RX65N Group

Sample Code for OTA Update of Secondary Device with Amazon Web Services Using FreeRTOS

Introduction

This application note uses an RX65N as a gateway device that communicates with Amazon Web Services™ (AWS) and an RX microcomputer (hereinafter, 2nd RX MCU) as a secondary device for sensor data measurement.

This application note describes a demo that uses AWS services to perform an OTA update of the 2nd RX MCU (hereinafter, 2nd OTA update).

In the demo, you can add a working sensor with the 2nd OTA update and see the working sensor added on the AWS screen on your browser.

IoT devices are required to fix security vulnerabilities as appropriate and update their functions according to customer requests. By implementing the 2nd OTA update in addition to the OTA of the gateway device that has been provided in the past, it is possible to realize product development that can respond to vulnerabilities in secondary devices and update flexible services.

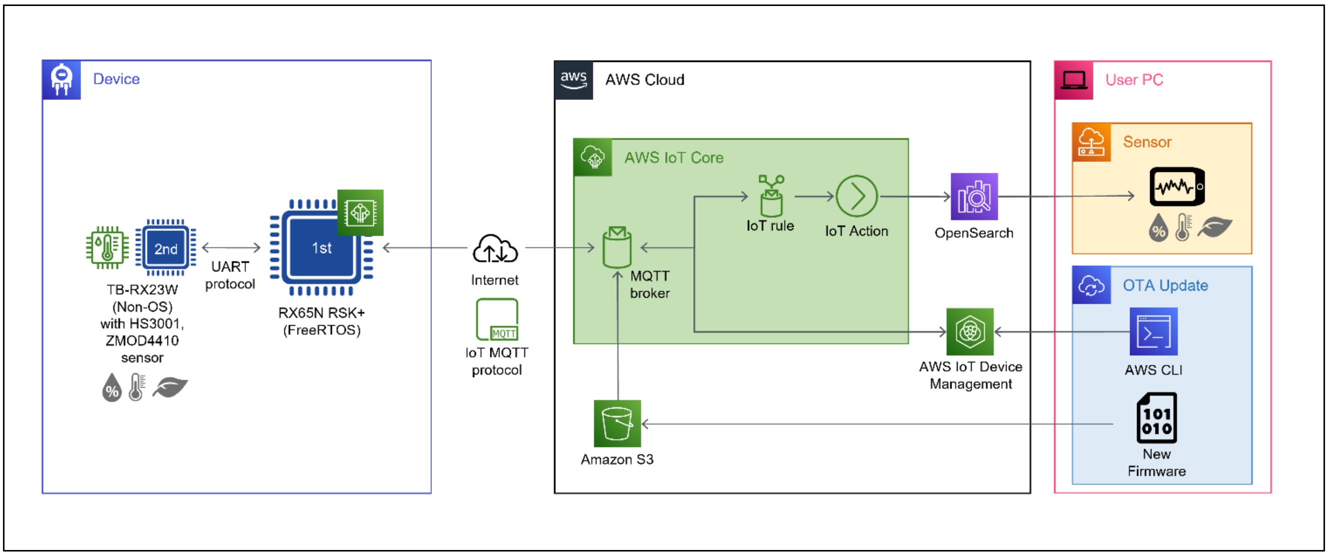

Figure System Configuration of Demo

Operation Confirmation Devices

- RX65N Group, RX651 Group
- RX23W Group
- Sensors
  Renesas Electronics HS3001 Relative Humidity and Temperature Sensor (HS3001 sensor)
  Renesas Electronics ZMOD4410 Indoor Air Quality Sensor (ZMOD4410 sensor)

Operation Confirmation Boards

- Renesas Starter Kit+ for RX65N-2MB (RSK+RX65N-2MB board)
- Target Board for RX23W (TB-RX23W board)
- HS3001 Sensor Board*1: Relative Humidity Sensor Pmod™ Board (US082-HS3001EVZ)
- ZMOD4410 Sensor Board*1: ZMOD4410 Sensor Pmod Board (US082-ZMOD4410EVZ)
- Converter board*1: Interposer board to convert Type2/3 to 6A (US082-INTERPEVZ)

Note: 1. Contact a Renesas representative for information on obtaining the Interposer Board.

Related Documents

This application note refers to and explains the following documents. The chapter structure may change when the document is updated. Please be careful when referencing.

Document Title	Document No.
RX Family Renesas Sensor Control Modules Firmware Integration Technology	R01AN5892EJ0110
RX Family Renesas HS300x Sensor Control Module Firmware Integration Technology	R01AN5893EJ0110
RX Family Renesas Sensor I2C Communication Middleware Control Module Firmware Integration Technology	R01AN5895EJ0110
Quick Connect IoT Manual	R36UZ0008EU0100
RX Family How to implement FreeRTOS OTA by using Amazon Web Services on RX65N	R01AN5549EJ0102
RX65N Group Visualizing and Controlling Sensor Information Using Amazon Web Services with RX65N Cloud Kit and FreeRTOS	R20AN0623EJ0100
RX Family Firmware Update Module Using Firmware Integration Technology	R01AN5824EJ0102

Contents

1. Overview

1.1 System Overview

The system comprises an RX65N cloud connectivity solution that provides functionality for controlling communication with AWS and an 2nd RX MCU connected to the HS3001 sensor and ZMOD4410 sensor. The two MCUs intercommunicate via UART.

The RX65N firmware, which utilizes AWS-certified Program, includes an implementation of FreeRTOS™ with IoT Library, and it makes use of AWS IoT Core and AWS IoT Device Management, a managed service provided by Amazon AWS, to perform the OTA firmware updating and upload data to the cloud via MQTT communication.

The demo also can perform the OTA firmware updating of the 2nd RX MCU connected to the sensors and uploading to the cloud and visualization of sensor data obtained from the sensor boards via UART communication

1.1.1 System Configuration

The AWS IoT Over-the-air Update Library is used to control the RX65N. The previously released application note How to implement FreeRTOS OTA by using Amazon Web Services on RX65N describes the procedure for OTA updating of the RX65N's own firmware using this library. The 2nd OTA update uses the library, and the firmware received by the RX65N is then transmitted to the 2nd RX MCU, where the firmware update is applied.

Note that the Firmware Update Module Using Firmware Integration Technology is used to control firmware updating on the 2nd RX MCU.

For a reference implementation of uploading sensor data obtained from the sensor boards to the cloud and data visualization using the AWS OpenSearch service, refer to Visualizing and Controlling Sensor Information Using Amazon Web Services with RX65N Cloud Kit and FreeRTOS.

Figure **System Configuration of Demo** on the cover shows the system configuration of the demo.

The RX23W is used as the RX MCU, and it is mounted on the TB-RX23W board.

The RSK+RX65N-2MB board is used as the RX65N cloud connectivity solution.

In addition, the RSK+RX65N-2MB board and TB-RX23W board are connected via UART.

1.2 Package Structure

1.2.1 Folder and File Structure

(1) General

Figure 1-1 shows the folder and file structure. Up to 6 folders in the FIT Demos folder are listed.

```
r01an6220xx0110-rx-2nd-ota-apl
├─FITDemos
│   ├─rx23w_tb_2ndota_demo
│   ├─rx23w_tb_bootloader
│   ├─aws_demos
│   └─rx65n_rsk_bootloader
├─r01an6220ej0110-rx-2nd-ota-apl.pdf
└─r01an6220jj0110-rx-2nd-ota-apl.pdf
```

Figure 1-1 Folder and File Structure

(2) Projects for RSK+RX65N-2MB Board (Boot Loader Project and User Program Project)

A pair of projects, a boot loader and a user program, are provided for the RSK+RX65N-2MB board.

(a) Boot Loader Project for RSK+RX65N-2MB Board

Figure 1-2 shows the folder and file structure. The description of some folders and files is omitted.

```
rx65n_rsk_bootloader
├─.cproject
├─.project
├─rx65n_rsk_bootloader.scfg
├─.settings
└─src
    ├─key
    │   └─code_signer_public_key.h
    ├─rx65n_rsk_bootloader.c
    ├─smc_gen
    └─src
        ├─base64_decode.c
        ├─base64_decode.h
        ├─r_simple_filesystem_on_dataflash.c
        ├─r_simple_filesystem_on_dataflash_if.h
        └─tinycrypt
            └─LICENSE
```

Figure 1-2 Folder and File Structure of Boot Loader Project for RSK+RX65N-2MB Board

The following open source is used. For licensing information, refer to 1.2.2, Open Source License Information to Use.

- TinyCrypt: It is stored in the \src\src\tinycrypt folder.

