How to Connect Milesight Gateway to the Exosite IoT platform

Version Change Log			
Version	Revision Date	Revision Details	Revised By
V1.0	20240430	Initial	lockon

Platform Introduction

The ExoSense condition monitoring application can be deployed off the shelf and configured for your machines, equipment and assets. Use ExoSense along with Exosite's Device IoT Connectors, all of which run on our scalable, secure IoT Platform.Exosite's Murano platform handles connected IoT devices, IoT data processing and storage, application hosting, and orchestration of applications, services, and integrations.

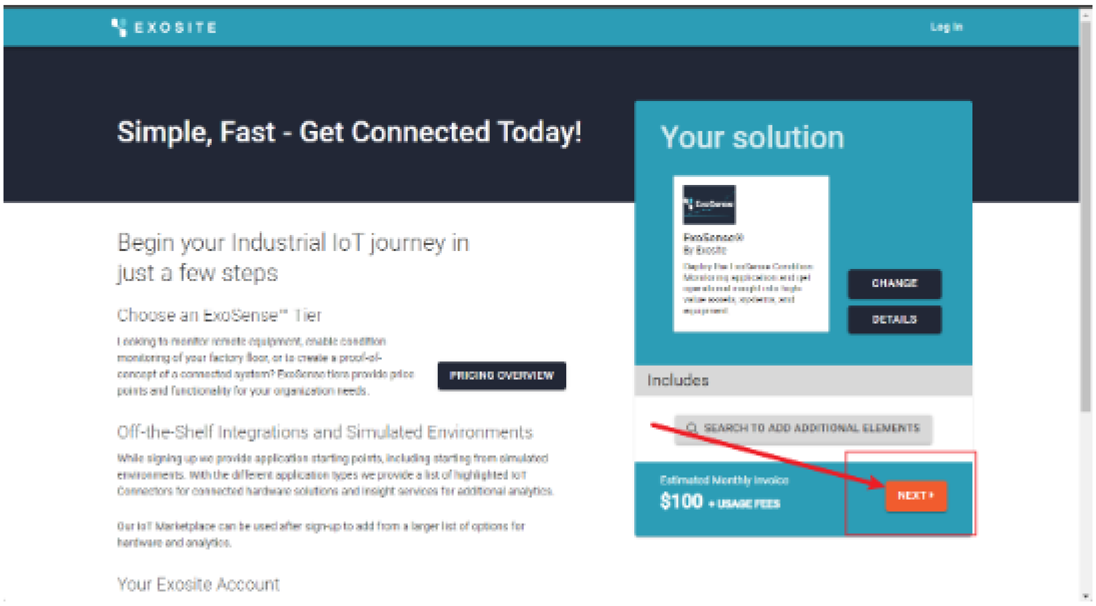
Prerequisites

1. Miesight UG56/UG63 V1/UG65/UG67 LoRaWAN Gateway
2. Milesight LoRaWAN End Devices (Sensors or Controllers)
3. Exosite Account: Sign-up and deploy ExoSense - Exosite Documentation

1. Exosite IoT Platform Configuration

1.1. Exosite IoT Platform Account Registration

Navitage to website Exosite - IoT Platform to register an account following the steps. Note that it is necessary to use a Visa card to complete the process successfully.



Simple, Fast - Get Connected Today!

Begin your Industrial IoT journey in just a few steps

Choose an ExoSense™ Tier

Looking to monitor remote equipment, enable condition monitoring of your factory floor, or to create a proof-of-concept of a connected system? ExoSense tiers provide price points and functionality for your organization needs.

Off-the-Shelf Integrations and Simulated Environments

While signing up we provide application starting points, including starting from simulated environments. With the different application types we provide a list of highlighted IoT Connectors for connected hardware solutions and insight services for additional analytics.

Our IoT Marketplace can be used after sign-up to add from a larger list of options for hardware and analytics.

Your Exosite Account

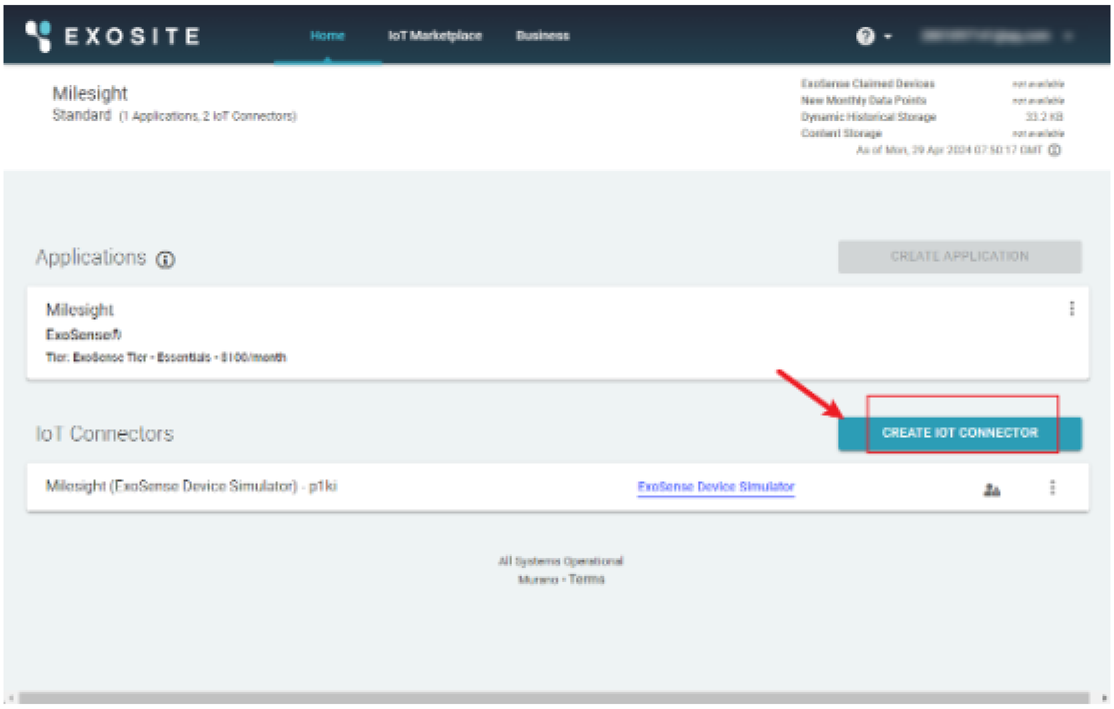
Your solution

ExoSense®
By Exosite
[illegible]

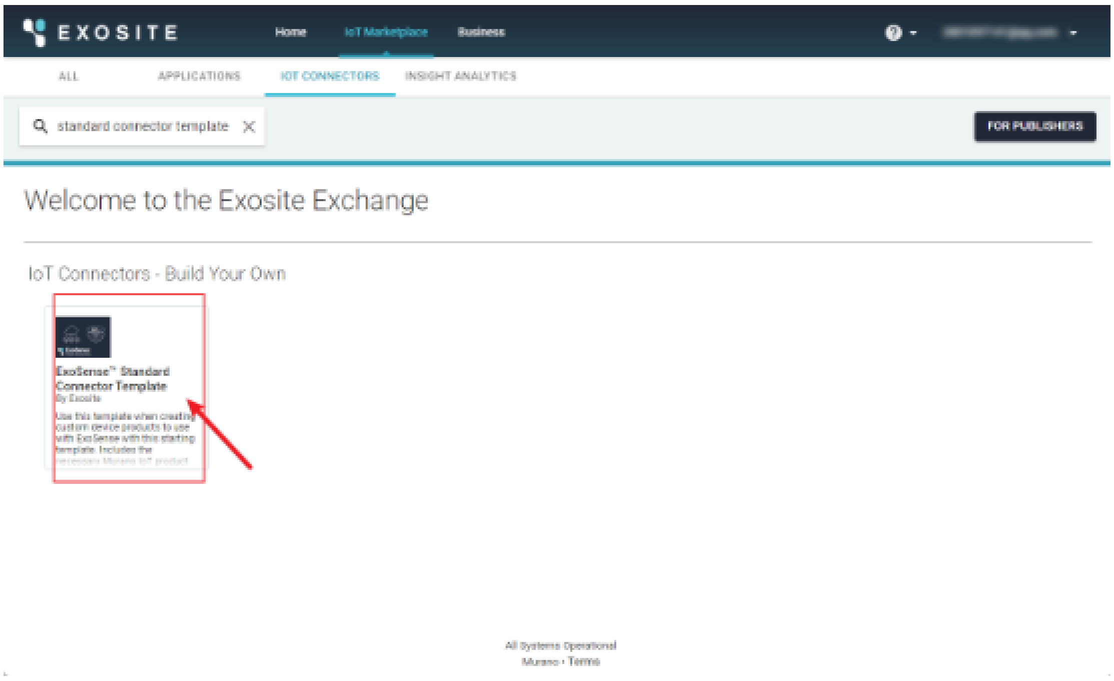
Estimated Monthly Invoice
$100 + USAGE FEES

1.2. Create IoT Connectors

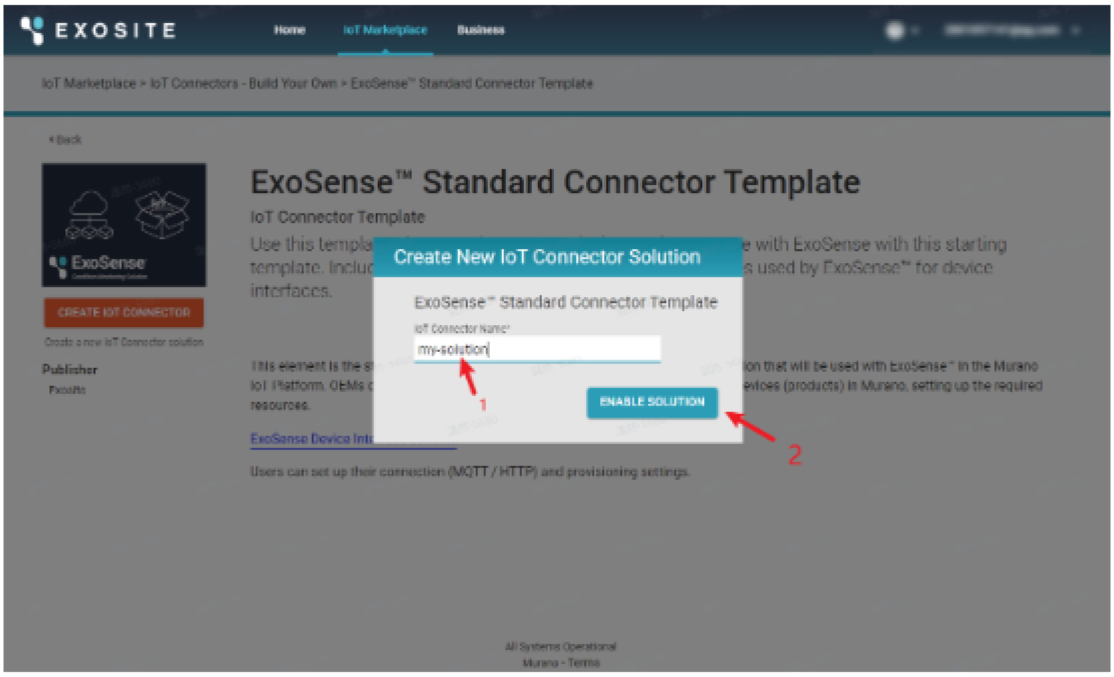
Log in the Exosite account, click **CREATE IOT CONNECTOR**.


