Milesight AIoT Inference Platform

User Guide

Readers

This guide is intended for the following users:

- Distributors
- Network Planners
- On-site technical support and maintenance personnel
- Network administrators responsible for network configuration and maintenance

For assistance, please contact

Milesight technical support:

Email: iot.support@milesight.com

Support Portal: support.milesight-iot.com

Tel: 86-592-5085280

Fax: 86-592-5023065

Address: Building C09, Software Park Phase III, Xiamen 361024, China

Revision History

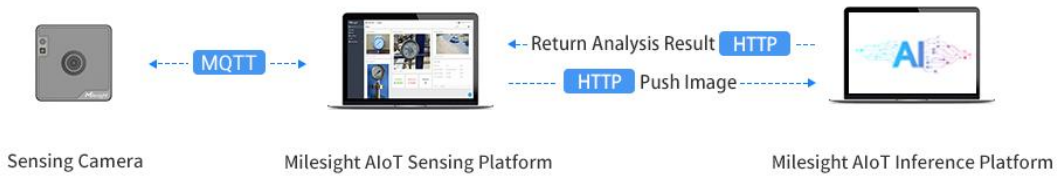
Date	Doc Version	Description
July 20, 2023	V 1.0	Initial version

Contents

1. Product Introduction

1.1 Overview

Milesight AIoT Inference Platform, based on Triton inference server, provides an intelligent AI solution to recognize the data from Milesight sensing cameras.



1.2 Recommended System

Hardware

For 1 to 300 devices

- RAM: 4 GB

For 300 to 500 devices

- RAM: 8 GB

Software

Operating System:

- Ubuntu Kinetic 22.10
- Ubuntu Jammy 22.04 (LTS)
- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)

2. Installation

2.1 Requirement

- Milesight AIoT Inference Platform Image Package
- WinSCP
- Putty (or other SSH tool)
- Install Docker: for Ubuntu

2.2 Compose Installation

1. Download Milesight AIoT Inference platform image package from Milesight website and import it to local path of system via WinSCP or other tools.
2. Navigate to the path of image package.
3. Push image to docker.

```
sudo -i
docker load < msiotinferenceplatform-min-1.0.tar
```

4. Create docker compose file:

```
nano docker-compose.yml
```

Add the following lines to the yml file:

```
version: '3.0'
services:
  msinfer:
    image: ms-inference-server-min:1.0
    shm_size: '1gb'
    ports:
      - "5221:8080"
    volumes:
      - /var/ms-inference-server:/ms-inference-server
    command: infer_daemon
```

Parameter introduction:

- 5221:8080 - connect local port 5221 to exposed internal HTTP port 8080, internal port 8080 is not allowed to be changed, or the platform may not work well
- /var/ms-inference-server:/ms-inference-server - mounts the host's dir /var/ms-inference-server to platform logs directory
- msinfer - friendly local name of this machine
- image: ms-inference-server-min:1.0 - image name

Note: if you already has created a compose file, please copy the contents "msinfer....infer_daemon" to the same compose file.

```
version: '3.0'
services:
  mysp:
    restart: always
    image: "msaiotsensingplatform:1.0.1.0-a4"
    ports:
      - "5220:9090"
      - "1883:1883"
      - "7070:7070"
      - "5683-5688:5683-5688/udp"
    environment:
      TB_QUEUE_TYPE: in-memory
      CASSANDRA_URL: localhost:9042
      CASSANDRA_KEYSPACE_NAME: msaiotsensingplatform
    volumes:
      - /var/mysp-data:/data
      - /var/mysp-logs:/var/log/msaiotsensingplatform

  msinfer:
    image: ms-inference-server-min:1.0
    shm_size: 'lgb'
    ports:
      - "5221:8080"
    volumes:
      - /var/ms-inference-server:/ms-inference-server
    command: infer_daemon
```

5. Set the terminal in the directory which contains the docker-compose.yml file and execute the following commands to up this docker compose directly:

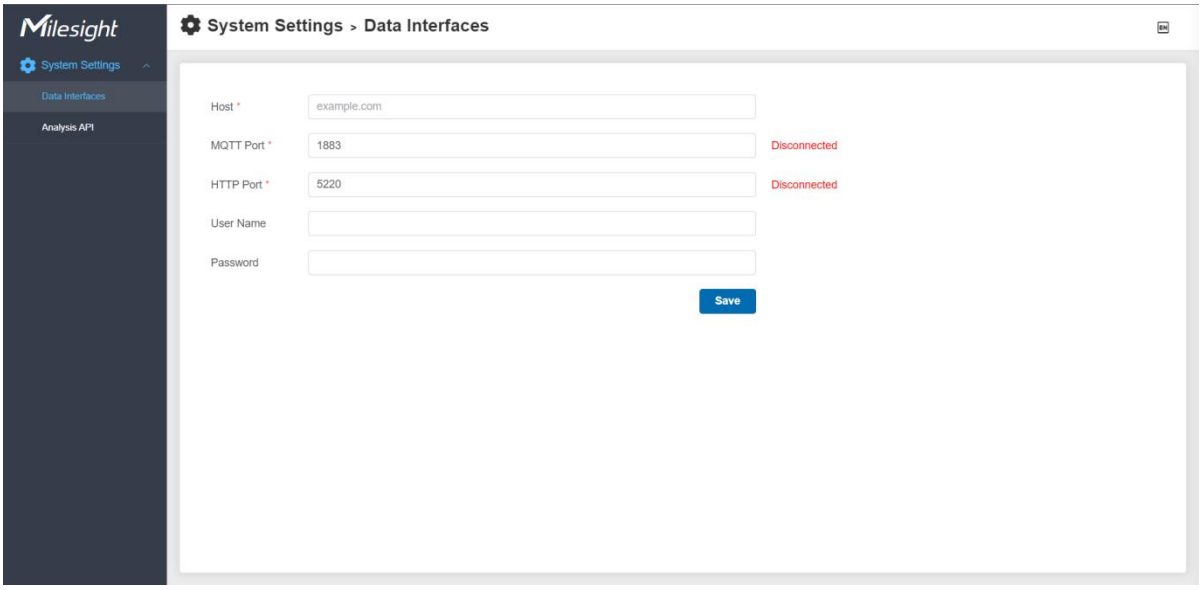
```
docker compose up -d
```

It will take about 1 minute to complete the installation and start the program.

Note: Docker Compose as docker-compose (with a hyphen) is deprecated. It is recommended to use Docker Compose V2 instead. If you still rely on docker compose as standalone here is the list of the above commands:

docker-compose up -d

6. After installation, type http://{your-host-ip}:5221 in your browser to visit the page.



7. In case of any issue you can examine service logs for errors. For example, you can check platform logs by executing the following command:

```
docker compose logs -f msinfer
```

To stop the Milesight AIoT Inference platform:

```
docker compose stop msinfer
```

To start the Milesight AIoT Inference platform:

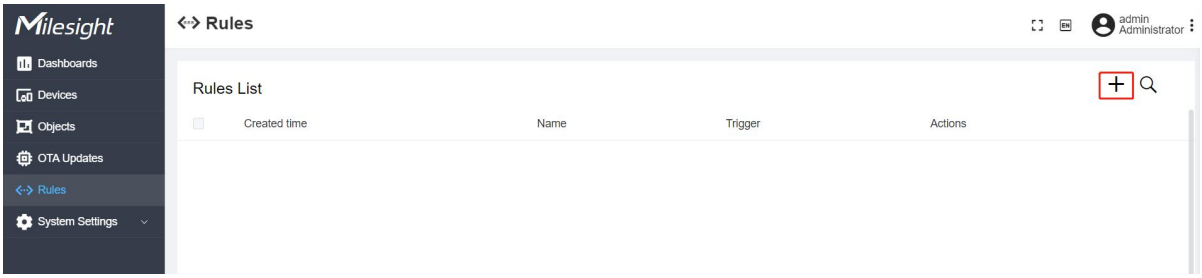
```
docker compose start msinfer
```

Note: Docker Compose as docker-compose (with a hyphen) is deprecated. It is recommended to use Docker Compose V2 instead. If you still rely on docker compose as standalone here is the list of the above commands:
docker-compose logs -f msinfer
docker-compose stop msinfer
docker-compose start msinfer