(b) User Program Project for RSK+RX65N-2MB Board

Figure 1-3 shows the folder and file structure. The description of some folders and files is omitted.

```
aws_demos
├─── demos
│    ├─── include
│    │    ├─── aws_clientcredential.h
│    │    ├─── aws_clientcredential_keys.h
│    │    ├─── aws_demo.h
│    │    ├─── aws_iot_metrics.h
│    │    ├─── iot_config_common.h
│    │    ├─── iot_demo_logging.h
│    │    └─── iot_demo_runner.h
│   ...
├─── freertos_kernel
├─── libraries
├─── projects
├─── tests
├─── tools
│    ├─── aws_config_quick_start
│    └─── certificate_configuration
│         ├─── CertificateConfigurator.html
│         ├─── PEMfileToCString.html
│         └─── js
├─── vendors
│    └─── renesas
│         ├─── amazon_freertos_common
│         ├─── boards
│         ├─── manifest.cmake
│         └─── rx_driver_package
│              └─── rx65n-rsk
│                   ├─── CMakeLists.txt
│                   ├─── ports
│                   └─── aws_demos
│                        ├─── application_code
│                        ├─── flash_project
│                        ├─── src
│                        └─── config_files
│                             ├─── ota_demo_config.h
│                            ...
...
```

Figure 1-3 Folder and File Structure of User Program Project for RSK+RX65N-2MB Board

The following open source is used. For licensing information, refer to 1.2.2, Open Source License Information to Use.

- FreeRTOS

(3) Projects for TB-RX23W Board (Boot Loader Project and User Program Project)

A pair of projects, a boot loader and a user program, are provided for the TB-RX23W board.

(a) Boot Loader Project for TB-RX23W Board

Figure 1-4 shows the folder and file structure. The description of some folders and files is omitted.

```
rx23w_tb_bootloader
├──.cproject
├──.project
├──rx23w_tb_bootloader.scfg
├──.settings
└──src
    ├──key
    │   └──code_signer_public_key.h
    ├──rx23w_tb_bootloader.c
    ├──smc_gen
    └──src
        ├──base64_decode.c
        ├──base64_decode.h
        └──tinycrypt
            └──LICENSE
```

Figure 1-4 Folder and File Structure of Boot Loader Project for TB-RX23W Board

The following open source is used. For licensing information, refer to 1.2.2, Open Source License Information to Use.

- TinyCrypt: It is stored in the \src\src\tinycrypt folder.

(b) User Program Project for TB-RX23W Board

Figure 1-5 shows the folder and file structure. The description of some folders and files is omitted.

```
rx23w_tb_2ndota_demo
├──.cproject
├──.project
├──rx23w_tb_2ndota_demo.scfg
├──.settings
└──src
    ├──base64_decode.c
    ├──base64_decode.h
    ├──common_if.h
    ├──common_task.c
    ├──common_task_if.h
    ├──fwup_2nd
    ├──hs300x.c
    ├──hs300x.h
    ├──key
    │   └──code_signer_public_key.h
    ├──rx23w_tb_2ndota_demo.c
    ├──smc_gen
    ├──tinycrypt
    │   └──LICENSE
    ├──zmod4410.c
    └──zmod4xxx.h
```

Figure 1-5 Folder and File Structure of User Program Project for TB-RX23W Board

The following open source is used. For licensing information, refer to 1.2.2, Open Source License Information to Use.

- TinyCrypt: It is stored in the \src\tinycrypt folder.

1.2.2 Open Source License Information to Use

The open source license information to use is shown below.

(1) FreeRTOS

URL https://www.freertos.org/
License FreeRTOS open source licensing, FreeRTOS license description, FreeRTOS license terms and OpenRTOS commercial licensing options.

(2) TinyCrypt

URL https://01.org/tinycrypt
License https://github.com/intel/tinycrypt/blob/master/LICENSE
It is used in Firmware Update Module Using Firmware Integration Technology

1.3 Operation Confirmation Conditions

The operation of the firmware has been confirmed under the following conditions.

Table 1-1 Demo Operation Confirmation Conditions (RX65N)

Item	Description
MCU	RX65N
MCU board	Renesas Starter Kit+ for RX65N-2MB (RSK+RX65N-2MB board)
Pmod LCD	Color LCD board (bundled with RSK+RX65N-2MB board)
IDE	e^2 studio 2022-01
Toolchain	CC-RX V3.04.00
RTOS	FreeRTOS AWS Reference Integrations Version 202107.00
Firmware concatenation tool	Renesas Secure Flash Programmer V.1.01
Firmware programming tool	Renesas Flash Programmer V3.09.00

Table 1-2 Demo Operation Confirmation Conditions (RX23W)

Item	Description
MCU	RX23W
MCU board	Target Board for RX23W (TB-RX23W board)
IDE	e^2 studio 2022-01
Toolchain	CC-RX V3.04.00
Firmware concatenation tool	Renesas Secure Flash Programmer V.1.01
Firmware programming tool	Renesas Flash Programmer V3.09.00

Table 1-3 Demo Operation Confirmation Conditions (Sensor Board)

Item	Description
Sensor 1	HS3001 sensor
Sensor 1 board	Relative Humidity Sensor Pmod Board (US082-HS3001EVZ)
Sensor 2	ZMOD4410 sensor
Sensor 2 board	ZMOD4410 Sensor Pmod Board (US082-ZMOD4410EVZ)
Interface converter board	Interposer board to convert Type2/3 to Type 6A (US082-INTERPEVZ) Converter board for connecting the PMOD connector on the TB-RX23W board to the sensor board

1.4 Code Size

Figure 1-4 shows the code size of each project.

Table 1-4 Code Size Lists

Board	Project	ROM*1	RAM
RSK+RX65N-2MB board	Boot Loader	40KB*2	139KB
	User Program	457KB*2	383KB
RSK+RX65N-2MB board	Boot Loader	39KB	18KB
	User Program	89KB	34KB

Note: 1. Value excluding data flash memory.

Note: 2. Code flash memory is used in dual mode.

1.5 About AWS Regions and User Privileges for Demo

The region used and User Permissions are shown below when setting up the AWS for the demo.

<Region used >

This demo run in the AWS ap-northeast-1 (Asia Pacific (Tokyo) region).

If you want to run this demo in another region, please confirm that the service used in the demo is available in that region.

<Region used >

This demo runs by a user with Administrator Access permission in AWS Identity and Access Management (IAM). Therefore, there is no particular description regarding the granting of necessary permissions in IAM when using various services.

2. Firmware Programming

You can use the Renesas Flash Programmer to program a MOT file format to the RX65N on the RSK+RX65N-2MB board and to the RX23W on the TB-RX23W board.

Obtain the Renesas Flash Programmer V3.09.00 or later from the link below.

https://www.renesas.com/software-tool/renesas-flash-programmer-programming-gui

Follow the steps below using the TB-RX23W board.

(1) Step 1: ESW1 switch setting when programming TB-RX23W board

Make the following setting.

- Firmware programming and debugging: Set ESW1 2-4 to ON.

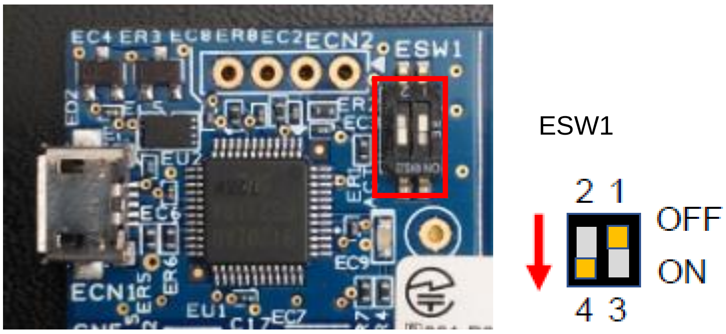

Figure 2-1 ESW1 Switch Setting when Programming TB-RX23W Board

(2) Step 2: Connecting the PC to the TB-RX23W board

Connect the PC to the emulator connector (ECN1) with a USB cable as shown in Figure 2-2.

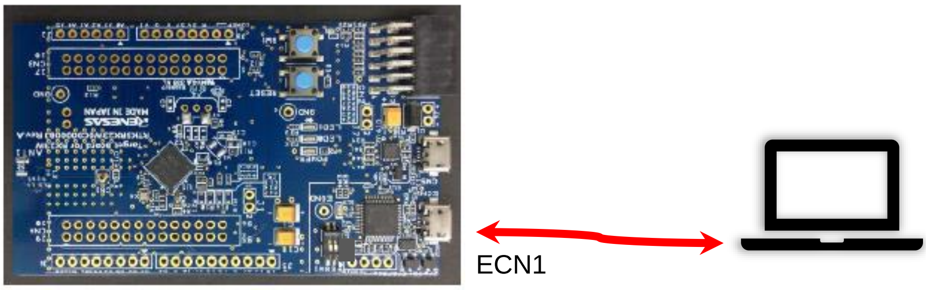

Figure 2-2 Connecting the PC to the TB-RX23W Board

(3) Step 3: Renesas Flash Programmer startup and connection processing

Launch the Renesas Flash Programmer.