EXOSITE
Home
IoT Marketplace
Business
[redacted]
Milesight
Standard (1 Applications, 2 IoT Connectors)
ExoSense Claimed Devices not available
New Monthly Data Points not available
Dynamic Historical Storage 33.2 KB
Content Storage not available
As of Mon, 29 Apr 2024 07:50:17 GMT
Applications
CREATE APPLICATION
Milesight
ExoSense®
Tier: ExoSense Tier - Essentials - $100/month
IoT Connectors
CREATE IOT CONNECTOR
Milesight (ExoSense Device Simulator) - p1ki
ExoSense Device Simulator
All Systems Operational
Murano · Terms

Locate the search box and enter 'Standard Connector Template', and select the template as shown in the picture:


EXOSITE
Home
IoT Marketplace
Business
[redacted]
ALL
APPLICATIONS
IOT CONNECTORS
INSIGHT ANALYTICS
standard connector template
FOR PUBLISHERS
Welcome to the Exosite Exchange
IoT Connectors - Build Your Own
ExoSense™ Standard Connector Template
By Exosite
Use this template when creating custom device products to use with ExoSense with this starting template. Includes the necessary Murano IoT product
All Systems Operational
Murano · Terms

Click **CREATE IOT CONNECTOR**, then in the pop-up window, enter a connector name and click **ENABLE SOLUTION**.

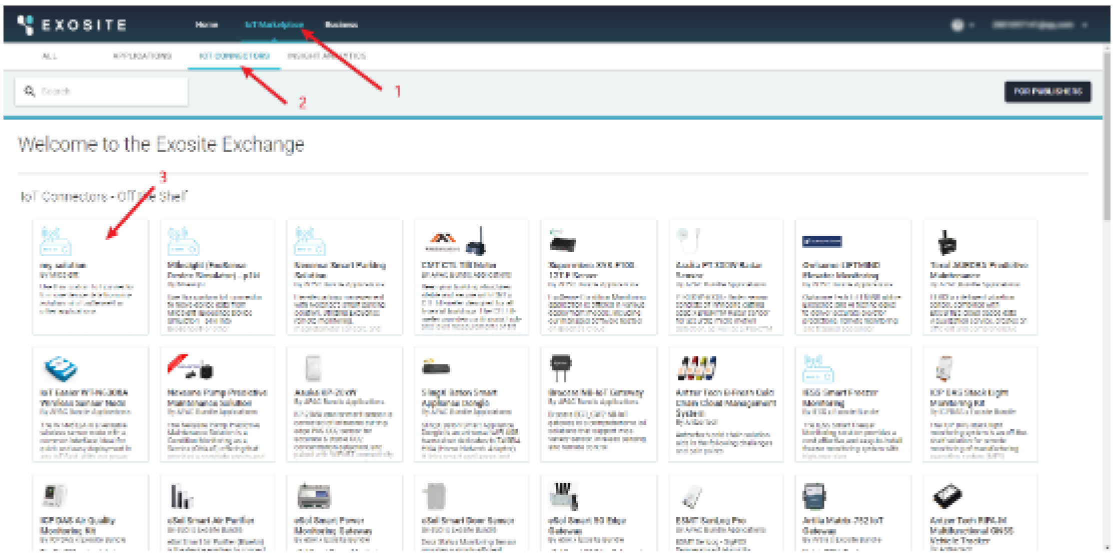

EXOSITE
Home
IoT Marketplace
Business
IoT Marketplace > IoT Connectors - Build Your Own > ExoSense™ Standard Connector Template
Back
ExoSense
CREATE IOT CONNECTOR
Create a new IoT Connector solution
Publisher
Exosite
ExoSense™ Standard Connector Template
IoT Connector Template
Create New IoT Connector Solution
ExoSense™ Standard Connector Template
IoT Connector Name*
my-solution
1
ENABLE SOLUTION
2
ExoSense Device Int
All Systems Operational
Murano - Terms

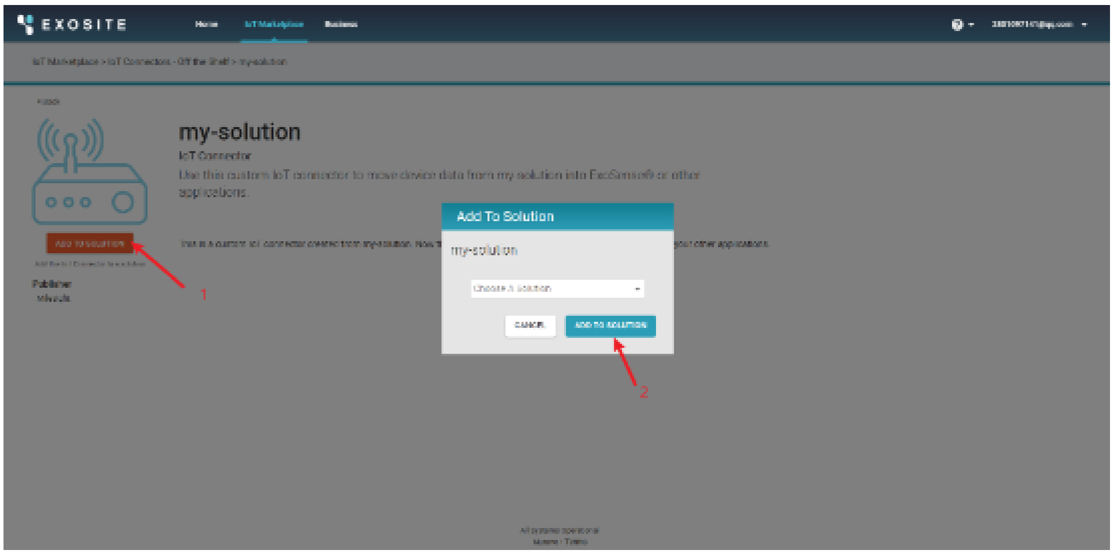
Use this templa… …e with ExoSense with this starting template. Incluc… …s used by ExoSense™ for device interfaces.

This element is the s… …ion that will be used with ExoSense™ in the Murano IoT Platform. OEMs c… …evices (products) in Murano, setting up the required resources.

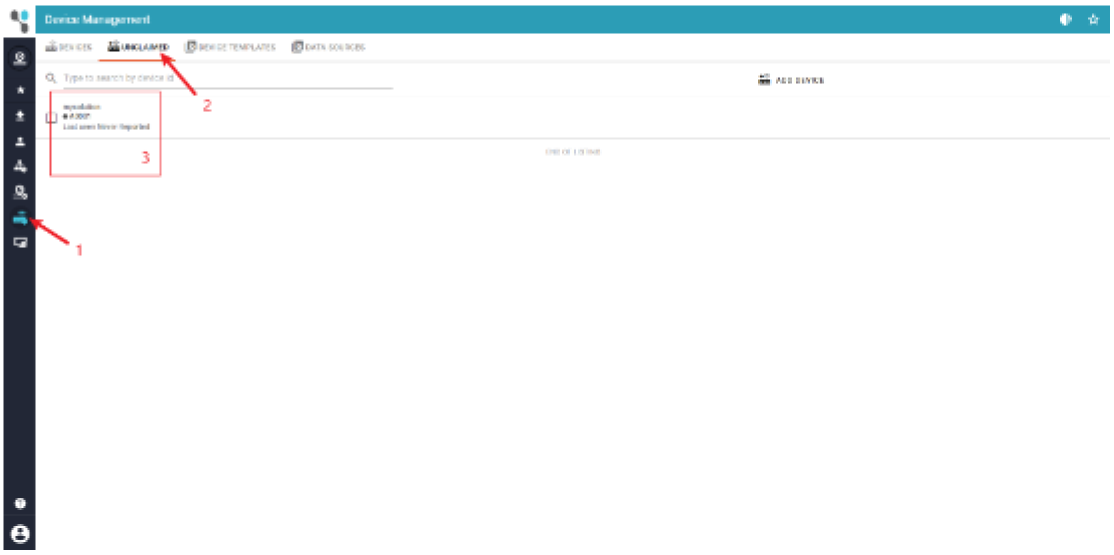
Users can set up their connection (MQTT / HTTP) and provisioning settings.

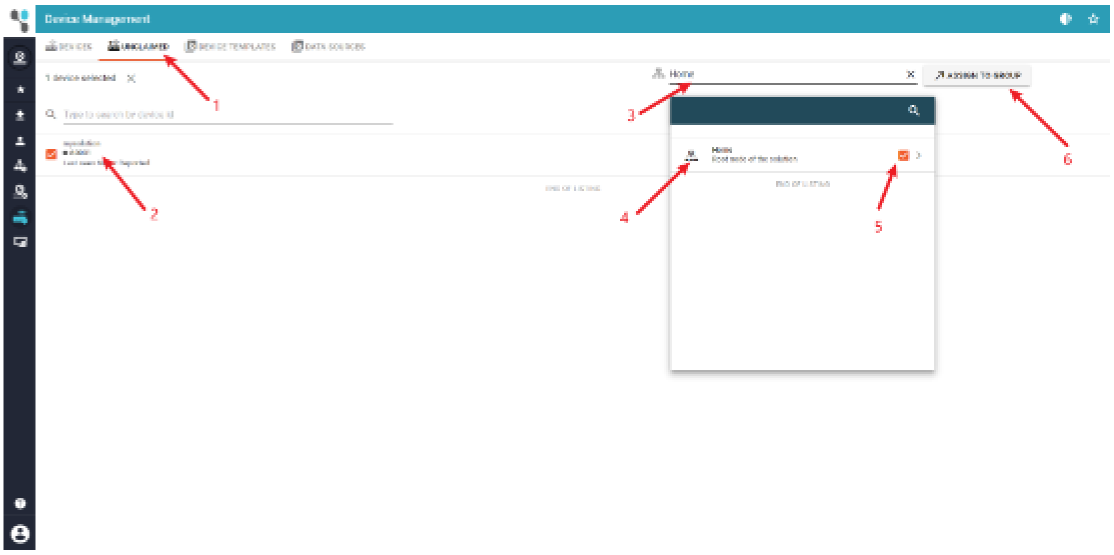
Add to solution as follows:



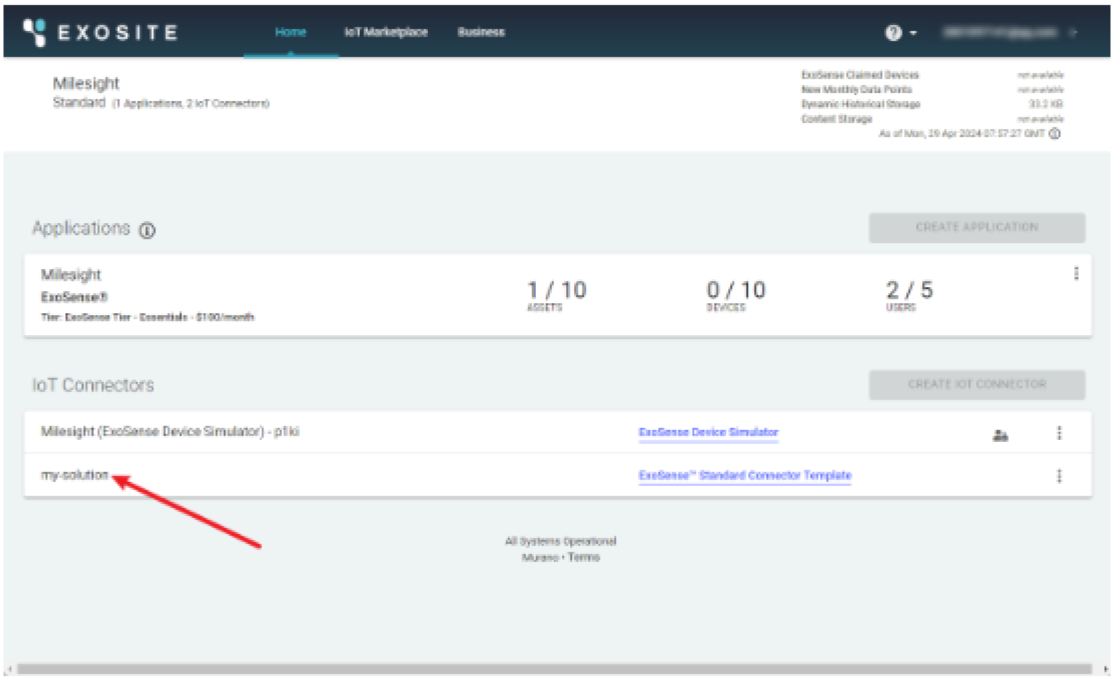


After that, login https://milesight.apps.exosite.io/ using the same username and do as follows :





