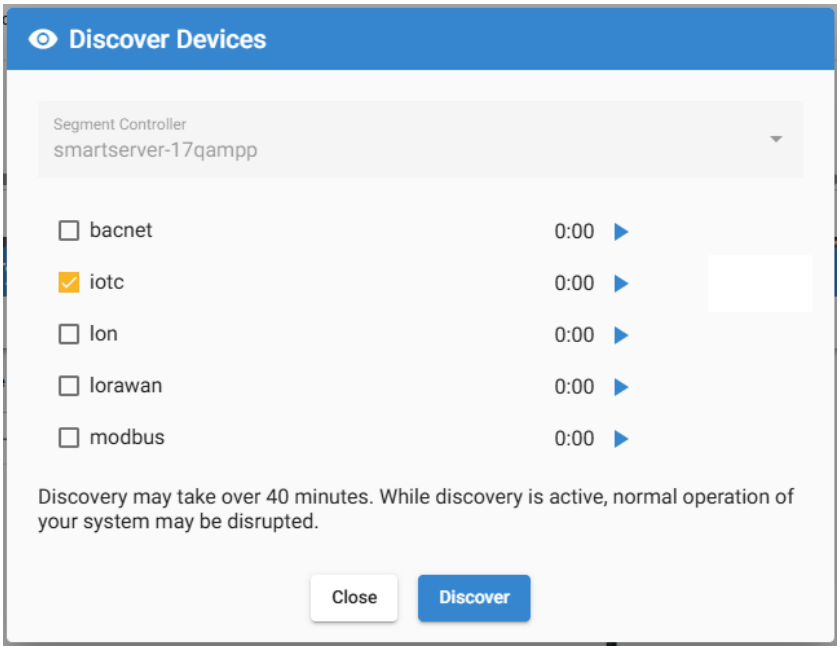
Step 6: Adding Remote EnOcean Devices

Once you have enabled remote mode and imported EnOcean device types, you can discover the EnOcean devices following below:

1. Open the devices widget.
2. Click the Discover button.



3. The Discover Devices dialog box appears as shown in the example below. Deselect all the options and keep only the iotc driver then click discover.



Note

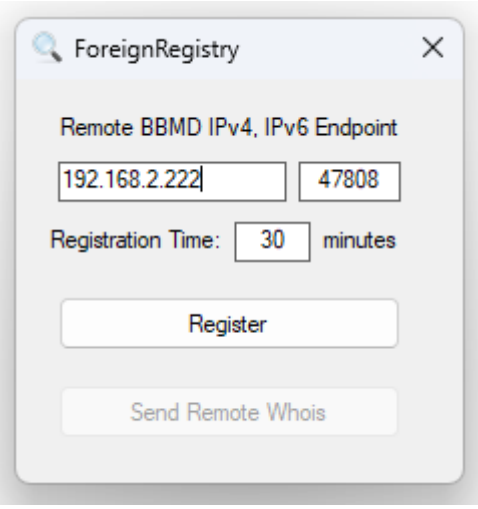
If you are using SmartSupervisor, then select the SmartServer where you want to initiate device discovery from the Segment Controller dropdown menu in the Discover Devices widget.

Step 7: Discover EnOcean Devices from a BACnet workstation

You can view and interact with the EnOcean devices connected to your SmartStudio account using a BACnet client. The following is an example of this using Yabe:

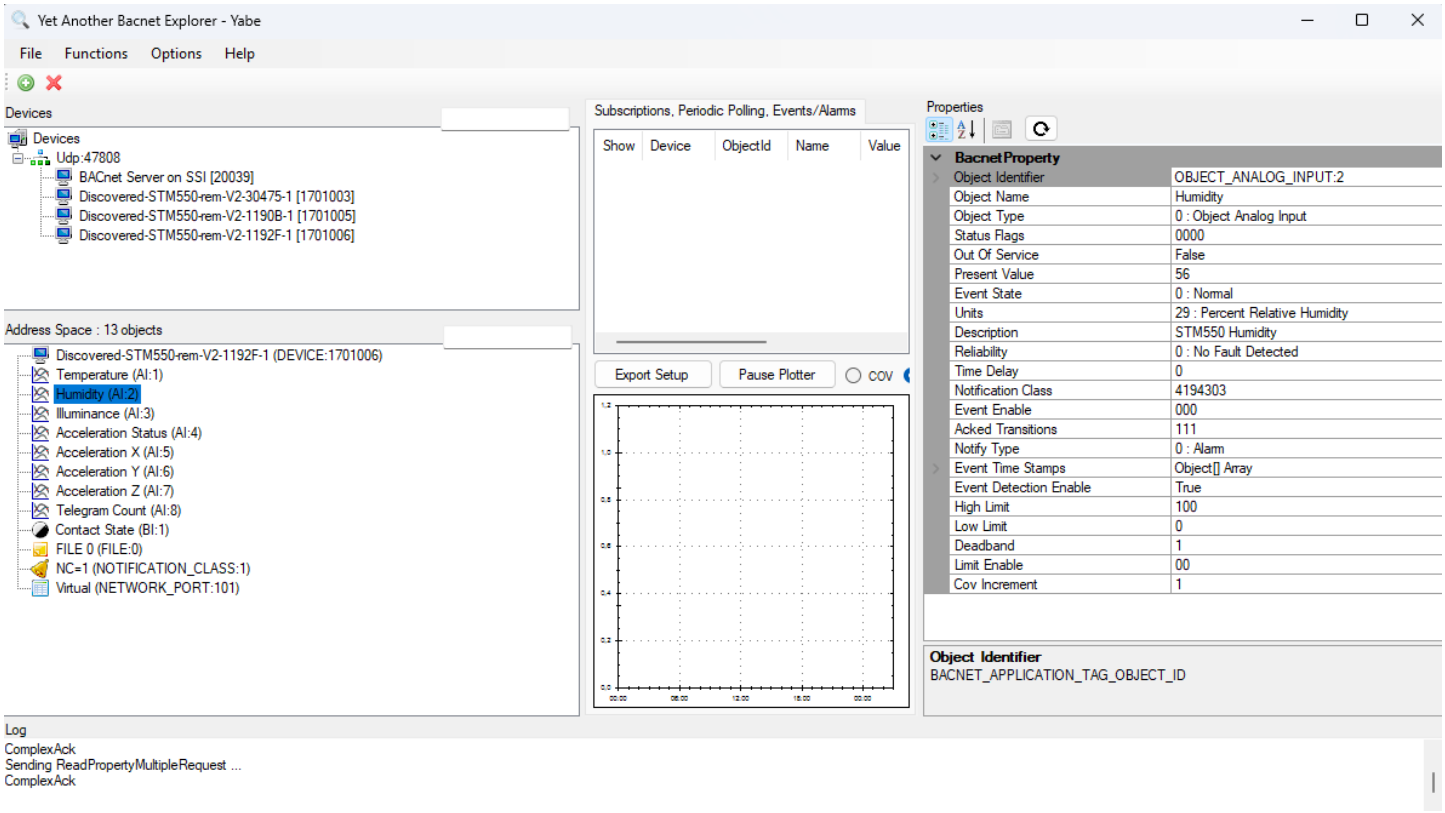
1. Launch **Yabe**.
2. Click **Add device (Search)**.
3. Under **BACnet/IP**, set the **Local endpoint** to your computer's IP address and use port `BAC0`. Click **Start**.
4. In the top menu, navigate to **Functions → IP Services → Foreign Device Registration**.
5. Enter the following details:

    - **Remote IPv4 address**: Set this to your SmartServer's IP address for the `eth0` network interface.
    - **Port**: If you are using a BACnet MS/TP Router with your SmartServer IoT on the LAN port, the port number can be found under the BACnet tab in the SmartServer configuration UI. The default port is `47809` (labeled as `BAC1` by default).



6. Click **Register**.

Once registered, all SmartServer devices and any SmartStudio devices for which `.btm` files have been imported will appear in Yabe under BACnet devices, allowing you to browse and interact with them.



5. More

5.1 Release Notes

This page contains information on all major releases, new features, enhancements, bug fixes, and deprecated functionality. Be sure to check this page regularly to stay up to date with the latest changes.

5.1.1 2026.01 (Latest)

New Features and Integrations

- Device QR scanning via mobile phones for easy commissioning.
- Enforce Security option to only allow encrypted sensor data from connected devices.

5.1.2 2025.10

New Features and Integrations

- **Planon IWMS Integration**:
  - Seamlessly connect Planon IWMS platform with SmartStudio, enabling Planon customers to leverage EnOcean connected devices.

Improvements

- **Device Connection Monitoring**:
  - Improved device connection status notification and offline detection.
- **People Counting Service**:
  - Improved accuracy by automatically resetting people counts based on in-room motion sensor data.

Bug Fixes

- **EnOcean PTM Integration**:
  - Corrected decoding of EnOcean PTM sensor data.