The procedure differs depending on whether or not a Renesas Flash Programmer project has already been created. The following two versions of the procedure are described here.

a. Procedure when Renesas Flash Programmer project has not yet been created
b. Procedure when Renesas Flash Programmer project has already been created

(a) Procedure when Renesas Flash Programmer project has not yet been created

1. Creating a New Project

From the **File** menu, select **Create New Project…**.
Make the following settings.

- **Project Information** → **Microcontroller:** RX200
- **Communication** → **Tool: E2 emulator Lite** (Specifies the onboard emulator on the TB-RX23W board.)
- **Communication** → **Interface**: **FINE**

Confirm that Power: None is displayed under Communication.

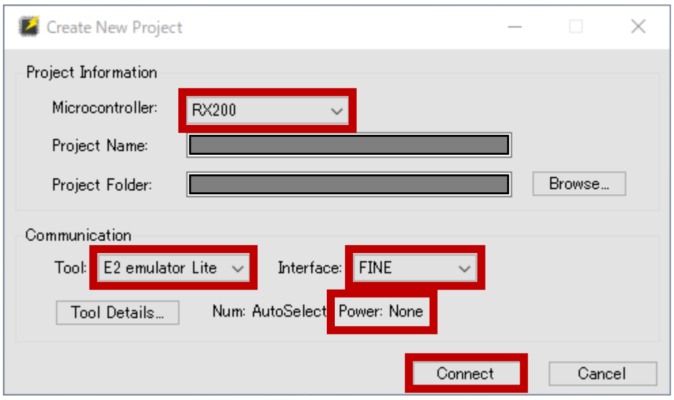

Figure 2-3 Project Information and Communication Settings

2. Connection Processing

Click the **Connect** button shown in Figure 2-3. The Renesas Flash Programmer starts connection processing.
Confirm that the following is displayed in the log output window.

```
Disconnecting the tool
Operation completed.
```

Figure 2-4 Display in Log Output Window upon Successful Connection

(b) Procedure when Renesas Flash Programmer project has already been created

1. Opening the Project

From the **File** menu, select **Open Project…** and select the project file.
Select the **Connect Settings** tab and confirm the following settings under **Communication**.

- **Tool: E2 emulator Lite** (Specifies the onboard emulator on the TB-RX23W board.)
- **Interface: FINE**
- **Power: None**

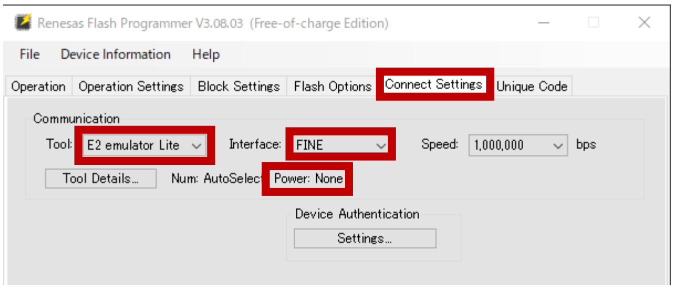

Figure 2-5 Settings on Connect Settings Tab

(4) Step 4: Programming to the TB-RX23W Board

Follow the operation procedure of the Renesas Flash Programmer to program the firmware to the TB-RX23W board.

(5) Step 5: ESW1 Switch Setting after Programming to TB-RX23W Board

Set ESW1 2-4 to OFF to run the demo.

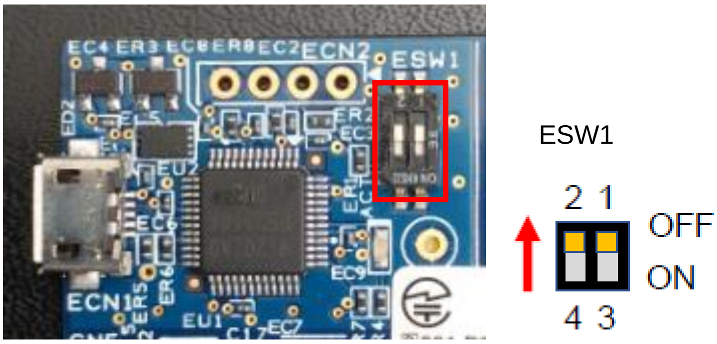

Figure 2-6 ESW1 Switch Setting after Programming to TB-RX23W Board

(6) Step 6: Connecting the PC to the TB-RX23W board

Use a USB cable to connect the PC to the USB serial converter connector (CN5) as shown in Figure 2-7.

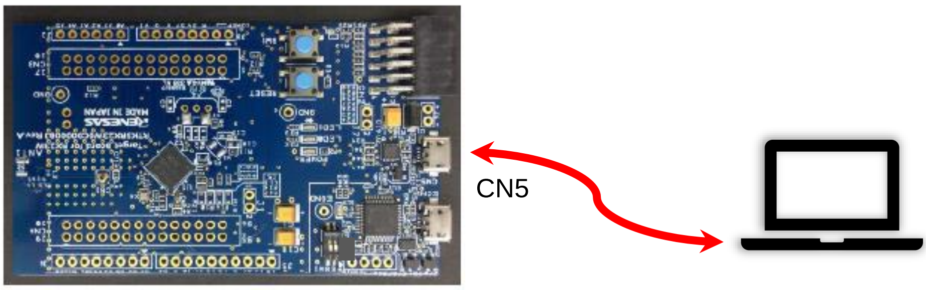

Figure 2-7 Connecting the PC to the TB-RX23W Board

3. Demo

The procedure for running the demo is described below.

3.1 Demo Overview

Table 3.1 lists the available demo operations.

Table 3-1 Overview Description of Demo Operations

Demo Application	Overview of Demo Operations
2nd OTA update on TB-RX23W board	• Creates an OTA update job by Python script. • Updates the firmware on the TB-RX23W board via the RSK+RX65N-2MB board.
Uploading to cloud and visualization of sensor data	• Obtains sensor data from HS3001 sensor and ZMOD4410 sensor and outputs a log. • Displays sensor data on Pmod LCD. • Transmits sensor data to AWS. • Performs visualization of HS3001 sensor data and ZMOD4410 sensor data using AWS OpenSearch. • Changes operation by applying a firmware update by 2nd OTA update. [Before update] <RSK+RX65N-2MB board> Displays the received sensor data of HS3001 sensor on Pmod LCD. <TB-RX23W board> Measures using the HS3001 sensor. LED1 blinks. [After update] <RSK+RX65N-2MB board> Displays the received sensor data of the HS3001 sensor and ZMOD4410 sensor on Pmod LCD. <TB-RX23W board> Measures using the HS3001 sensor and ZMOD4410 sensor. Both LED1 and LED2 blinks simultaneously.

3.2 Hardware Environment

Figure 3-1 shows the hardware configuration of the demo.