The page automatically redirected as shown in the following picture:



Milesight
Standard (1 Applications, 2 IoT Connectors)

ExoSense Claimed Devices	not available
New Monthly Data Points	not available
Dynamic Historical Storage	33.2 KB
Content Storage	not available

As of Mon, 29 Apr 2024 07:57:27 GMT

Applications

Milesight ExoSense® Tier: ExoSense Tier - Essentials - $100/month	1 / 10 ASSETS	0 / 10 DEVICES	2 / 5 USERS

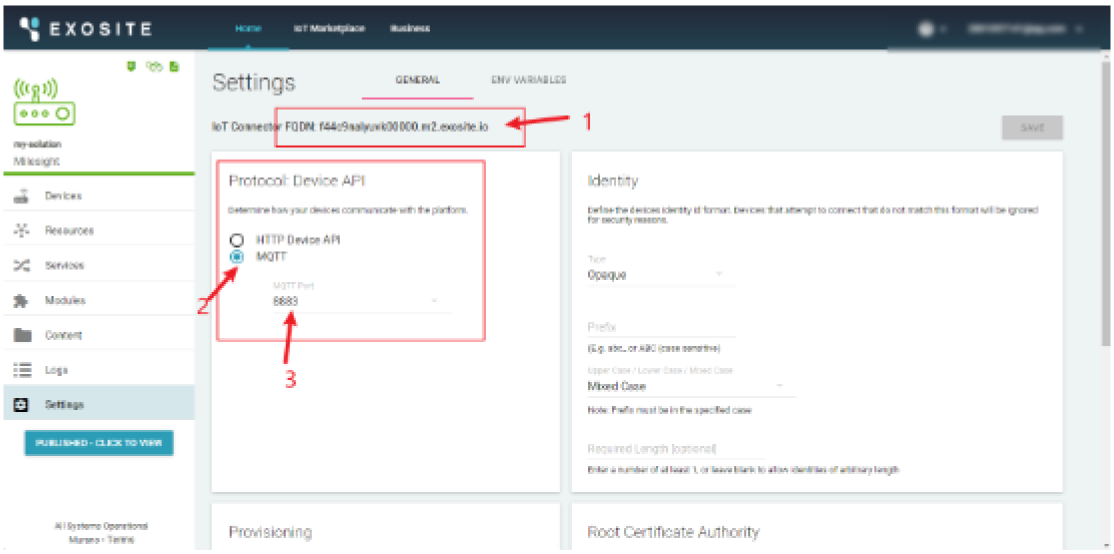
IoT Connectors

Milesight (ExoSense Device Simulator) - p1ki	ExoSense Device Simulator
my-solution	ExoSense™ Standard Connector Template

Click on the location indicated by the arrow to navigate to the configuration web page and begin the configuration process.

1.3. Platform Interface Settings

Locate the website configuration interface under 'Settings' and record the following relevant configuration parameters:



Settings

IoT Connector FQDN: f44c9aalyuvk00000.m2.exosite.io

Protocol: Device API

Determine how your devices communicate with the platform.

HTTP Device API [✓ MQTT]

MQTT Port: 8883

Identity

Define the devices identity id format. Devices that attempt to connect that do not match this format will be ignored for security reasons.

Type: Opaque

Prefix

(E.g. abc_ or ABC (case sensitive))

Upper Case / Lower Case / Mixed Case: Mixed Case

Note: Prefix must be in the specified case

Required Length (optional)

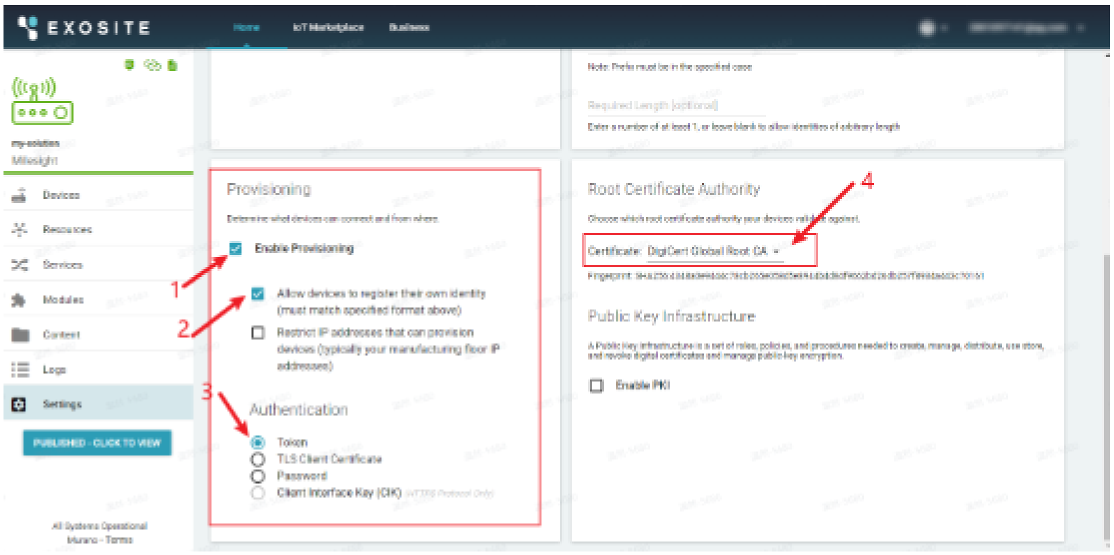
Enter a number of at least 1, or leave blank to allow identities of arbitrary length

The FQDN address here is the platform's external domain address, as shown in the picture. You need to change the Device API type to MQTT and select port 8883. The key parameters used in this case are as follows:

FQDN Address: <iot connector id>.m2.exosite.io

MQTT Port: 8883

Scroll down the current page to continue the configuration, as shown in the picture:



Note: Prefix must be in the specified case

Required Length (optional)

Enter a number of at least 1, or leave blank to allow identities of arbitrary length

Provisioning

Determine what devices can connect and from where.

☑ Enable Provisioning

☑ Allow devices to register their own identity (must match specified format above)

☐ Restrict IP addresses that can provision devices (typically your manufacturing floor IP addresses)

Authentication

● Token
○ TLS Client Certificate
○ Password
○ Client Interface Key (CIK) (HTTPS Protocol Only)

Root Certificate Authority

Choose which root certificate authority your devices validate against.

Certificate: DigiCert Global Root CA

Fingerprint: [illegible]

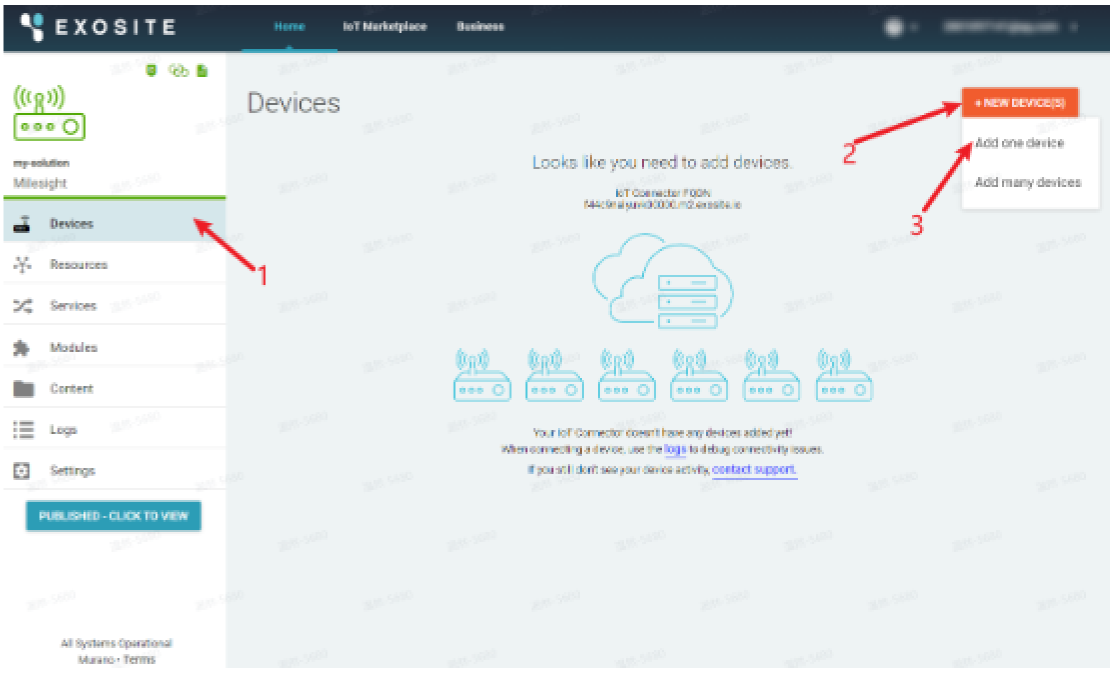
Public Key Infrastructure

A Public Key Infrastructure is a set of roles, policies, and procedures needed to create, manage, distribute, use store, and revoke digital certificates and manage public-key encryption.

☐ Enable PKI

Refer to the above picture for modifications. You need to enable the platform's Provisioning function and adjust the Authentication method to Token. Also, set the Root Certificate Authority to DigiCert Global Root CA, leaving the rest of the options as default. Once the configuration is complete, scroll up to the top of the current page and click the Save button to save the parameter configuration.It's important to note that all parameter configurations take effect immediately.

1.4. Creating Device Configuration



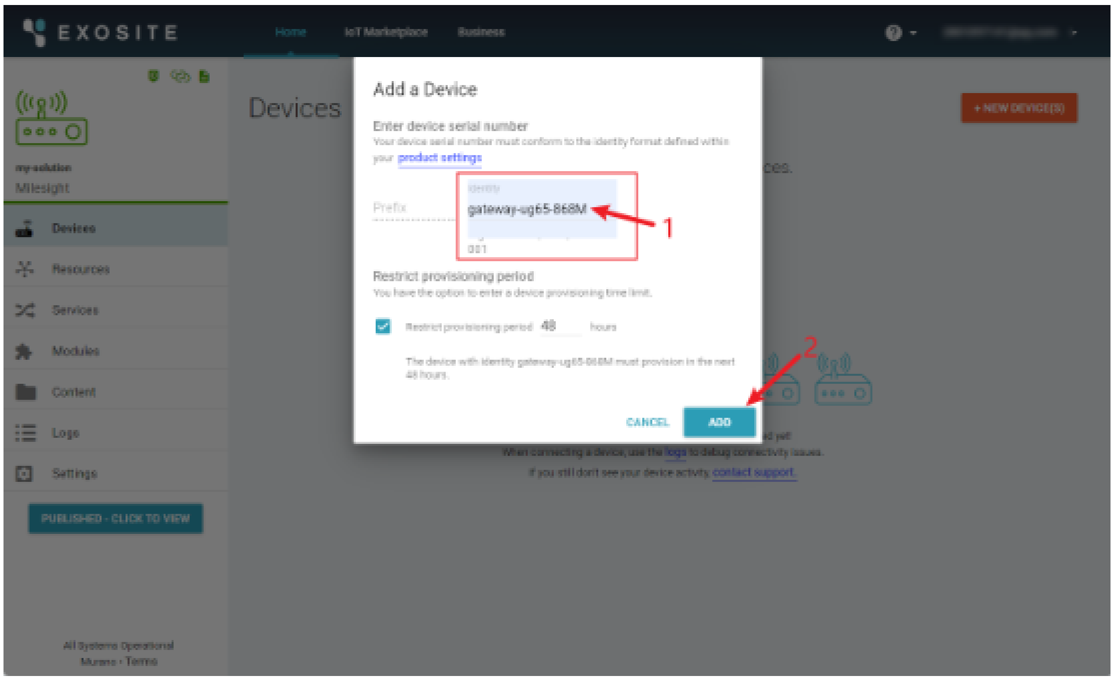
Devices

Looks like you need to add devices.