5.1.3 2025.08

Improvements

- **EEP Support**:
  - Added support for EEP F6-05-01.
  - Added support for EEP D2-14-53.
  - Added support for EEP D2-14-5D.
- **Location Input Enhancement**:
  - Improved location filtering and display.
- **Device and Gateway State**:
  - Enhanced device and gateway status indicators.
- **Security & Encryption**
  - Improved CMAC verification and rolling code handling with better handling of SEC_TI and SEC_CDM packets.

Bug Fixes

- **RPC Control Issues**
  - Fixed RPC control failures on Aruba AOS 10 SmartStudio App.

5.1.4 2025.06

New Features

- **Webhook Integration**:
  - Added support for webhook integrations with customizable JSON payload schemas, authentication options, and real-time data forwarding capabilities.
- **Generic Tunnel Support**:
  - Implemented native Aruba AOS-10 tunnel application with improved BLE and EnOcean device handling.
- **Device Location Management**:
  - Enhanced topology assignment system allowing devices to be assigned to specific topology assets with improved location string matching.

- **People Counter Support**:
  - Added dedicated endpoint for people counter devices using EEP A5-12-00.
- **Support for Sub-desk mounted Occupancy Sensor**:
  - Implemented desk occupancy support for The Sub-desk mounted Occupancy devices using EEP A5-07-01.
- **TLS Support for MQTT**:
  - Added TLS support for MQTT using 8883 port.

Improvements

- **Gateway Management**:
  - Fixed duplicate gateway issues from Aruba 10 APs and improved active/inactive gateway tracking.
- **UI/UX**:
  - Implemented responsive design improvements.

Bug Fixes

- **TRV Commands**:
  - Fixed issue where TRV commands were being sent constantly after first request.
- **Device Updates**:
  - Resolved device creation issues when location UUIDs were invalid.
- **Integration States**:
  - Fixed webhook and MQTT broker integration state management and configuration validation.
- **Telemetry Processing**:
  - Corrected telegram counts for STM-550B devices and improved telegram deduplication.

5.1.5 2025.03

New Features

- **New Undagrid Integration**:
  - Undagrid is now available as a new type of integration in EnOcean SmartStudio.
- **New MQTT Client Integration**:
  - Implemented MQTT broker and client integration with support for:
    - Customizable MQTT topics
    - TLS authentication and certificate validation
- **BLE Support**:
  - Added comprehensive Bluetooth Low Energy (BLE) support through Aruba for the following devices:
    - EMDCB devices
    - STM550B devices

API Improvements

- Implemented activity count endpoint for EPAC devices.
- Implemented energy management endpoint for CT Clamp devices.
- Enhanced occupancy tracking and heatmap functionality to provide more accurate real-time data visualization for space utilization.

Other Enhancements

- Allowed EURID in addition to UUID in v0 requests.

Bug Fixes

- Fixed RSSI calculation.
- Fixed page loading issues related to null configurations.
- Fixed integration handling within tenant onboarding/offboarding.
- Fixed duplicate detection and telegram counting.
- Fixed device EEP and EURID validation.
- Fixed WSS URLs in integrations.
- Prevented best RSSI calculation from converging to all-time best.
- Fixed parsing of regular serial data.

5.1.6 2025.01

New Features

- **New User Interface**:
  - Introduced a new UI for managing devices and gateways.
  - Added support for integrating different types of devices and services through the new UI, including SmartServer Local and Active Antenna integrations.
- **EEP Support**:
  - Added support for EEP D2-14-58.
  - Added support for EEP D2-14-59.
  - Added support for EEP D2-14-5c.

Bug Fixes

- Resolved a bug causing the RSSI to be shown from different gateways instead of the closest one.

5.1.7 2024.10

New Features

- **Authentication Improvements**:
  - Introduced refresh token support in `/auth/accessToken` for extended session management.

MQTT Topic Changes

- **Device and Gateway Data**:
  - Updated the MQTT topics used for device and gateway data streams to improve clarity and organization:
    - Device data topic changed from `sensor/{deviceId}/#` to `tenant@email.com/v0/sensor/{deviceId}/#`.
    - Gateway data topic changed from `gateway/{mac}/#` to `tenant@email.com/v0/gateway/{mac}/#`.

Gateways Connection Changes

- **Connection Parameters**:
  - Server URL is updated to `ingress.enocean.cloud`.
  - Connection port is updated to the default https port `443`.

Bug Fixes

- Resolved a bug causing `/devices/{deviceId}/signal` to return incorrect signal strength values for certain devices.

Other Improvements

- Enhanced API response times for the `/gateways` endpoints by optimizing database queries.
- Improved error handling across all endpoints to provide more informative error messages.

5.2 EnOcean SmartStudio Support

For support requests please contact EnOcean support from here.

Alternatively, contact support@enocean.com.