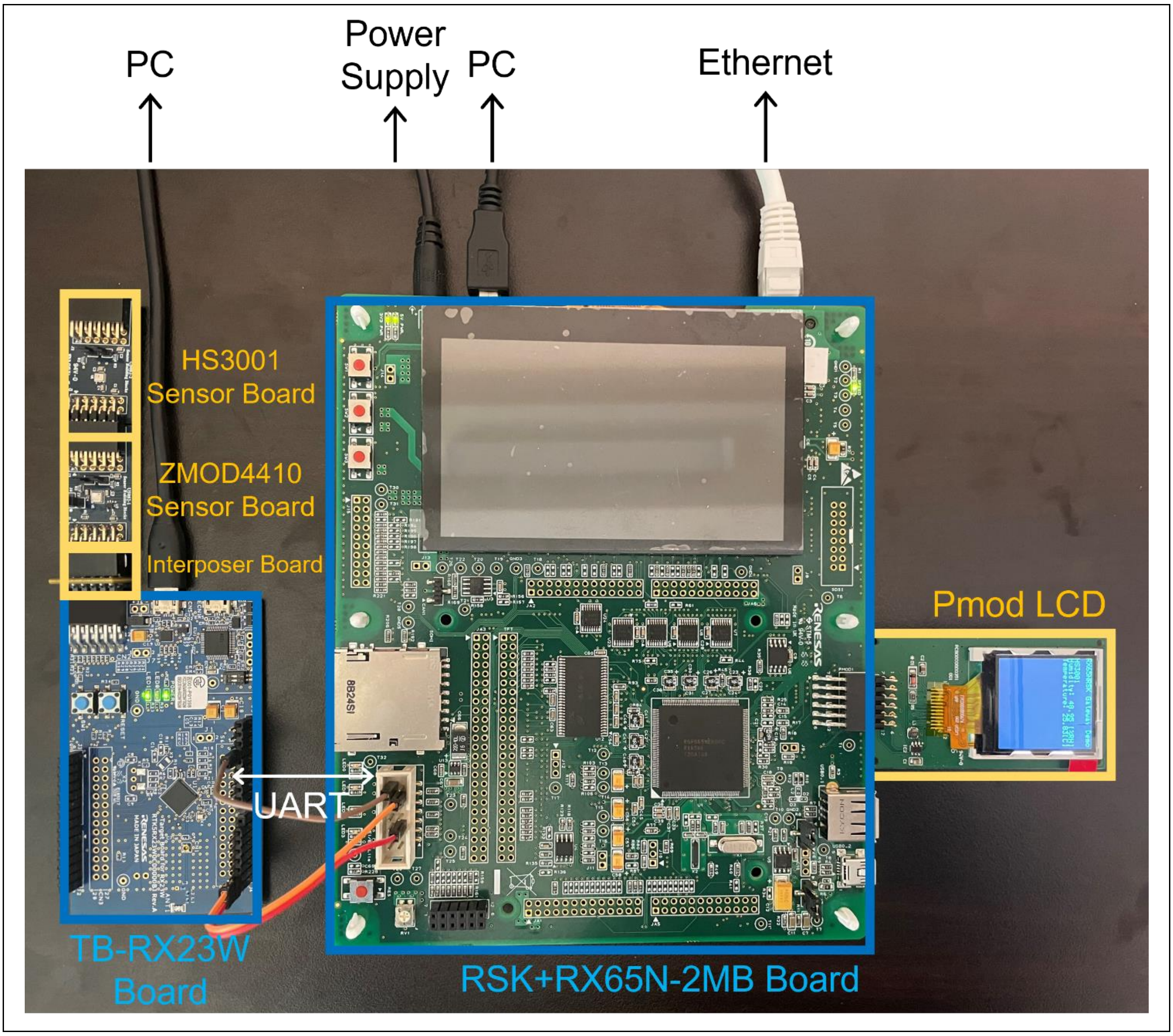

Figure 3-1 Hardware Configuration of Demo

3.2.1 Configuration of RSK+RX65N-2MB Board

The procedure for configuring the RSK+RX65N-2MB board is described below.

(1) Connecting the Pmod LCD Board

Connect the Pmod LCD board to the Pmod connector (PMOD1) on the RSK+RX65N-2MB board.

(2) Connections between TB-RX23W Board and Pins for UART Communication

Signals for UART communication are extracted from the E1/E2 Lite debugger connector as indicated below.

Make connections to the TB-RX23W board as shown in Figure 3-4.

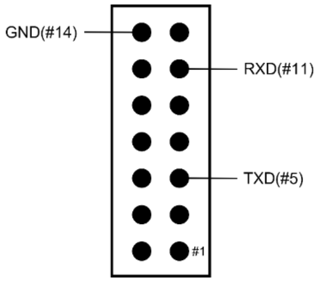

Figure 3-2 Extraction of UART Signals from E1/E2 Lite Debugger Connector

(3) Connection between PC and RSK+RX65N-2MB Board via Serial Cable for Log Output

It is possible to output a log from the RSK+RX65N-2MB board.

Connect the PC and the USB serial converter connector using a USB cable. Refer to Figure 3-1, Hardware Configuration of Demo.

3.2.2 Configuration of TB-RX23W Board

The procedure for configuring the TB-RX23W board is described below.

(1) Connections between RSK+RX65N-2MB Board and Pins for UART Communication

Signals for UART communication are extracted from pins 7 and 8 of the J4 connector and pin 4 of the J5 connector as indicated below.

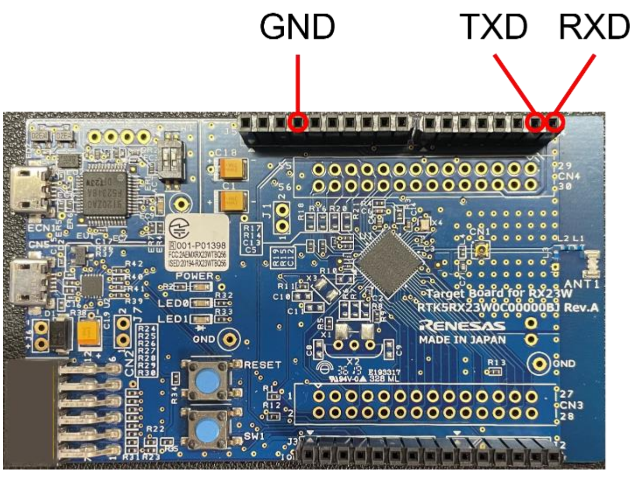

Figure 3-3 Extraction of UART Signals from J4 and J5 Connectors

Make connections to the UART pins on the RSK+RX65N-2MB board as shown in Figure 3-4.

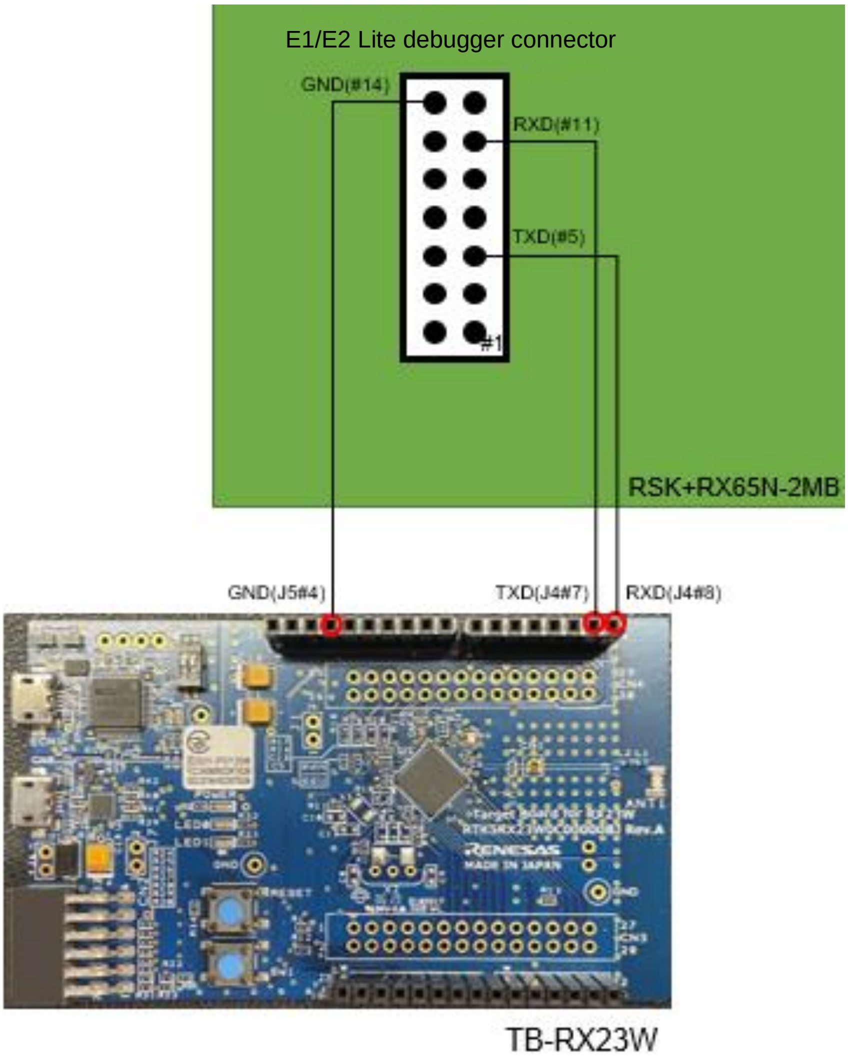

Figure 3-4 UART Connections between RSK+RX65N-2MB Board and TB-RX23W Board

(2) Confirming ESW1 Switch Setting on TB-RX23W Board

Make the following setting to run the demo.

- To execute firmware without using the onboard emulator: Set ESW1 2-4 to OFF.

The demo will not operate properly if the ESW1 switch setting is incorrect.