IoT Connector FQDN
[illegible].m2.exosite.io

Your IoT Connector doesn't have any devices added yet!
When connecting a device, use the logs to debug connectivity issues.
If you still don't see your device activity, contact support.

In the pop-up interface, this document takes the gateway device with the model UG65 as an example. Fill in the following information: 'gateway-ug65-868M'.



Add a Device

Enter device serial number
Your device serial number must conform to the identity format defined within your product settings

Prefix
Identity
gateway-ug65-868M
001

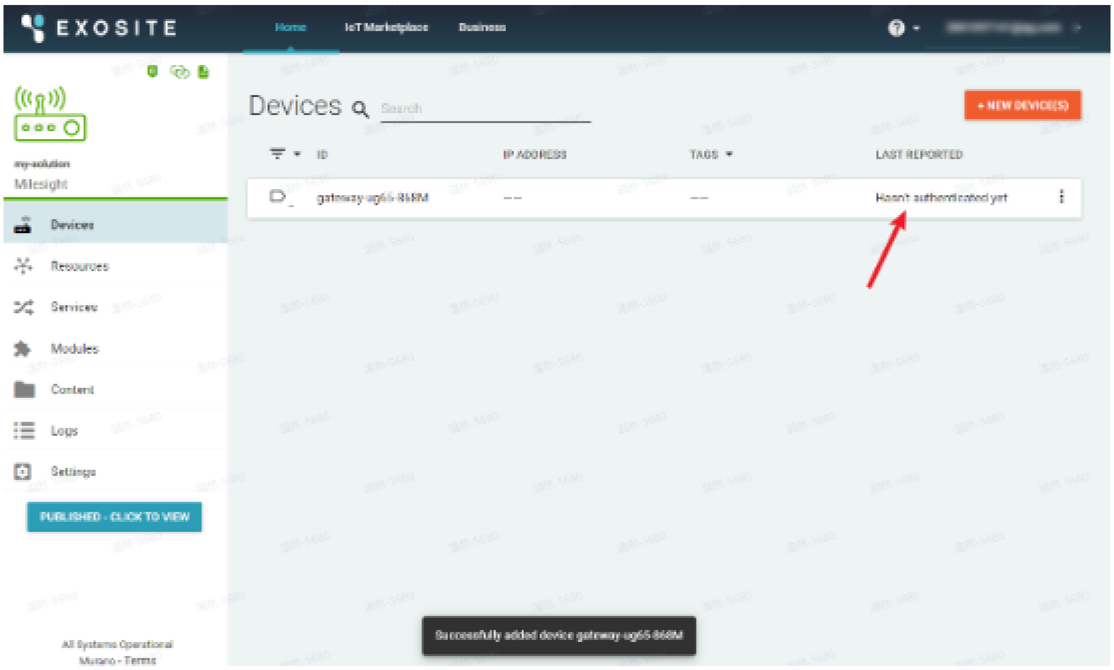
Restrict provisioning period
You have the option to enter a device provisioning time limit.

Restrict provisioning period 48 hours

The device with identity gateway-ug65-868M must provision in the next 48 hours.

When connecting a device, use the logs to debug connectivity issues.
If you still don't see your device activity contact support.

However, after adding, we see that the newly added device has not been activated.

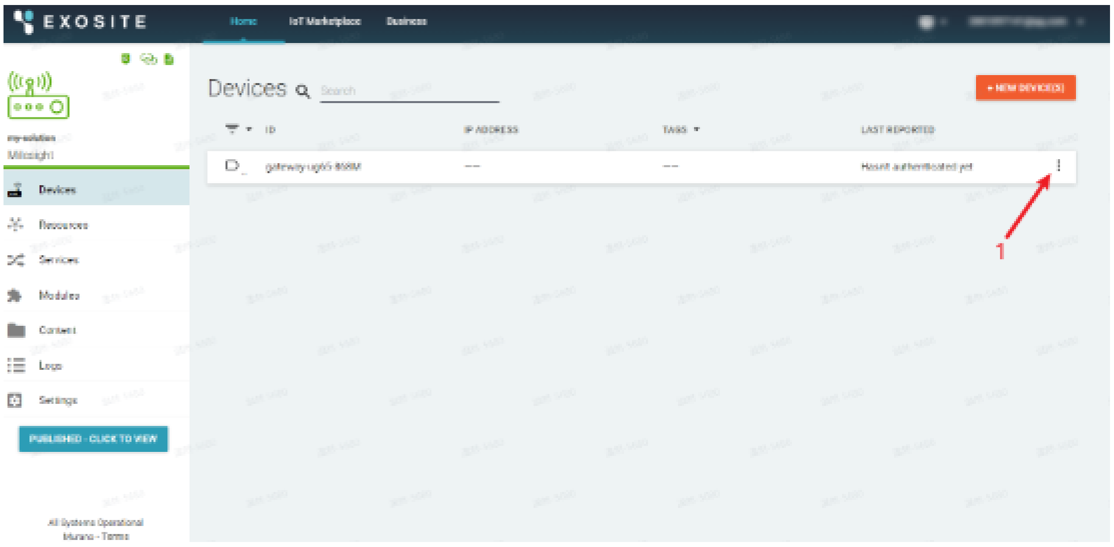


ID	IP ADDRESS	TAGS	LAST REPORTED
gateway-ug65-868M	---	---	Hasn't authenticated yet

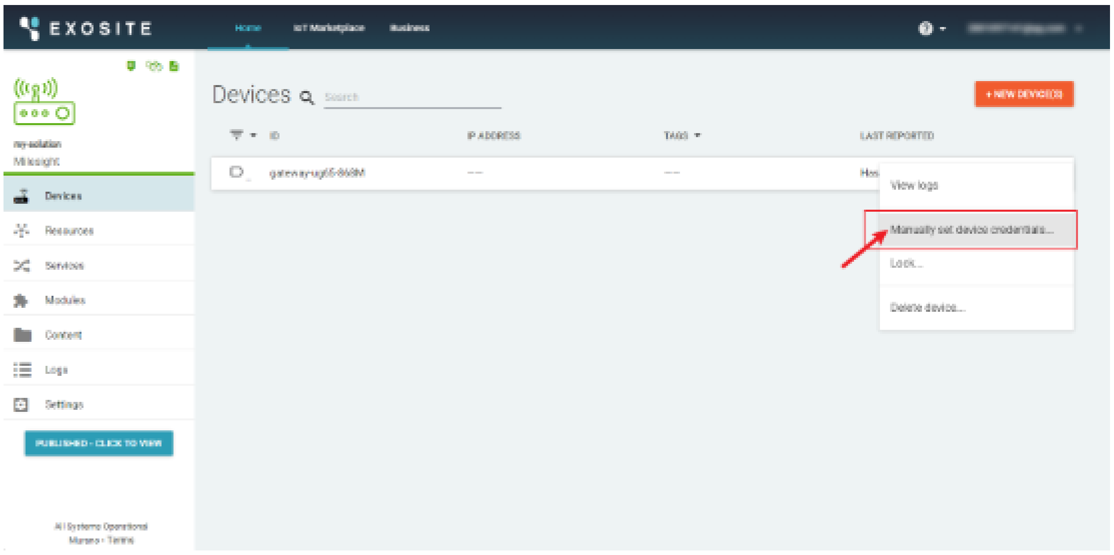
Successfully added device gateway-ug65-868M

1.5. Activate Device

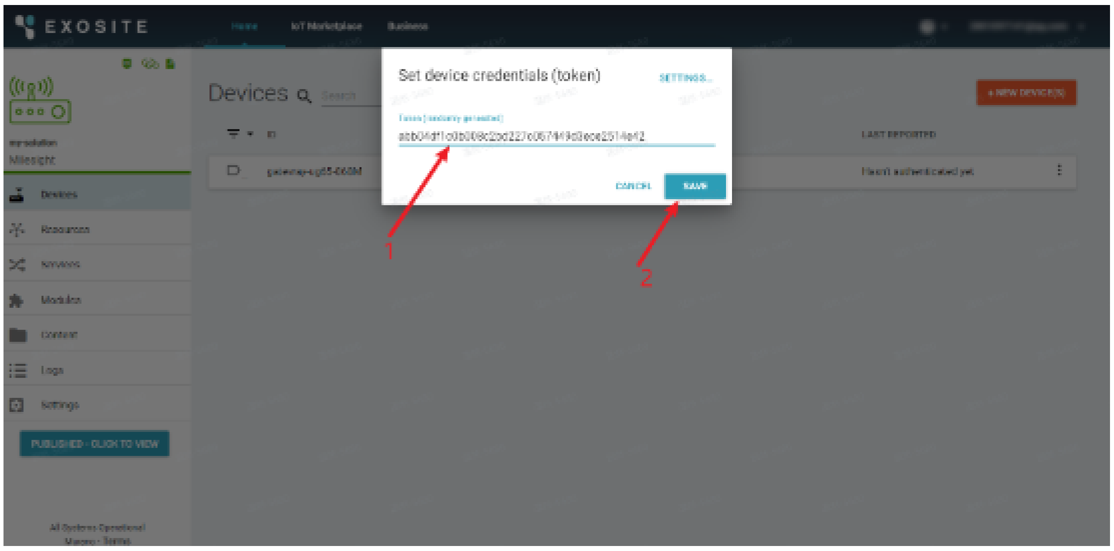
In the interface shown in the following picture, click on the location indicated by the arrow:



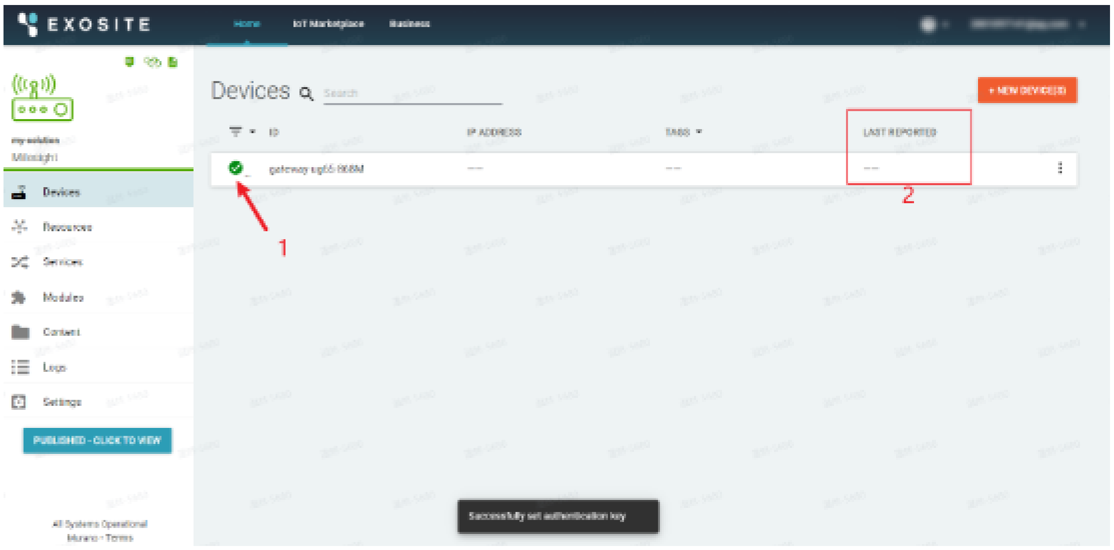
In the pop-up menu, select 'Manually set device credentials'.