3. Operation Guide

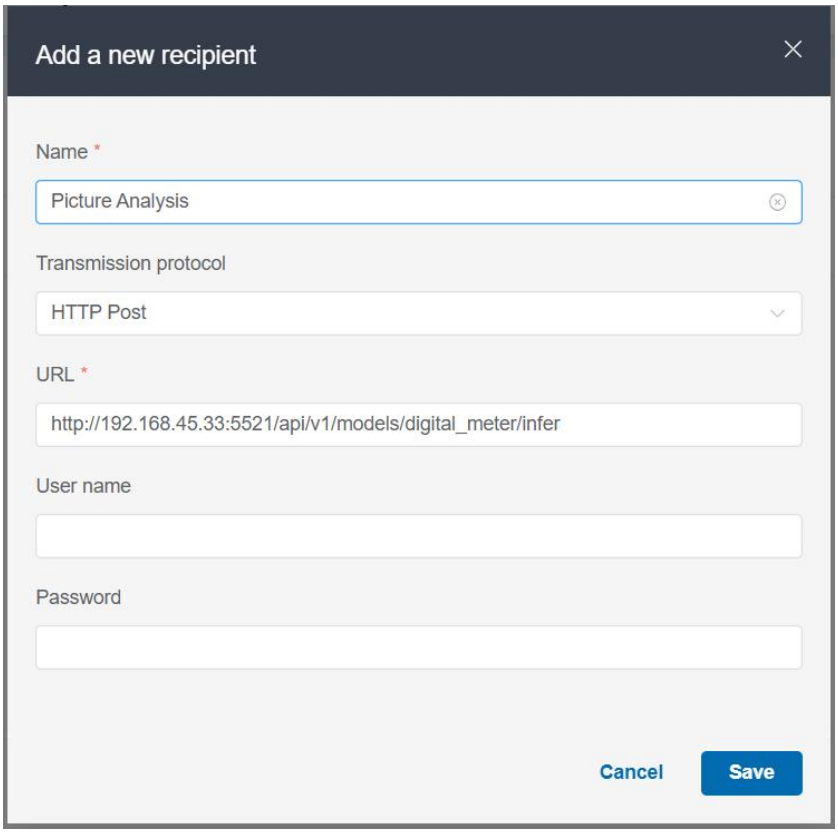
AIoT Sensing Platform Configuration

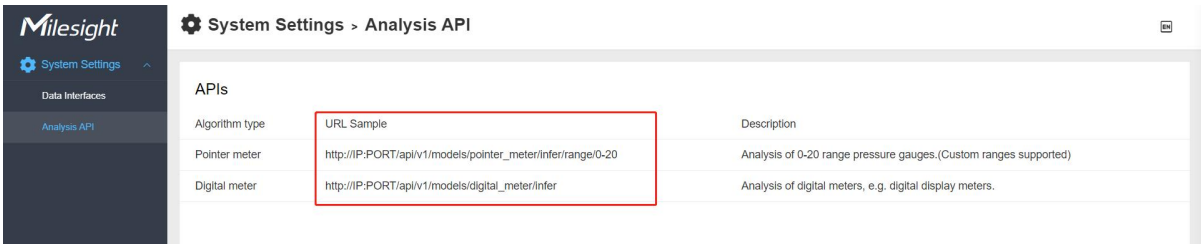
Step 1: Ensure that the device has been added and objects has been created on Milesight AIoT Sensing Platform. For more details please refer to ***AIoT Sensing Platform User Guide***.
Step 2: Go to **System Settings > Recipients** page, click "+" to add a new recipient.



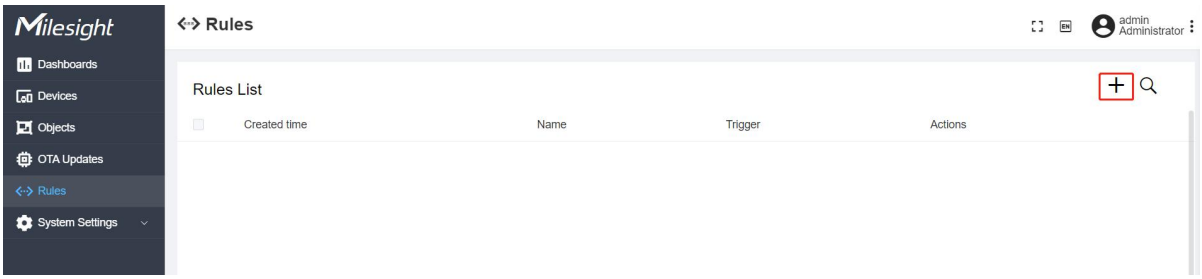
Set the recipient information as below:

- Name: user-defined
- Transmission protocol: HTTP Post
- URL: API URL of Milesight AIoT Inference platform depending on meter type, the "IP:PORT" should be replaced as real IP and port of Milesight AIoT Inference platform.
- Username/Password: leave blank