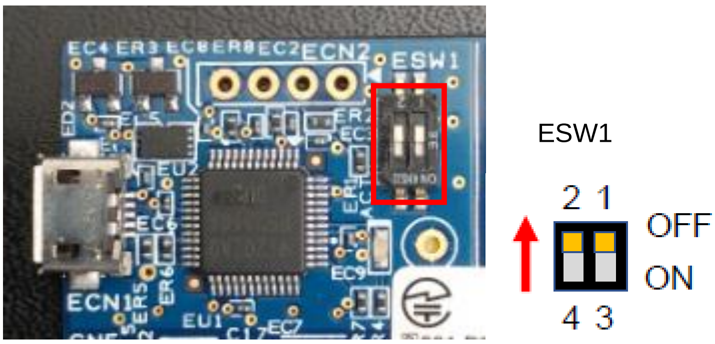

Figure 3-5 ESW1 Switch Setting on TB-RX23W Board for Demo

(3) Connecting HS3001 Sensor Board and ZMOD4410 Sensor Board to TB-RX23W Board

Use a multi-stage connection to connect the ZMOD4410 sensor board (US082-ZMOD4410EVZ) and HS3001 sensor board (US082-HS3001EVZ) to the PMOD connector (CN2) via the converter board (US082-INTERPEVZ), as shown in Figure 3-6. The demo will still work without problems if the order of the sensor boards is reversed.

Confirm the mark indicating pin 1 on the converter board (US082-INTERPEVZ) and confirm to plug it into the PMOD connector (CN2) correctly.

Short both pairs of jumper pins (J4 and J5) on the HS3001 sensor board (US082-HS3001EVZ) to enable pull-up processing of I^{2}C bus signals. Also short jumper pins J3, J4, and J5 on the ZMOD4410 sensor board (US082-ZMOD4410EVZ).

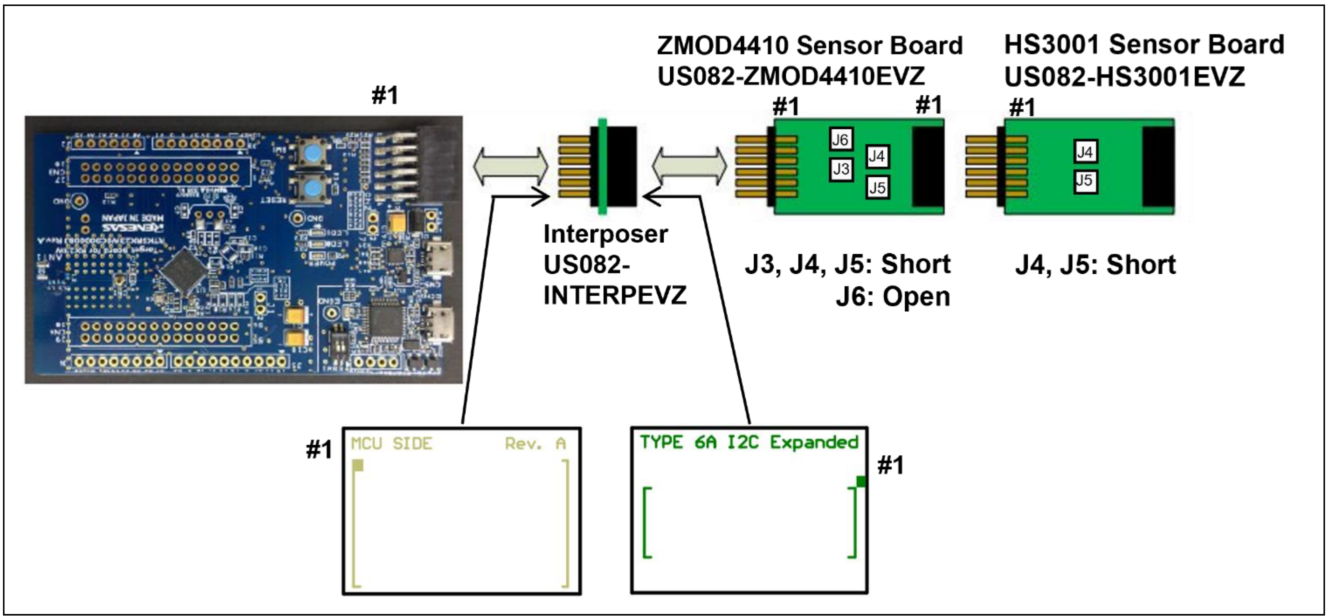

Figure 3-6 Connecting the HS3001 and ZMOD4410 Sensor Boards to the TB-RX23W Board

(4) Connecting the Serial Cable for Log Output to PC and TB-RX23W Board

Each TB-RX23W board can produce log output.

Use a USB cable to connect the PC to the USB serial converter connector (CN5) as shown in Figure 3-7.

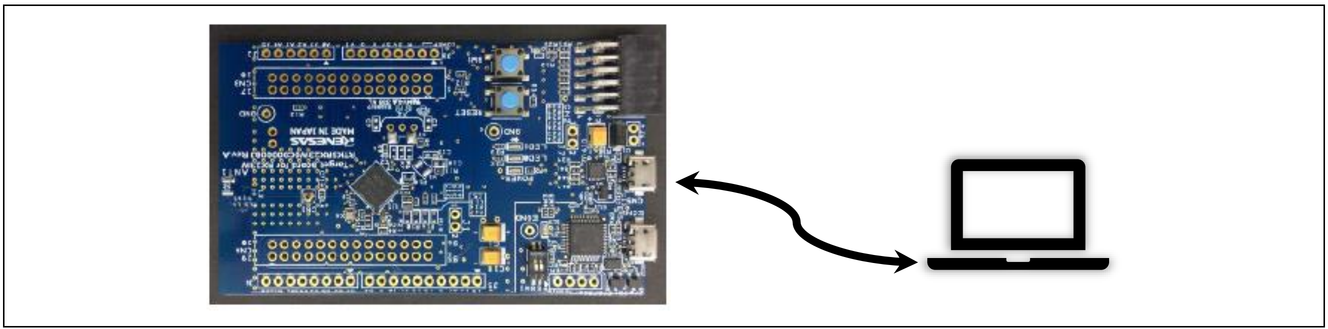

Figure 3-7 Serial Cable Connection between PC and TB-RX23W Board

3.3 Software Environment

3.3.1 Creating and Programming Initial Firmware for RSK+RX65N-2MB Board

Create and program the initial firmware for the RSK+RX65N-2MB board.

The procedure is shown below.

(1) Step 1: Project Import

Import the **rx65n_rsk_bootloader** project of the Boot Loader and the **aws_demos** project of the User Program for the RSK+RX65N-2MB board into e^2 studio.

Note: When importing the **aws_demos** project, extract the sample code zip file, select the extracted folder as the root directory, and then uncheck "Copy Project to Workspace" as shown in Figure 3-8.

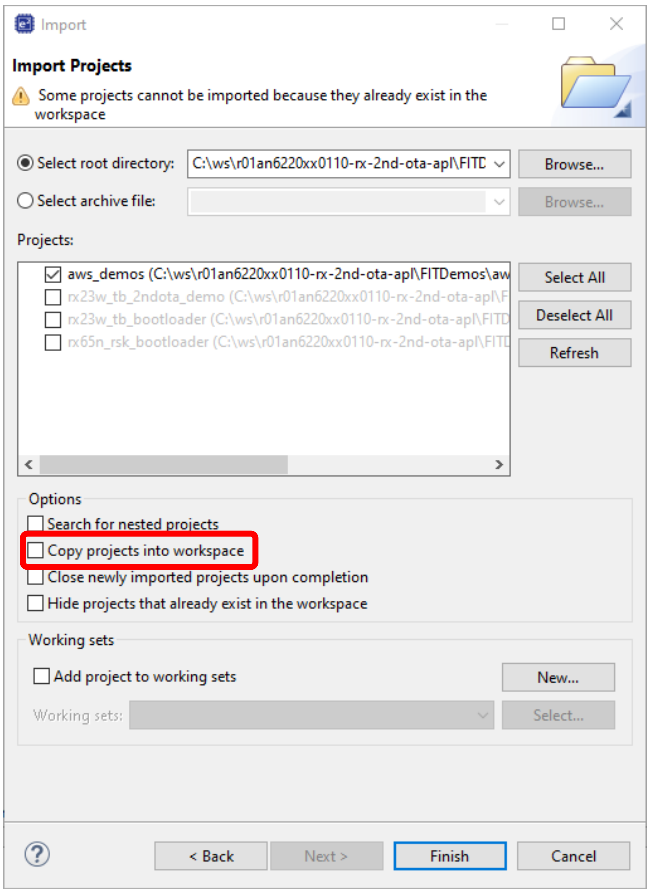

Figure 3-8 e2studio Import Projects Screen

(2) Step 2: Project Modification

Refer to steps ⑪ to ⑮ in section 2.1 of How to implement FreeRTOS OTA by using Amazon Web Services on RX65N, and make the following changes to the **rx65n_rsk_bootloader** project and **aws_demos** projects.

- **rx65n_rsk_bootloader** project
  — Enter the public key information for code-signer verification.

- **aws_demos** project
  — Enter the MQTT broker endpoint.
  — Enter the thing name.
  — Replace **aws_clientcredential_keys.h**.
  — Enter the code-signing certificate.

signingcredentialSIGNING_CERTIFICATE_PEM in aws_ota_codesigner_certificate.h in step ⑮ in section 2.1 of the above application note should be replaced by otapalconfigCODE_SIGNING_CERTIFICATE in ota_demo_config.h.