The popped-up web interface is as follows:



Here, the token needs to be copied separately for future use, then click the Save button to close the current pop-up box.
The token shown here is abb04df1c0b008c2bd227c057449d3ece2514e42.
NOTICE: This operation will generate a new token every time and automatically invalidate the existing token.

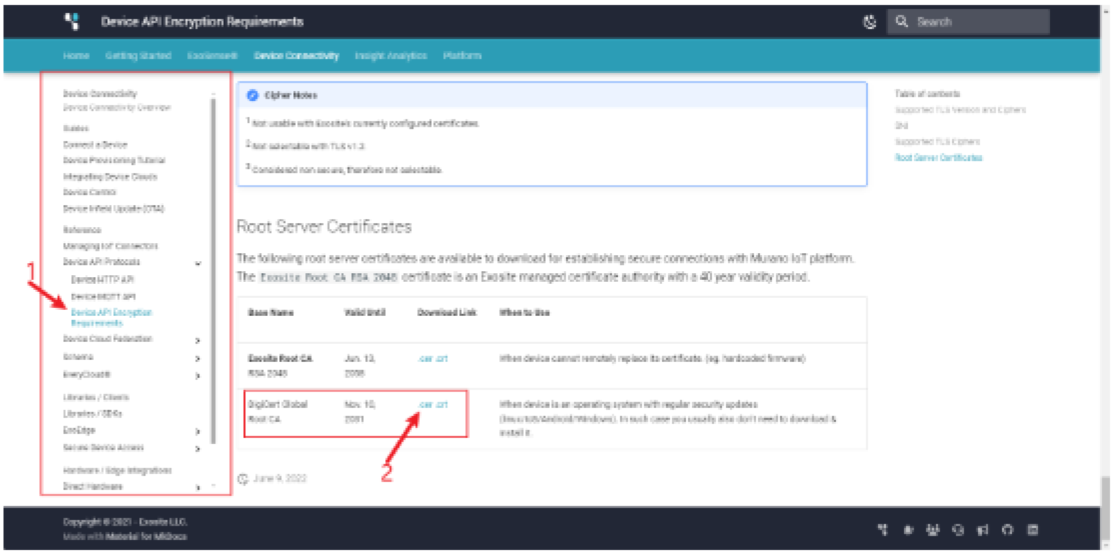


Successfully set authentication key

Here we see that the status of the newly added device is green, indicating that the device has been successfully created on the platform and its status is normal. Next, we will introduce how to configure the platform's external interface.

1.6. Download Platform Certificate

Download CA certifcate from Exosite website:
🔗 Device API Encryption Requirements - Exosite Documentation



Cipher Notes

1 Not usable with Exosite's currently configured certificates.

2 Not selectable with TLS v1.3.

3 Considered non secure, therefore not selectable.

Root Server Certificates

The following root server certificates are available to download for establishing secure connections with Murano IoT platform.
The `Exosite Root CA RSA 2048` certificate is an Exosite managed certificate authority with a 40 year validity period.

Base Name	Valid Until	Download Link	When to Use
Exosite Root CA RSA 2048	Jun. 13, 2058	.cer .crt	When device cannot remotely replace its certificate. (eg. hardcoded firmware)
DigiCert Global Root CA	Nov. 10, 2031	.cer .crt	When device is an operating system with regular security updates (linux/iOS/Android/Windows). In such case you usually also don't need to download & install it.

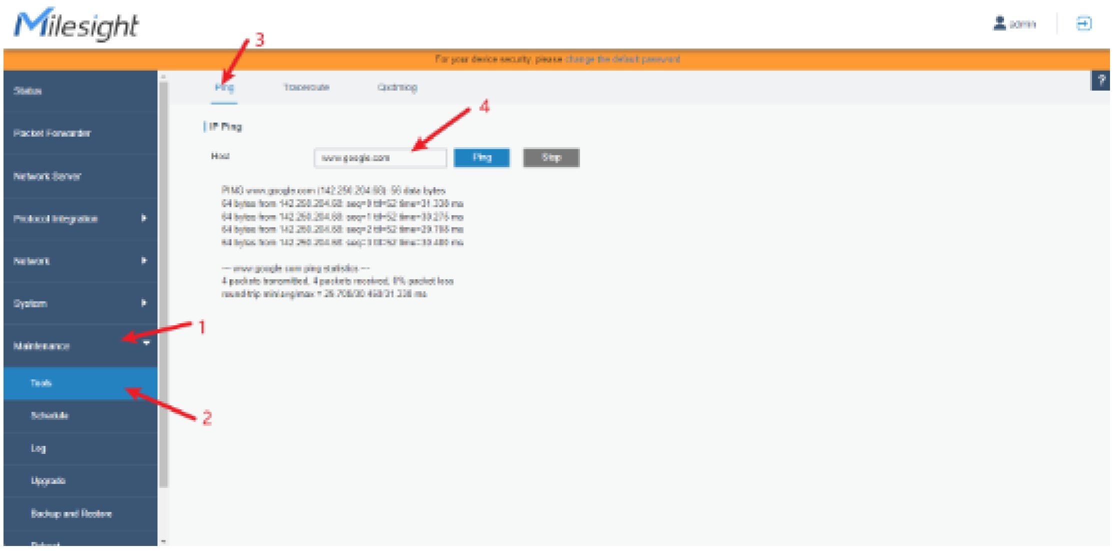
June 9, 2022

Copyright © 2021 - Exosite LLC.
Made with Material for MkDocs

2. Gateway Configuration

2.1. Configure the Gateway

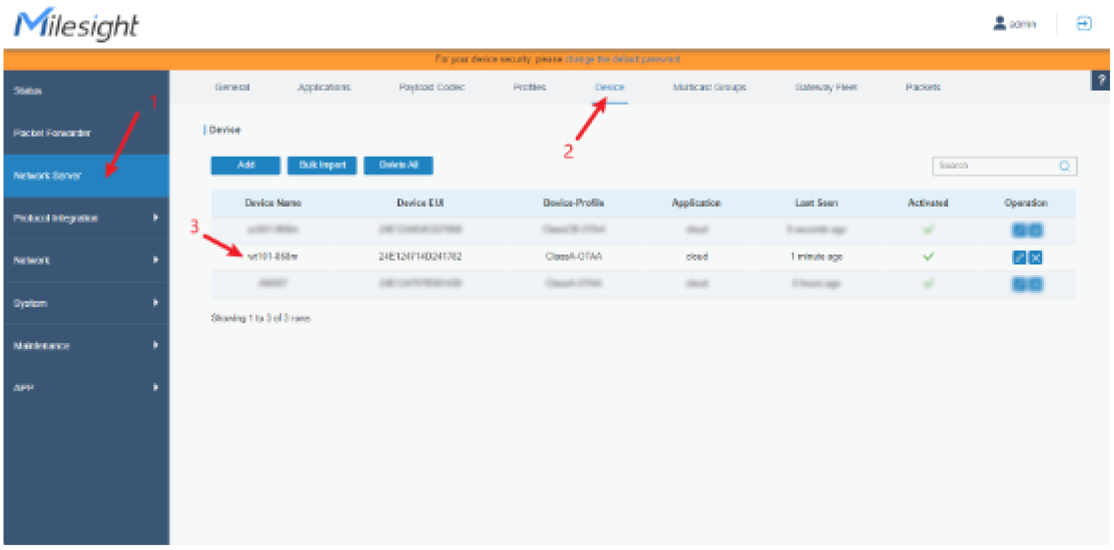
Ensure that the gateway device can access the internet properly. Navigate to **Tools > Maintenance > Ping** page to test the website 🔗 www.google.com to check network connectivity.



```
PING www.google.com (142.250.204.68): 56 data bytes
64 bytes from 142.250.204.68: seq=0 ttl=52 time=31.338 ms
64 bytes from 142.250.204.68: seq=1 ttl=52 time=30.276 ms
64 bytes from 142.250.204.68: seq=2 ttl=52 time=29.708 ms
64 bytes from 142.250.204.68: seq=3 ttl=52 time=30.480 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 29.708/30.458/31.338 ms
```

2.2. Add LoRaWAN Device(s)

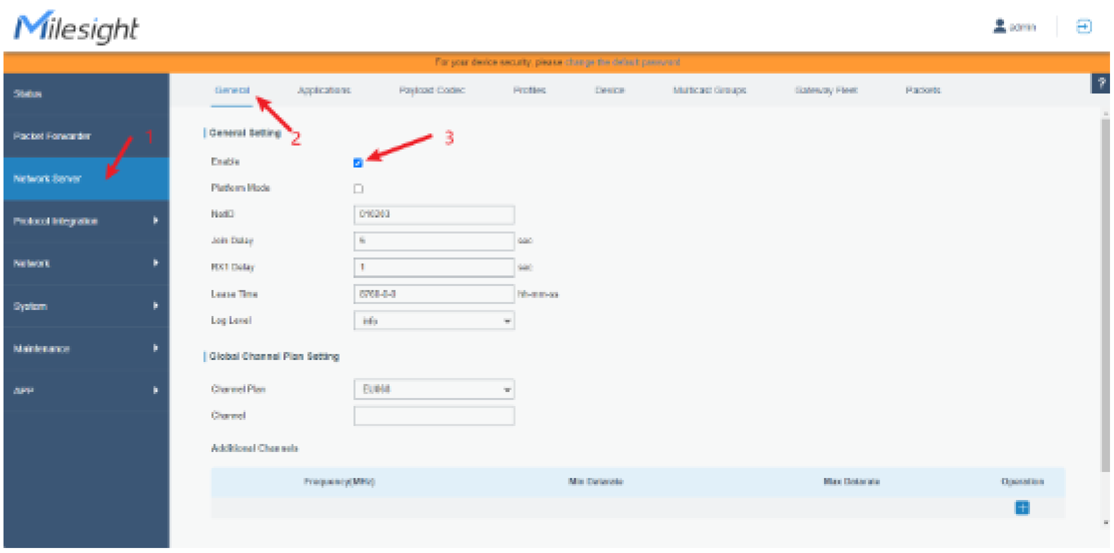
The process is straightforward, and the typical result is as shown in the picture.



Device Name	Device EUI	Device-Profile	Application	Last Seen	Activated	Operation
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	✓	
vs101-868m	24E124714D241782	ClassA-OTAA	cloud	1 minute ago	✓	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	✓	

Showing 1 to 3 of 3 rows

Note: The NS function of the gateway needs to be enabled.