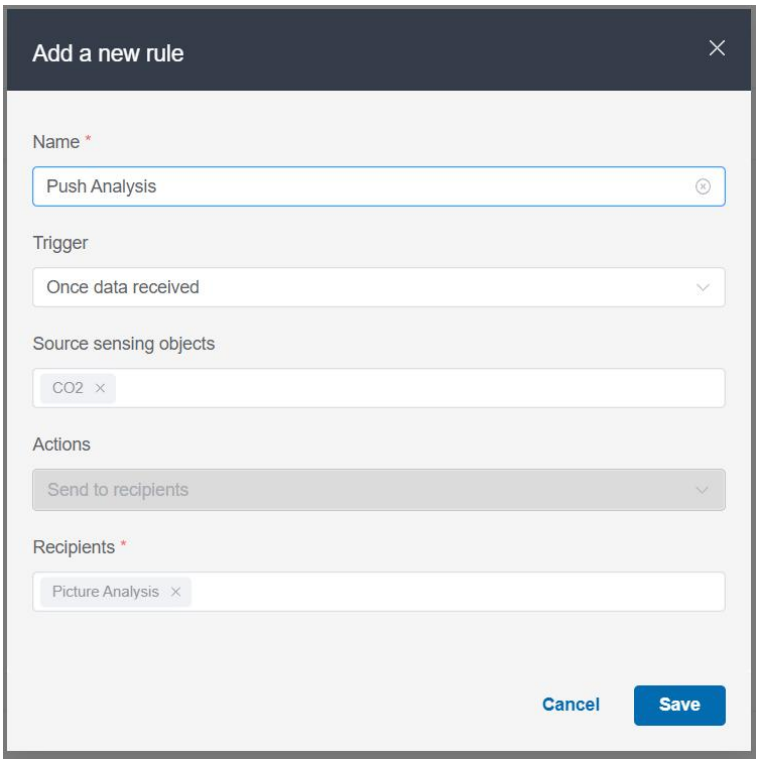


Step 3: Go to **Rules** page, click "+" to add a new rule.



Set the rule information as below:

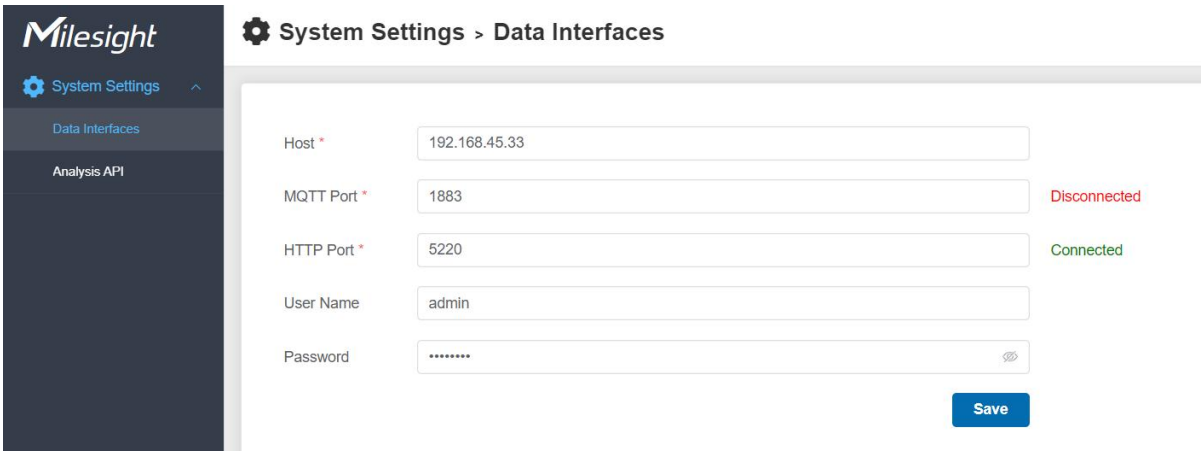
- Name: user-defined
- Trigger: Once data received
- Source sensing objects: the data needs to be recognized
- Recipients: Milesight AIoT Inference platform



AIoT Inference Platform Configuration

Add the server information to return the recognized results:

- Host: IP address or domain of Milesight AIoT Sensing platform
- MQTT Port: keep this parameter by default
- HTTP Port: HTTP port of Milesight AIoT Sensing platform, it is 5220 by default
- User Name: username of Milesight AIoT Sensing platform
- Password: password of Milesight AIoT Sensing platform



Note: if both platforms are installed to the same server and use a same public IP address, please find the correct platform IP addresses and ports via below steps:

1. Type “docker ps” to search for docker ID of both platforms.
2. Search for docker IP addresses of both platforms via below command, “<container ID>”

should be replaced as a real container ID.

```
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' <Container ID>
```