Figure 3-2 shows the storage locations of the files that need to be changed in both the **rx65n_rsk_bootloader** project and **aws_demos** project.

Table 3-2 Locations of Files Requiring Changes

File Name	Location in e^2 studio Project Explorer
code_signer_public_key.h	rx65n_rsk_bootloader/src/key/
aws_clientcredential.h	aws_demos/demos/include/
aws_clientcredential_keys.h	
ota_demo_config.h	aws_demos/config_files/

(3) Step 3: Project Building

Build both projects and generate MOT files.

(4) Step 4: Initial Firmware Concatenating

Concatenate the generated MOT files for the Boot Loader project and the User Program project to create the initial firmware for the RX65N. Use the Renesas Secure Flash Programmer*1 to concatenate the MOT files.

Note: 1. Use the version 1.x.x of the Renesas Secure Flash Programmer. The version 2.x.x of the Renesas Secure Flash Programmer is not supported the Amazon FreeRTOS projects currently.

Launch **Renesas Secure Flash Programmer.exe** and select **Initial Firm**. Then enter the following settings as shown in Table 3-3 and click **Generate**.

Table 3-3 Renesas Secure Flash Programmer Settings (RSK+RX65N-2MB Board)

Select MCU	RX65N(ROM 2MB)/Secure Bootloader=256KB
Select Firmware Verification Type	sig-sha256-ecdsa
Private Key Path (PEM Format)	Path to private key generated previously
Select Output Format	Bank0 User Program + Boot Loader (Motorola S Format)
Boot Loader File Path (Motorola Format)	Path to MOT file of Boot Loader for RSK+RX65N-2MB board generated previously
Bank0 User program Firmware Sequence Number	1
Bank0 File Path (Motorola Format)	Path to MOT file of User Program for RSK+RX65N-2MB board generated previously

When the message **Generate succeeded** appears in the yellow frame below in the Renesas Secure Flash Programmer, the Boot Loader and User Program MOT files have been concatenated to generate the file **userprog.mot**.

(5) Step 5: Initial Firmware Programming

Refer to section 2, Firmware Programming, and program the initial firmware to the RX65N.

3.3.2 Creating and Programming Initial Firmware for TB-RX23W Board

Create and program the initial firmware for the TB-RX23W board.

The procedure is shown below.

(1) Step 1: Project Import

Import the rx23w_tb_bootloader project of the Boot Loader and the rx23w_tb_2ndota_demo project for the TB-RX23W board into e^2 studio.

(2) Step 2: Project Modification by Entering Public Key Information

For both the **rx23w_tb_bootloader** project and the rx23w_tb_2ndota_demo project, enter the public key information for code-signer verification in **CODE_SIGNER_PUBLIC_KEY_PEM** in the file **src/key/code_signer_public_key.h**.

(3) Step 3: Project Building

Build both projects and generate MOT files.

(4) Step 4: Initial Firmware Concatenating

Concatenate the generated MOT files for the Boot Loader project and the User Program project to create the initial firmware for the RX23W. Use the Renesas Secure Flash Programmer*1 to concatenate the MOT files.

Note: 1. Use the version 1.x.x of the Renesas Secure Flash Programmer. The version 2.x.x of the Renesas Secure Flash Programmer is not supported the Amazon FreeRTOS projects currently.

Launch **Renesas Secure Flash Programmer.exe** and select **Initial Firm**. Then enter the following settings as shown in Table 3-4 and click **Generate**.

Table 3-4 Renesas Secure Flash Programmer Settings (TB-RX23W Board)

Select MCU	RX23W(ROM 512KB)/Secure Bootloader=64KB
Select Firmware Verification Type	sig-sha256-ecdsa
Private Key Path (PEM Format)	Path to private key generated previously
Select Output Format	Bank0 User Program + Boot Loader (Motorola S Format)
Boot Loader File Path (Motorola Format)	Path to MOT file of Boot Loader for TB-RX23W board generated previously
Bank0 User program Firmware Sequence Number	1
Bank0 File Path (Motorola Format)	Path to MOT file of User Program for TB-RX23W board generated previously

When the message **Generate succeeded** appears in the yellow frame below in the Renesas Secure Flash Programmer, the Boot Loader and User Program MOT files have been concatenated to generate the file **userprog.mot**.

(5) Step 5: Initial Firmware Programming

Refer to section 2, Firmware Programming, and program the initial firmware to the RX23W.

3.3.3 Terminal Emulator Software Settings

Terminal emulator software (e.g., Tera Term) is required to generate log output using serial communication.

(1) Serial Port Settings

The serial port settings are shown below.

Table 3-5 Serial port settings

Item	Setting
Baud rate	115,200 bps
Data	8 bits
Parity	None
Stop	1 bit
Flow control	None

3.3.4 AWS Settings

The AWS settings can be divided into the following two categories.

- 1. Settings for OTA updating
- 2. Settings for visualization of sensor data

The details are shown below.

(1) Settings for OTA Updating

Perform the settings described in section 1, Set up AWS of How to implement FreeRTOS OTA by using Amazon Web Services on RX65N.

Of the names included in these settings, the following are used by the demo.

- IAM user name
- Thing name
- S3 bucket name
- Service role name for OTA updates
- Code-signing certificate profile name

(2) Settings for Visualization of Sensor Data

The settings for Visualization can be divided into the following four phases.

- a. AWS: Settings for Amazon Open Search Service
- b. OpenSearch Dashboards: Settings
- c. AWS: Settings for AWS IoT Core
- d. OpenSearch Dashboards: Data Browsing

Set in the order of a → b → c → d.

(a) AWS: Settings for Amazon OpenSearch Service

The procedure is shown below.

1. Select **Services** → **Analytics** → **Amazon OpenSearch Service.**

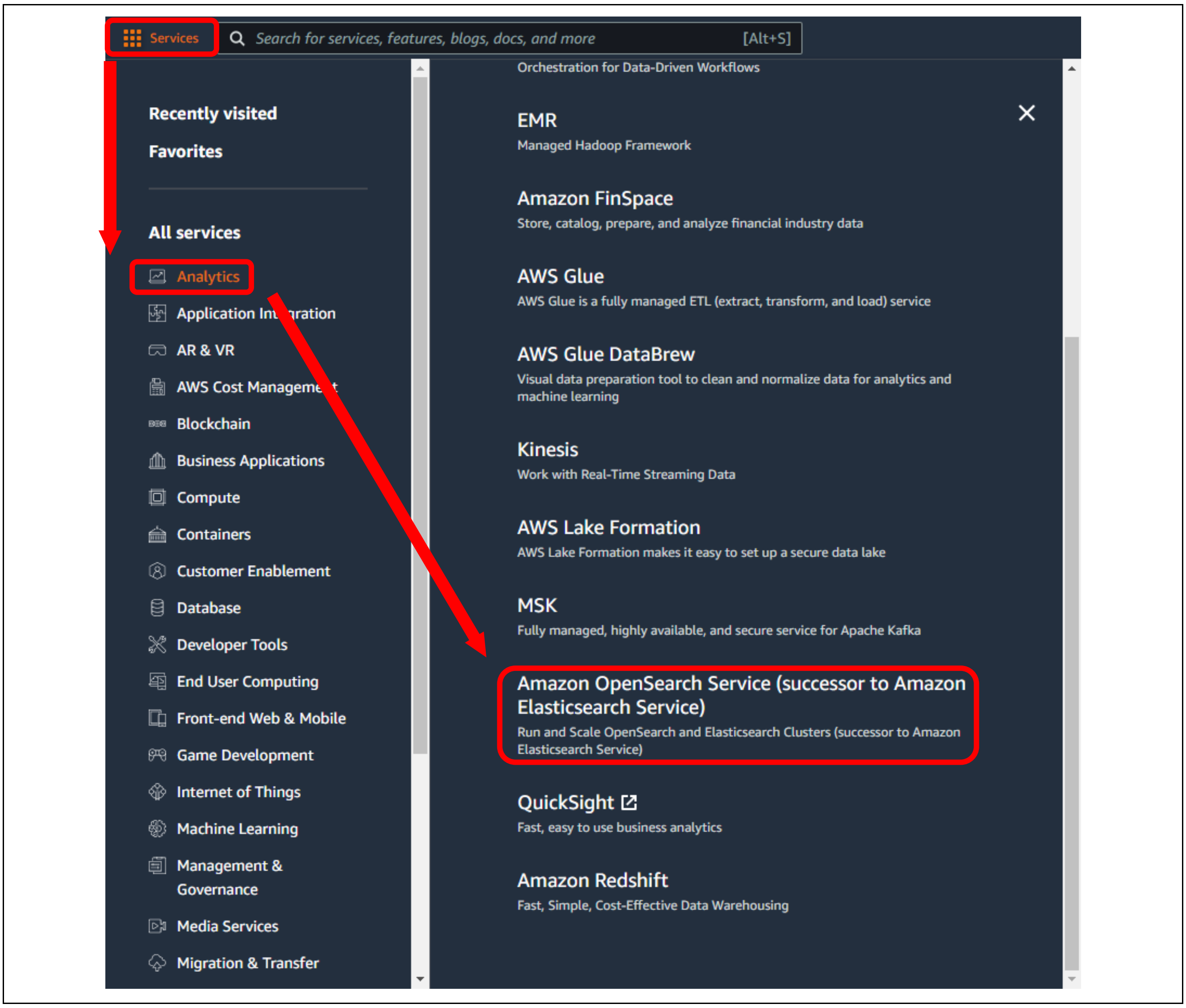

Figure 3-9 Services Screen of AWS

2. Amazon OpenSearch Service/Dashboard:

   Click **Create domain**.