General Setting

Enable	✓
Platform Mode	☐
NetID	010203
Join Delay	5 sec
RX1 Delay	1 sec
Lease Time	8760-0-0 hh-mm-ss
Log Level	info

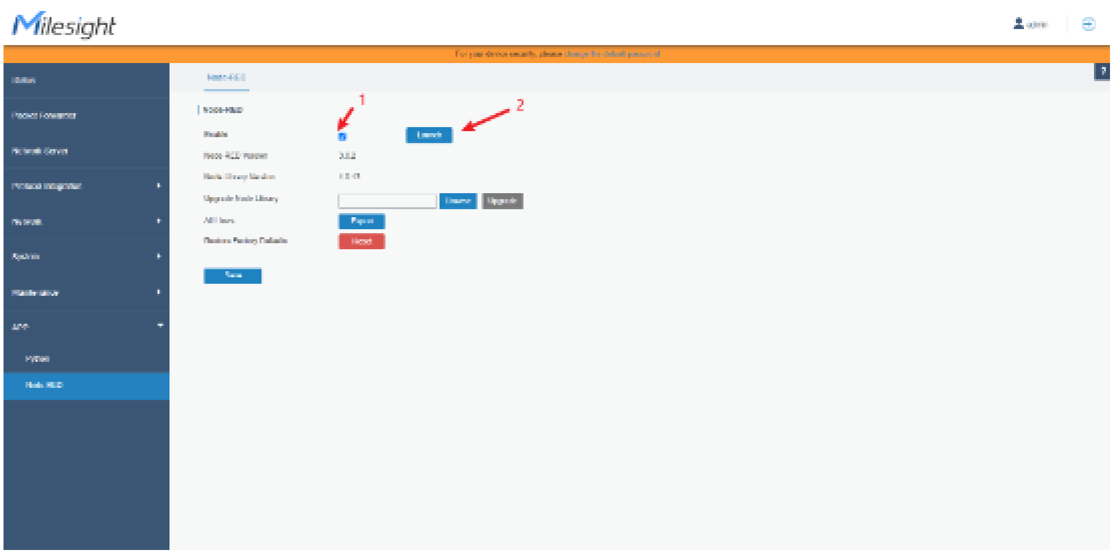
Global Channel Plan Setting

Channel Plan	EU868
Channel	

Additional Channels

Frequency(MHz)	Min Datarate	Max Datarate	Operation

Next, let's start configuring the Node-RED solution flow. Navigate to APP -> Node-RED in the gateway's management interface and follow the instructions in the picture.
Check the **Enable** option, then wait for 3-5 minutes, and click the **Launch** button to open the relevant configuration interface.

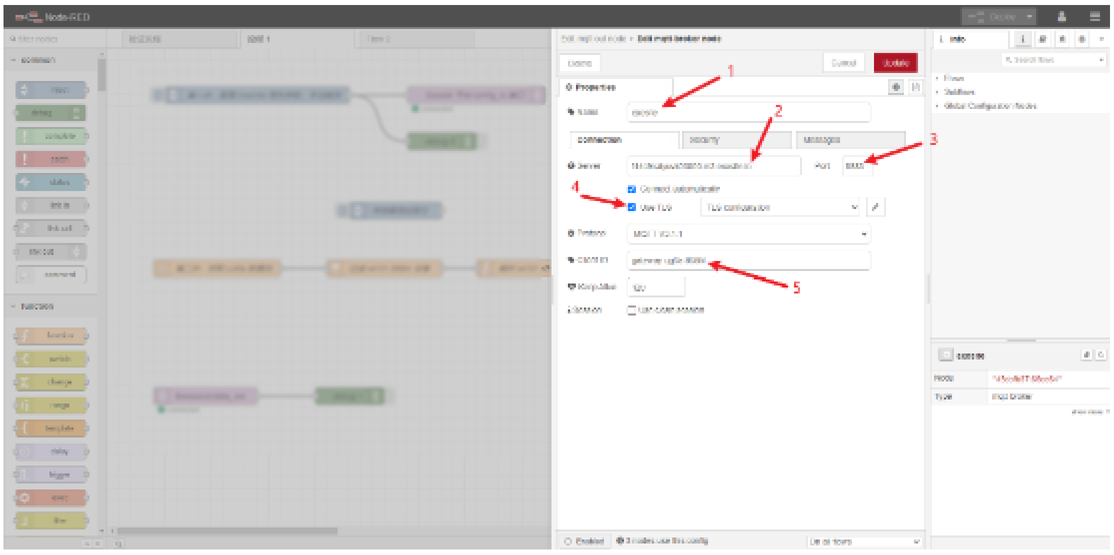


Node-RED

Enable	✓
Node-RED Version	3.0.2
Node Library Version	1.0.0
Upgrade Node Library	
API Docs	
Restore Factory Defaults	

2.3. Create MQTT connector in the Node-RED flow

Following the steps in the picture:


Edit mqtt out node > **Edit mqtt-broker node**

Delete | Cancel | Update

Properties

Name: exosite

Connection | Security | Messages

Server: [illegible] Port: [illegible]

☑ Connect automatically

☑ Use TLS — TLS configuration

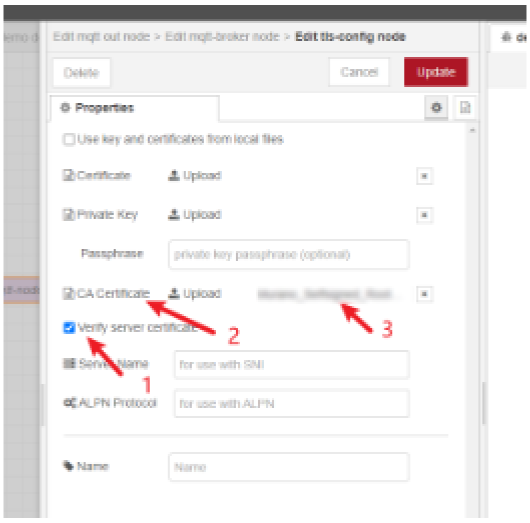
Protocol: MQTT V3.1.1

Client ID: [illegible]

Keep Alive: [illegible]

Session: ☐ Use clean session

NOTICE 1: Fill in the elements obtained from the previous steps, paying attention to the correct **FQDN** address , **port** number, and **Client ID**.
NOTICE 2: The MQTT connection tunnel must use **TLS certificates**. The specific certificate file mentioned earlier is the crt file. Please follow the instructions shown in the following image to proceed with the operation.


Edit mqtt out node > Edit mqtt-broker node > **Edit tls-config node**

Delete | Cancel | Update

Properties

☐ Use key and certificates from local files

Certificate: Upload

Private Key: Upload

Passphrase: private key passphrase (optional)

CA Certificate: Upload [illegible]

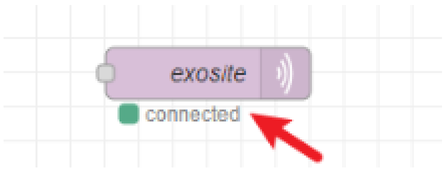
☑ Verify server certificate

Server Name: for use with SNI

ALPN Protocol: for use with ALPN

Name: Name

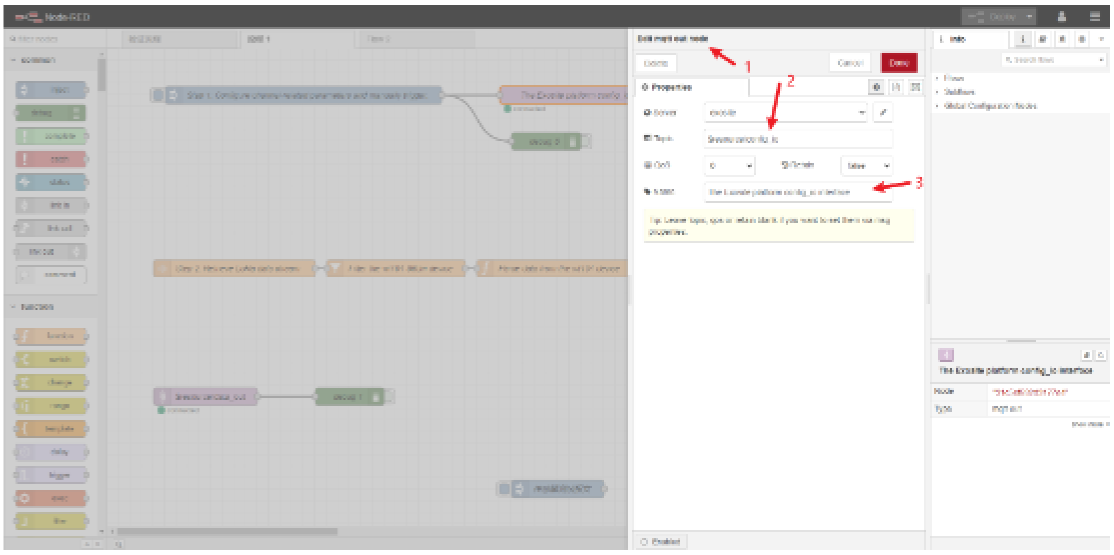
After completing the configuration as shown in the picture, click the "**Update**" button to save. You will see that the MQTT node has indicated "**connected**".



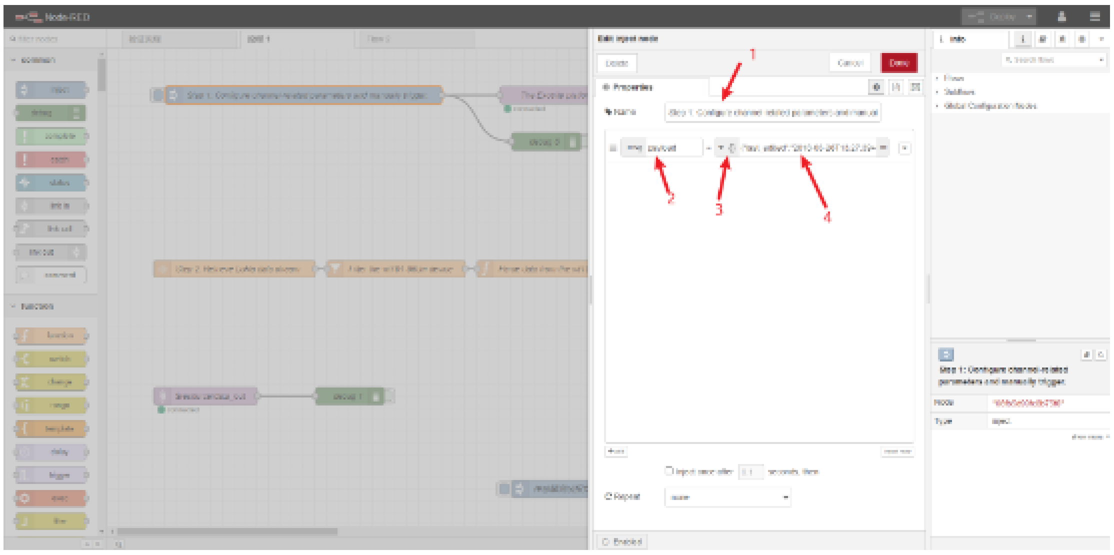
2.4. Create a config_io connector

Drag an MQTT Out node onto the canvas, then configure it as shown in the picture:

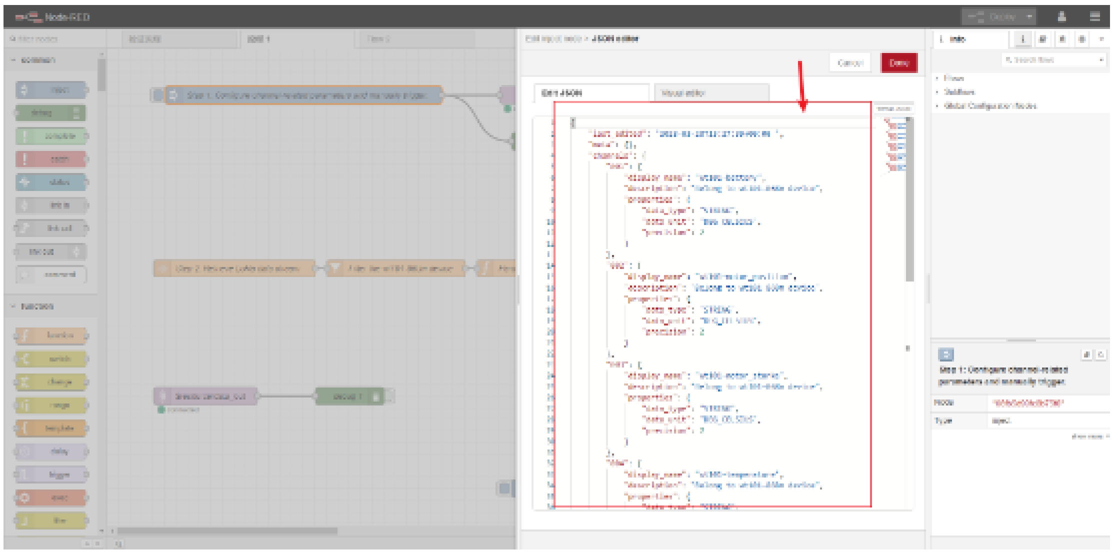
NOTICE: Ensure the topic path is configured correctly. Generally, it does not begin with characters like /.