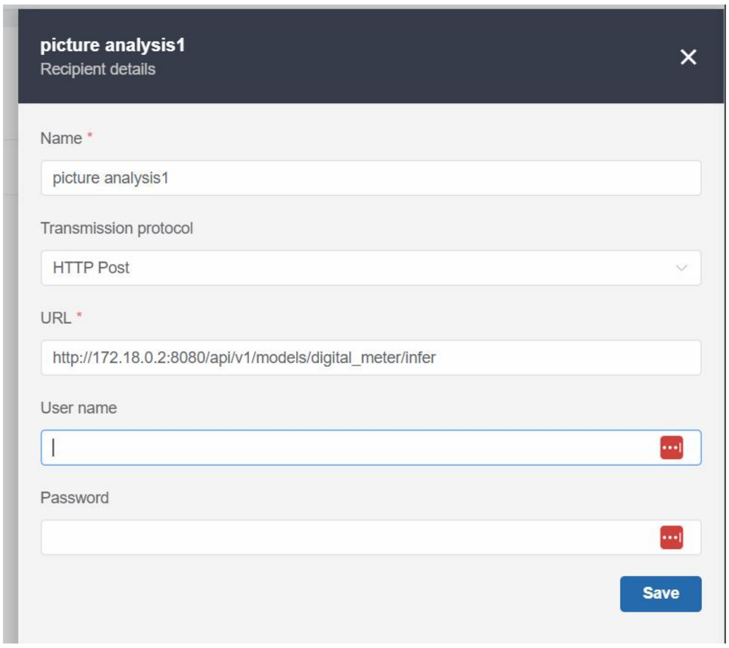
Take below example, AIoT Sensing platform docker IP address is 172.18.0.3, AIoT Inference platform docker IP address is 172.18.0.2.

```
root@aiotsense:~# docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED       STATUS       PORTS
                                                        NAMES
16d393156098   ms-inference-server-min:1.0      "/opt/nvidia/nvidia_…"   2 weeks ago   Up 2 weeks   0.0.0.0:5221->8080/tcp, :::5221->8080/tcp
                                                        root-msinfer-1
3f7df7ac93e4   msaiotsensingplatform:1.0.1.1    "start-tb.sh"            2 weeks ago   Up 2 weeks   0.0.0.0:1883->1883/tcp, :::1883->1883/tcp, 5432/tcp, 0.0.0.0:7070->7070/tcp, :::7070->7070/
5683-5688/udp, 0.0.0.0:5220->9090/tcp, :::5220->9090/tcp   root-mysp-1
root@aiotsense:~#
root@aiotsense:~# docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 16d393156098
172.18.0.2
root@aiotsense:~# docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 3f7df7ac93e4
172.18.0.3
```