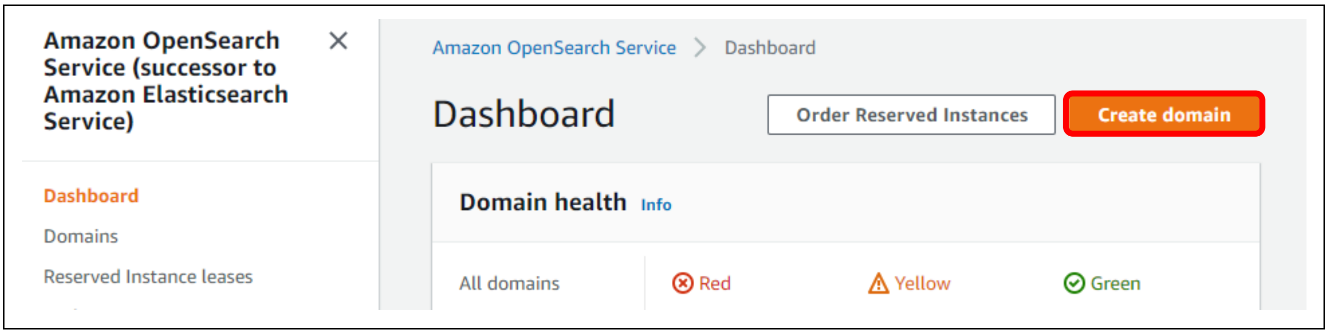

Figure 3-10 Amazon OpenSearch Service/Dashboard: Dashboard Screen

3. Amazon OpenSearch Service/Domains/Create domain: **Name**

   Set **Domain name** to name.
   Figure 3-11 is an example of creating the domain name **test-domain**.
   In the following, only the parts that need to be set are described.

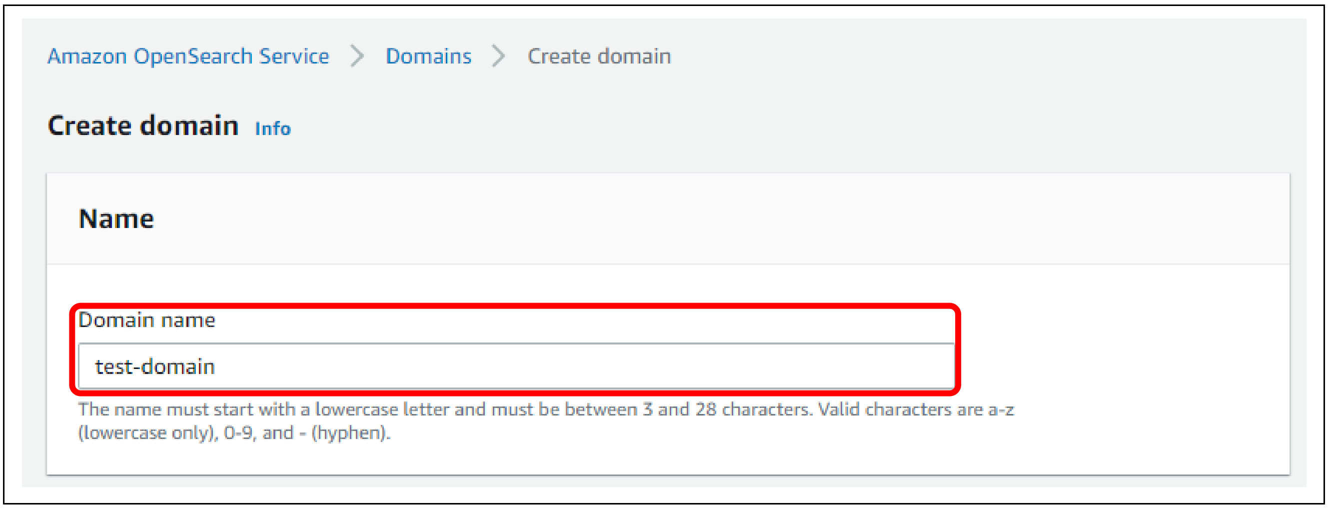

Figure 3-11 Amazon OpenSearch Service/Domains/Create Domain: Name Screen

4. Amazon OpenSearch Service/Domains/Create domain: **Deployment type**

   Select **Development and testing**.

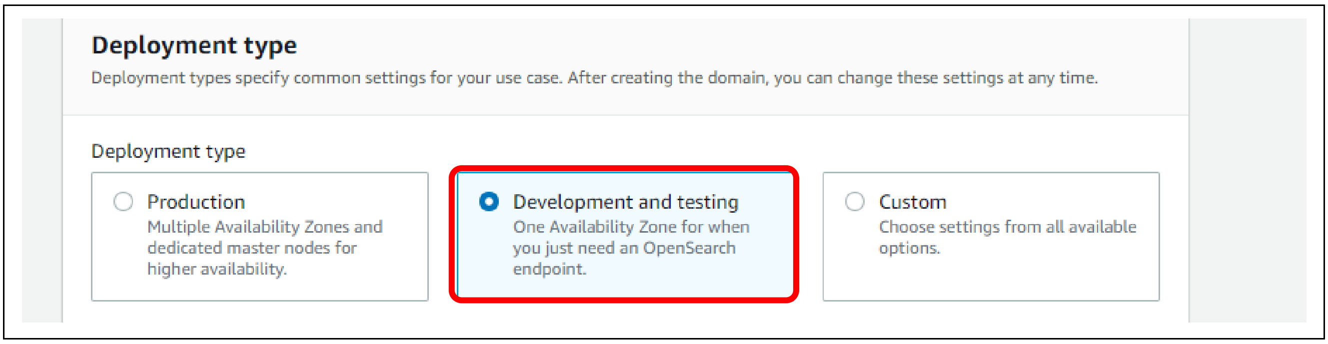

Figure 3-12 Amazon OpenSearch Service/Domains/Create Domain: Deployment Type Screen

5. Amazon OpenSearch Service/Domains/Create domain: **Data nodes**

Set **Instance type** to **t2.small.search**.

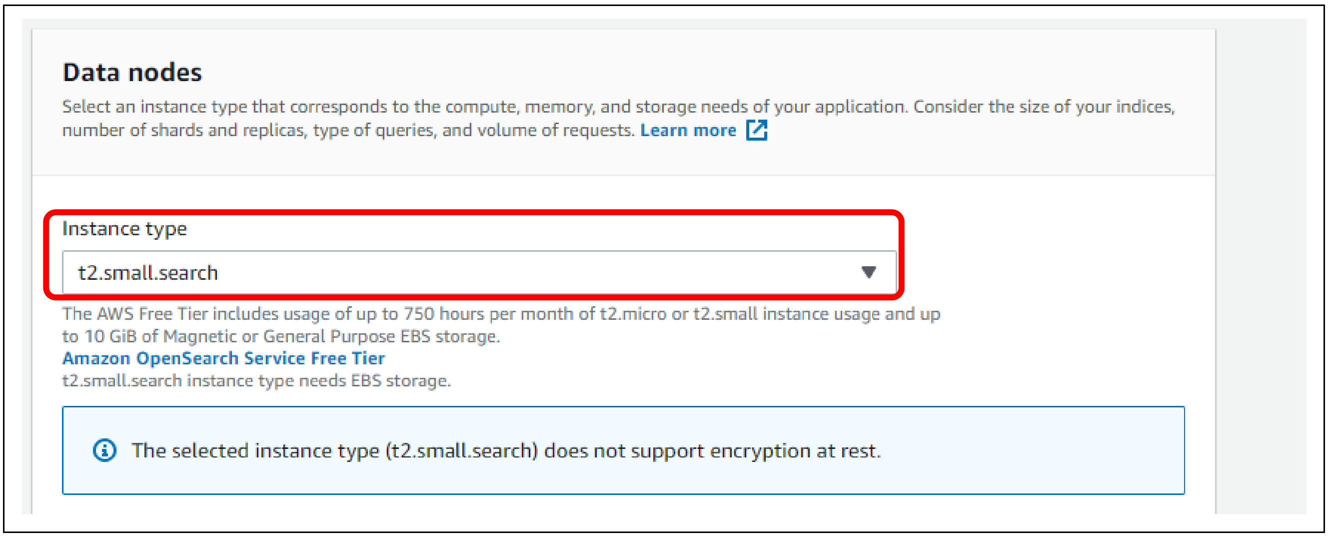

Figure 3-13 Amazon OpenSearch Service/Domains/Create Domain: Data Nodes Screen

6. Amazon OpenSearch Service/Domains/Create domain: **Network**

Select **Public access**.