Next, drag an Inject node onto the canvas, then configure the parameters as shown in the picture.



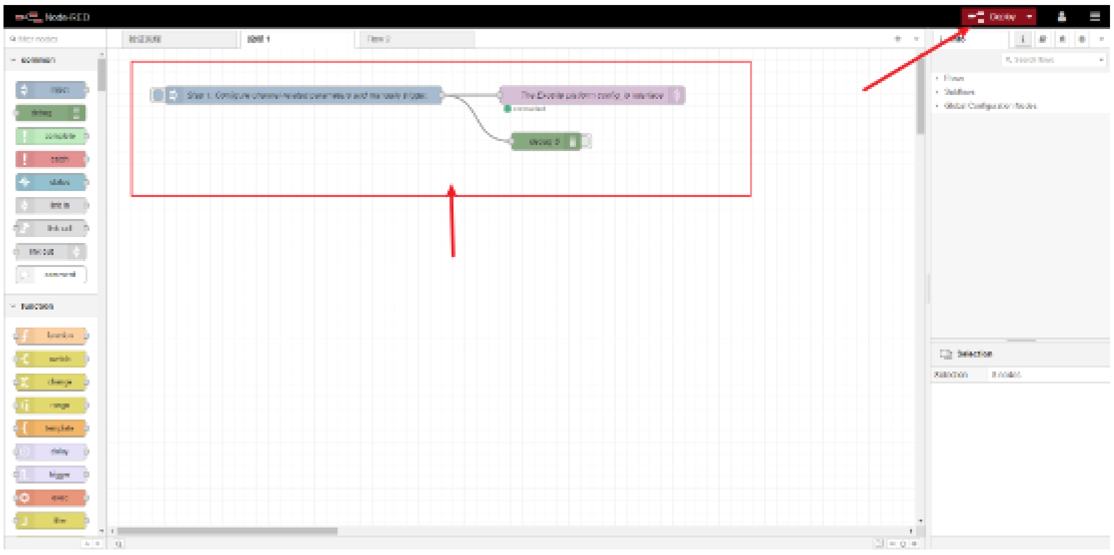
Here, configure it for the WT-101 device, as shown in the picture:



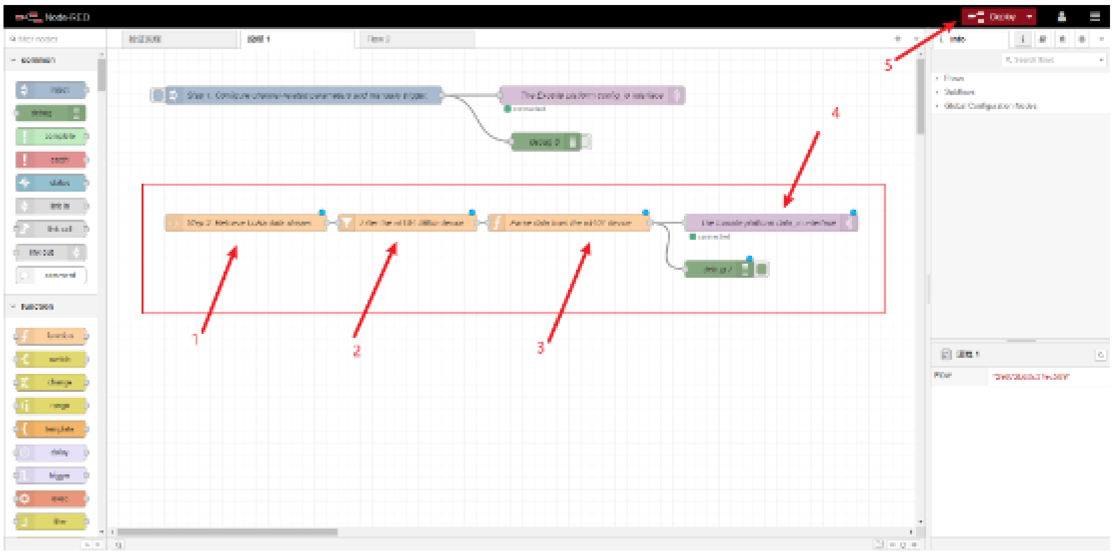
The complete JSON content of config_io is as follows:

```
{
    "last_edited": "2018-03-28T13:27:39+00:00 ",
    "meta": {},
    "channels": {
        "001": {
            "display_name": "wt101-battery",
            "description": "Belong to wt101-868m device",
            "properties": {
                "data_type": "STRING",
                "data_unit": "DEG_CELSIUS",
                "precision": 2
            }
        },
        "002": {
            "display_name": "wt101-motor_position",
            "description": "Belong to wt101-868m device",
            "properties": {
                "data_type": "STRING",
                "data_unit": "DEG_CELSIUS",
                "precision": 2
            }
        },
        "003": {
            "display_name": "wt101-motor_storke",
            "description": "Belong to wt101-868m device",
            "properties": {
                "data_type": "STRING",
                "data_unit": "DEG_CELSIUS",
                "precision": 2
```

Manually create a connection line to connect the two components, then click the **Deploy** button in the top right corner to save the scheme.

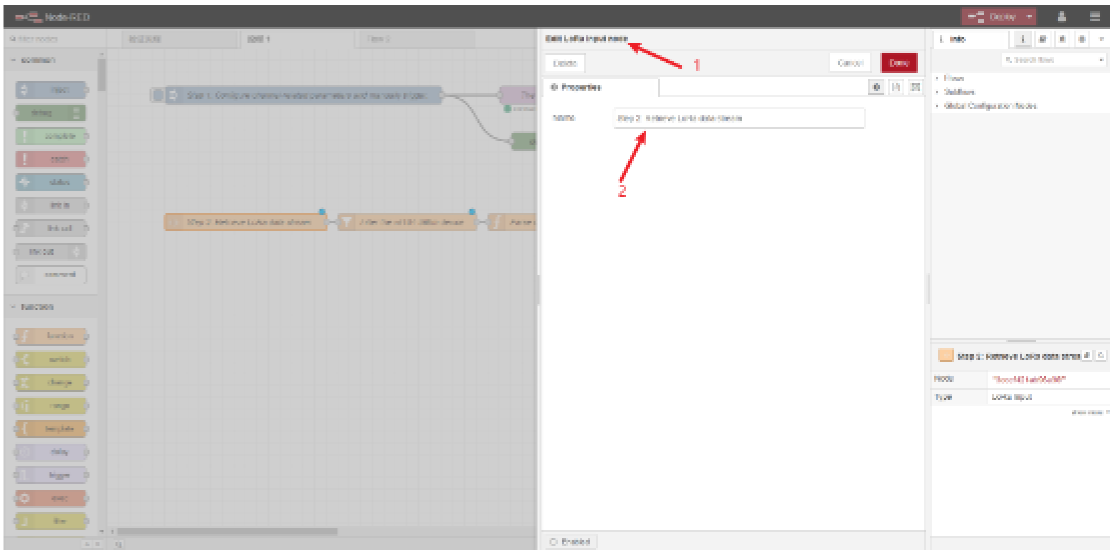


2.5. Create a data_in data reporting channel

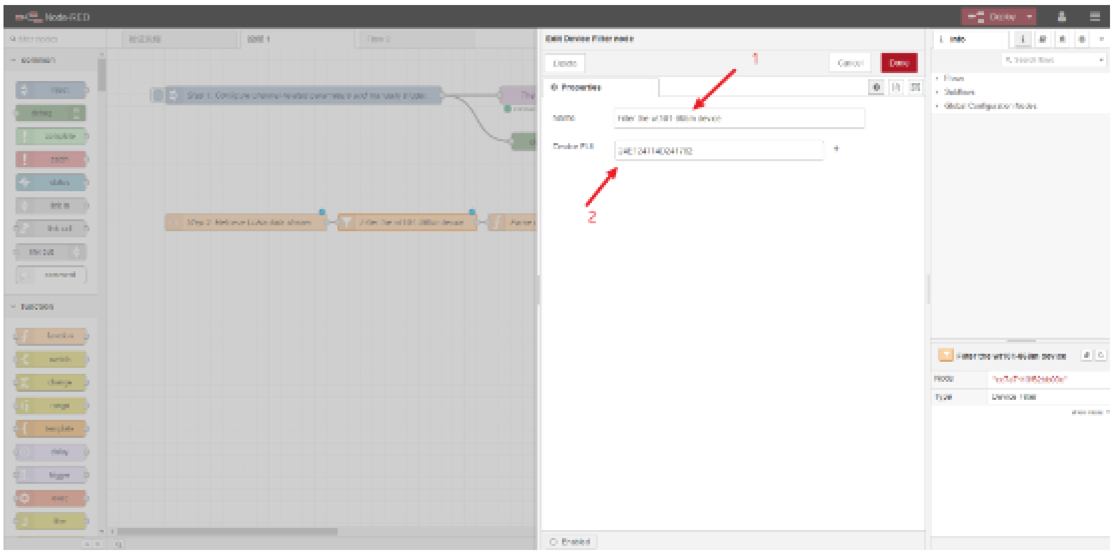


As shown in the picture, the basic configuration process is to first retrieve the device's data stream, then filter the required device data, parse the device's data packets, and finally use the data_in interface for data reporting. Next, let's start parsing step by step.

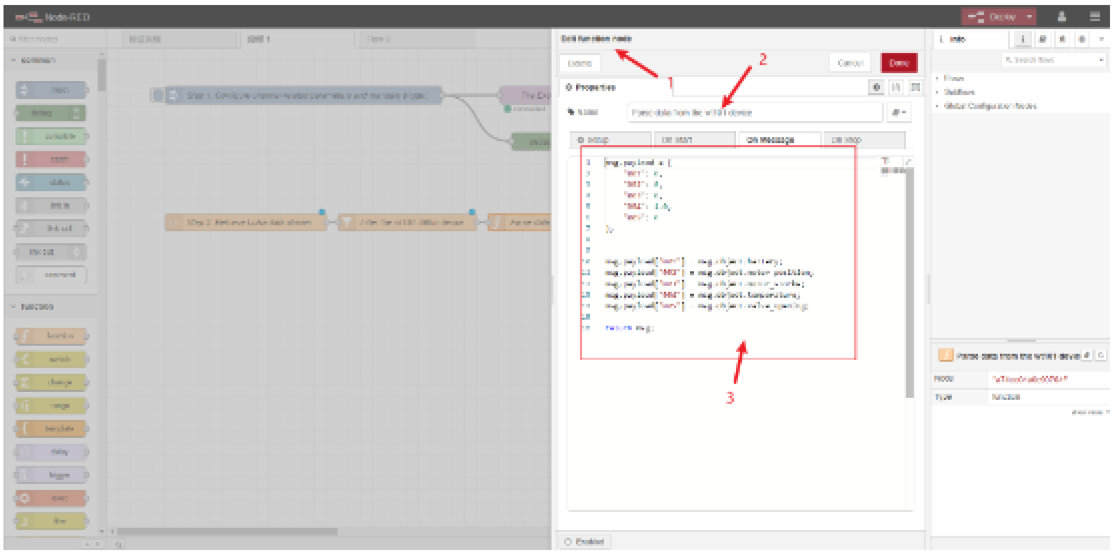
First, drag and drop an input node plugin onto the canvas, then configure it as shown in the picture



Then drag and drop another device filter node plugin, and configure it. For the **Device EUI**, please check the gateway interface yourself.