3. Check internal HTTP ports of both platforms from docker compose file:

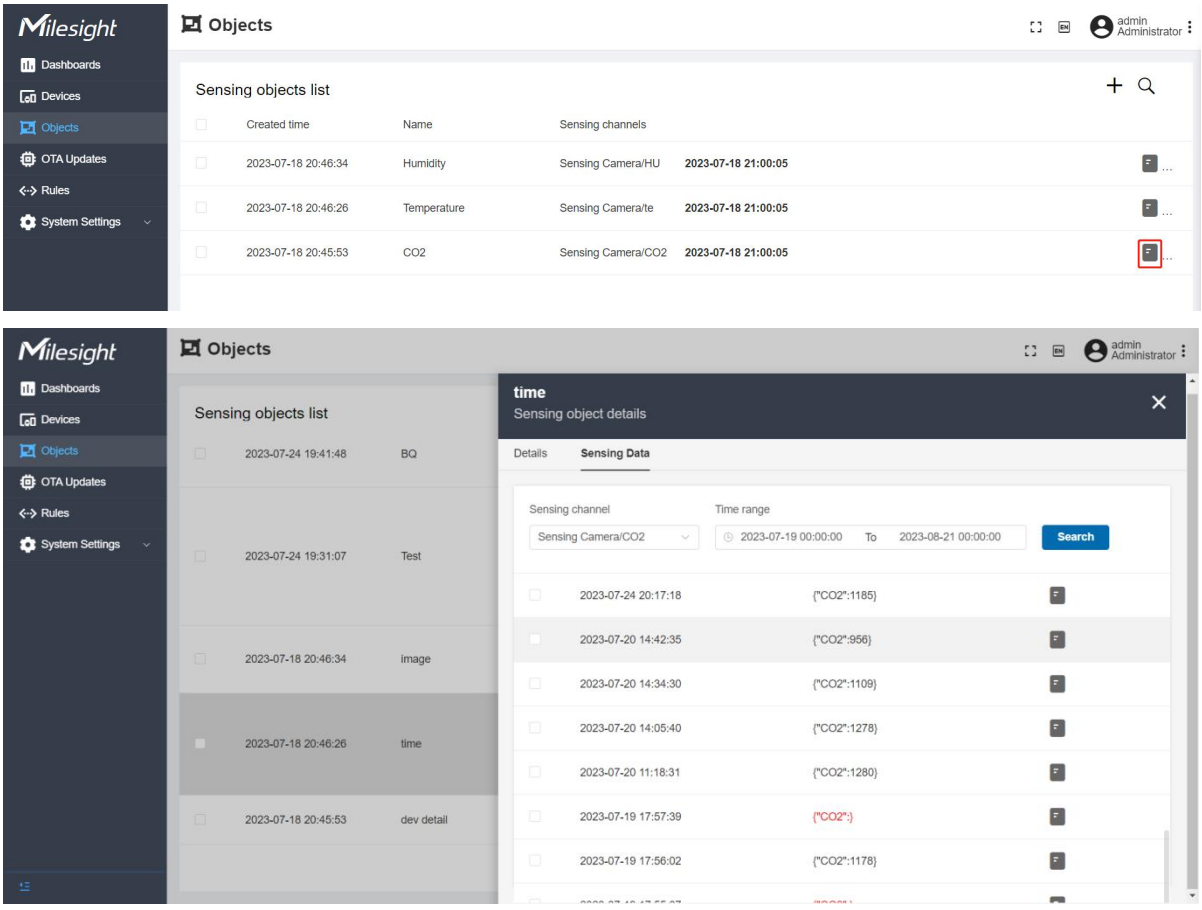
```
version: '3.0'
services:
  mysp:
    restart: always
    image: "msaiotsensingplatform:1.0.1.1-r1"
    ports:
      - "5220:9090"
      - "1883:1883"
      - "7070:7070"
      - "5683-5688:5683-5688/udp"
    environment:
      TB_QUEUE_TYPE: in-memory
      CASSANDRA_URL: localhost:9042
      CASSANDRA_KEYSPACE_NAME: msaiotsensingplatform
    volumes:
      - /var/mysp-data:/data
      - /var/mysp-logs:/var/log/msaiotsensingplatform

  msinfer:
    image: ms-inference-server-min:1.0
    shm_size: '1gb'
    ports:
      - "5221:8080"
      - "9001:9001"
    volumes:
      - /var/ms-inference-server:/ms-inference-server
    command: infer_daemon
```

4. Use docker IP address + internal HTTP port when setting up connections of both platforms.

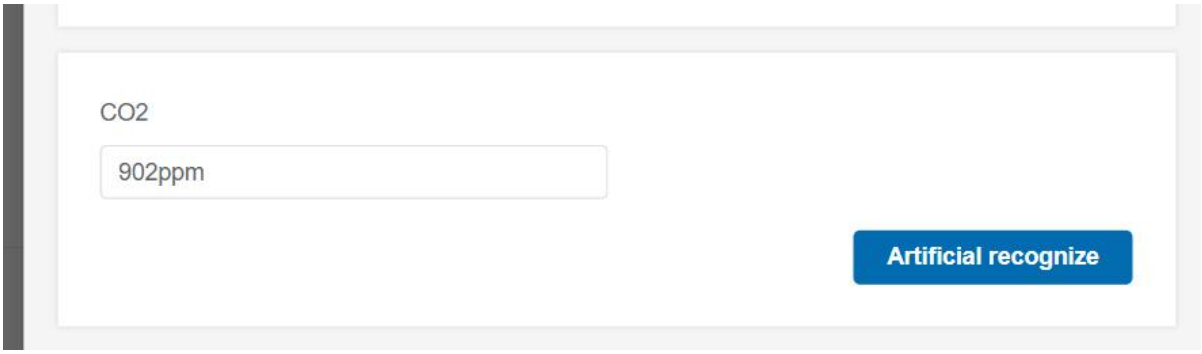


Check Recognition Results

Step 3: Go to **Objects** page, click the button on the right of object item to check the results.



If the value is unrecognized or error, click the button beside the value to manually type the data and click **Artificial recognize**.



-END-