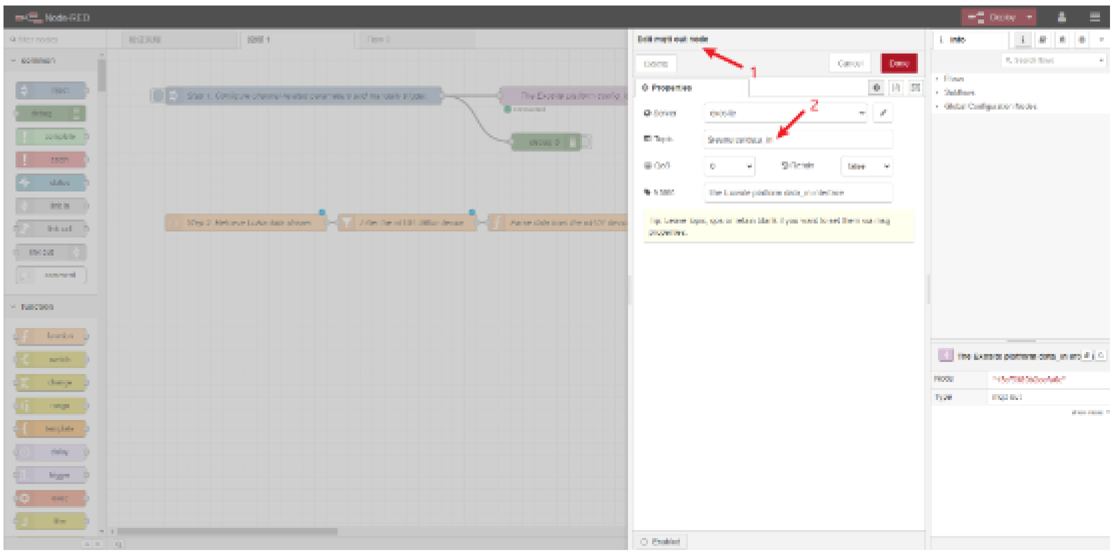
As shown in the picture, continue dragging and drop a function node for configuration, as shown in the picture:



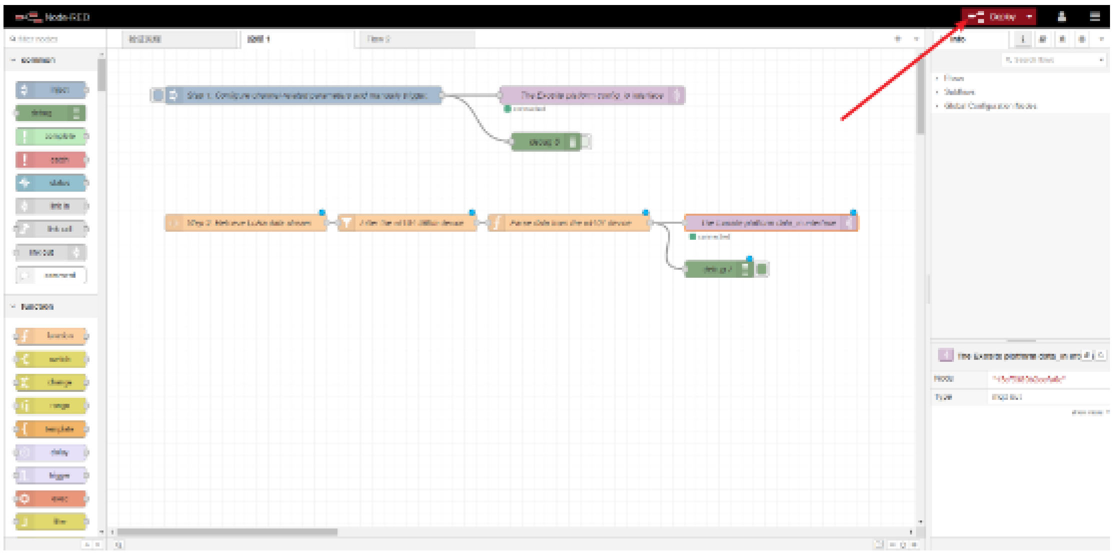
Note: The syntax used here is JavaScript syntax. The complete code snippet is as follows:

```
msg.payload = {
    "001": 0,
    "002": 0,
    "003": 0,
    "004": 1.0,
    "005": 0
};


msg.payload["001"] = msg.object.battery;
msg.payload["002"] = msg.object.motor_position;
msg.payload["003"] = msg.object.motor_storke;
msg.payload["004"] = msg.object.temperature;
msg.payload["005"] = msg.object.valve_opening;

return msg;
```

Finally, drag and drop an MQTT out node plugin, and configure the parameters as follows:

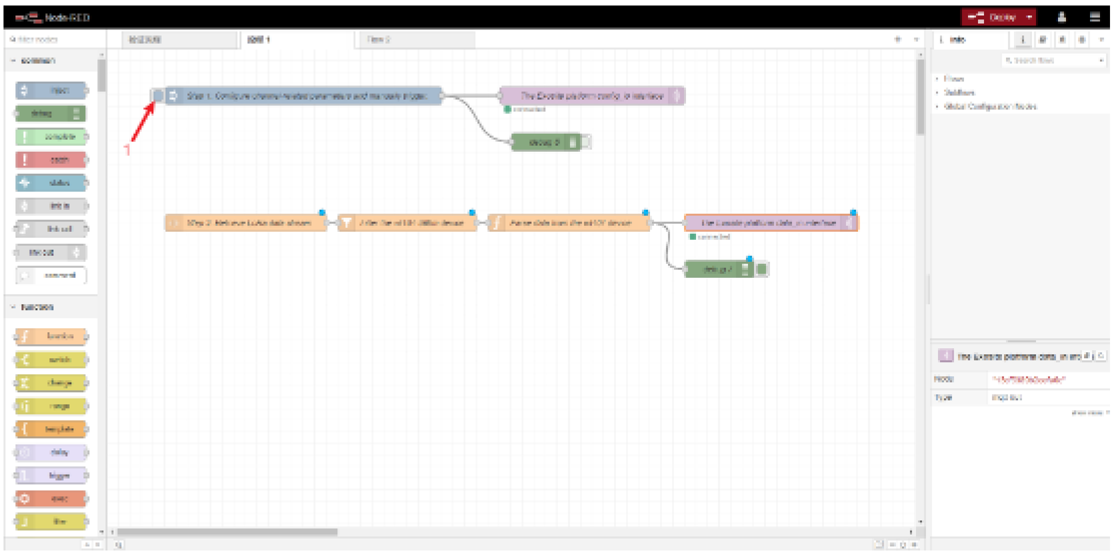


Click the **Deploy** button in the top right corner to save the scheme.

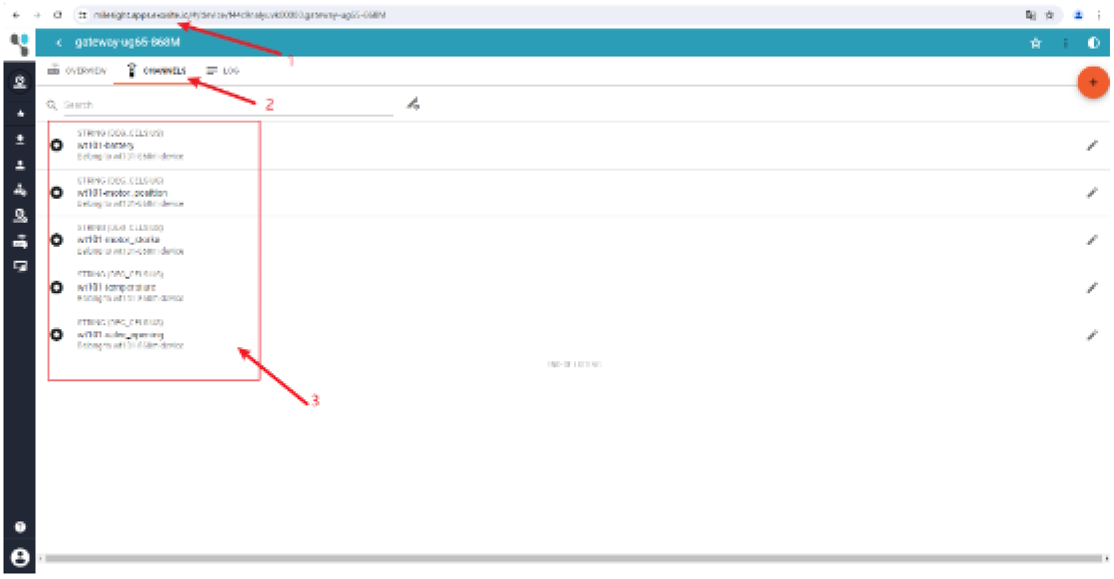


2.6. Initialize config_io data

Manually click on the location indicated by the arrow:

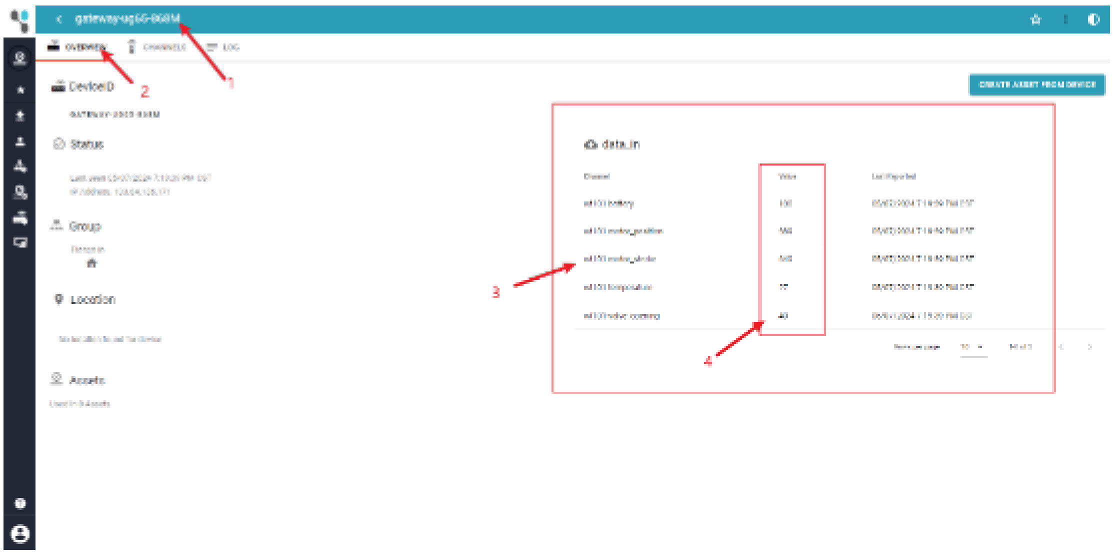


When we open the Exosite interface, we can see that the platform has received the messages and created the corresponding channels.



2.7. Submit data via the data_in interface

Wait a moment, and you will see the following interface. We can see that the data coming in through data_in has been correctly parsed according to config_io. With this, the platform integration is complete, and the data link is **ON**.



3. Appendix

For your convenience in configuration, this article includes an exported flow file for Node-RED. You can import it through the Node-RED interface, and then adjust the parameters inside according to the content of the article as needed.
File is here : Exosite Demo Flow.json

-END-