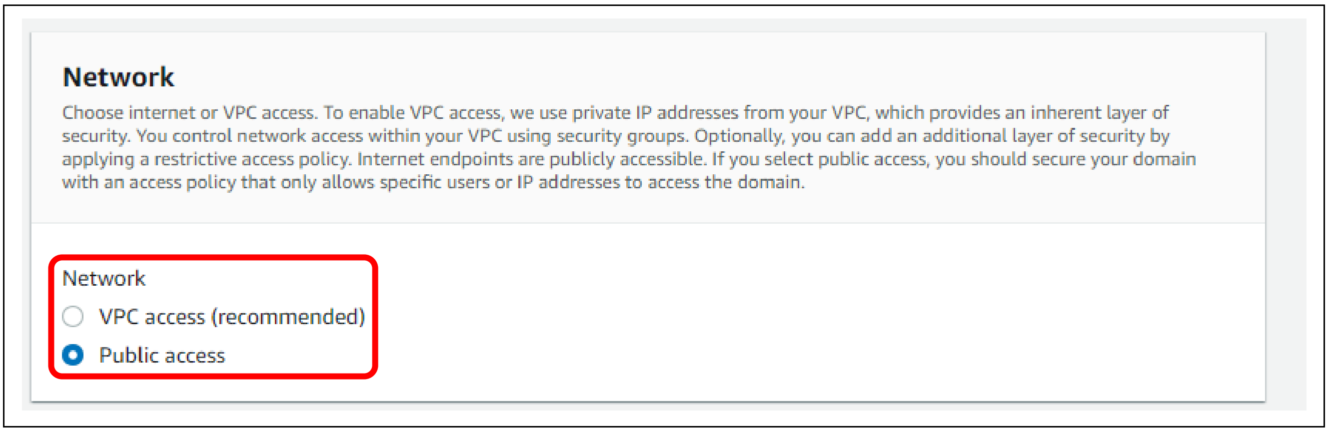

Figure 3-14 Amazon OpenSearch Service/Domains/Create Domain: Network Screen

7. Amazon OpenSearch Service/Domains/Create domain: **Access policy** and **Elements**

Set **Domain access policy** to **Configure domain level access policy**.
Set **type** of **Elements** to **IPv4 address, Principal** of Elements to **global IP address** of your PC and the **Action** of Elements to **Allow**.

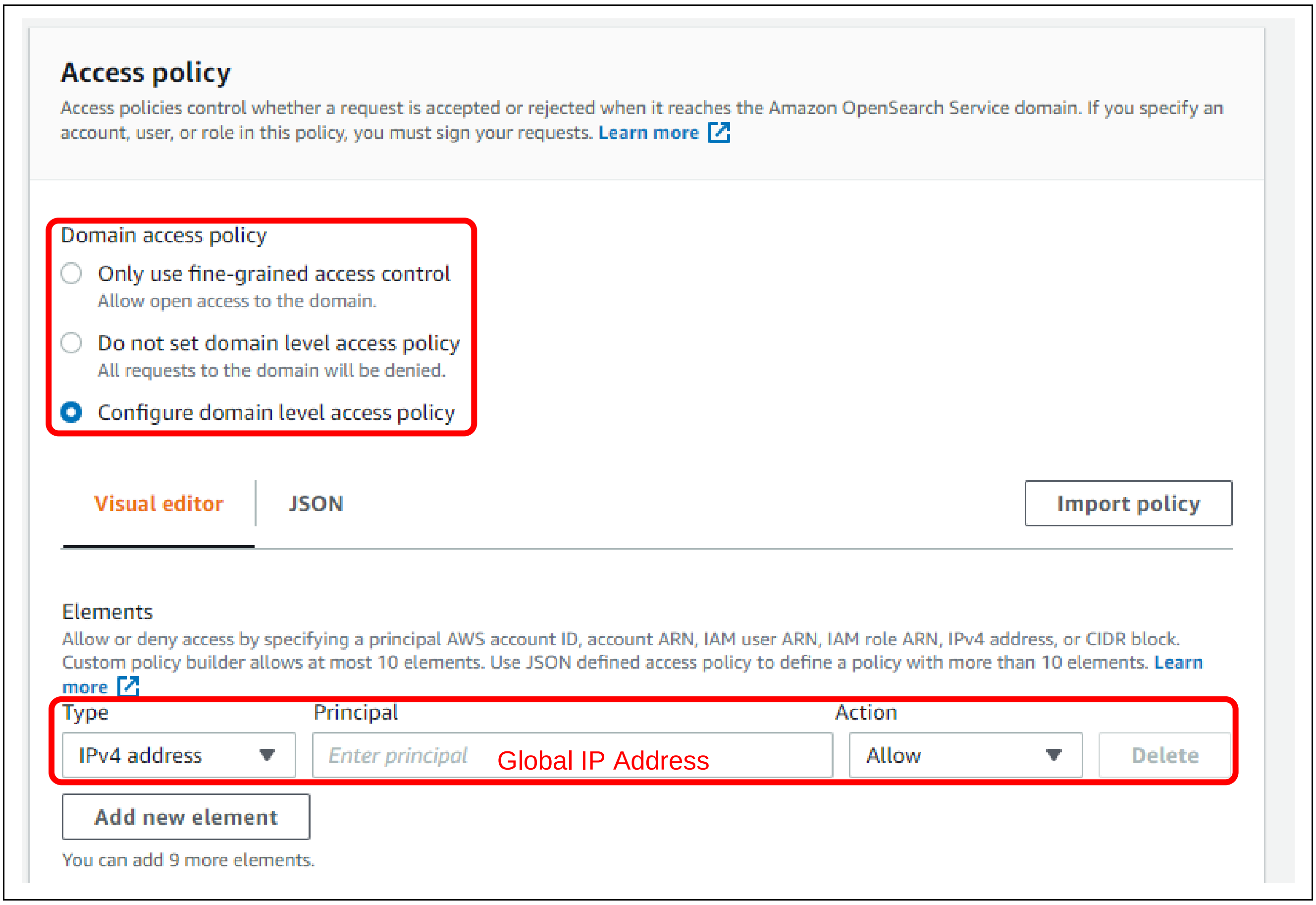

Figure 3-15 Amazon OpenSearch Service/Domains/Create Domain: Access Policy Screen

8. Amazon OpenSearch Service/Domains/Create domain:

Click **Create**.

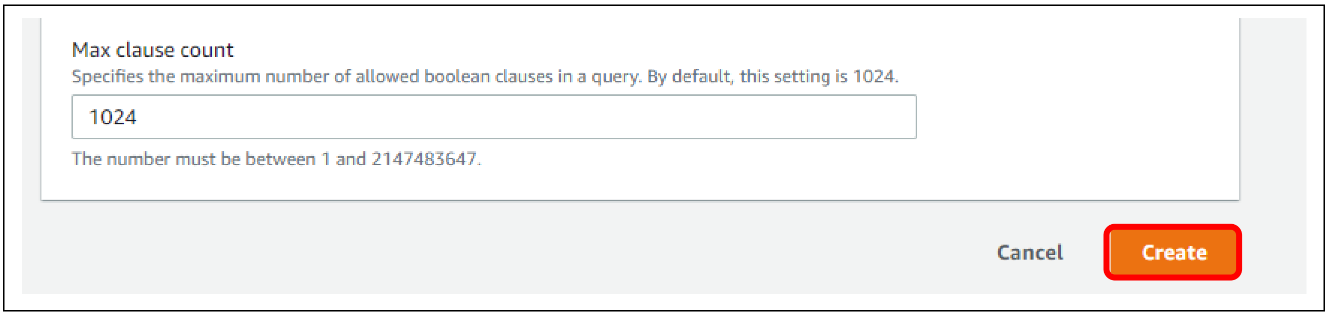

Figure 3-16 Amazon OpenSearch Service/Domains/Create Domain: Domain Creation Screen

9. Amazon OpenSearch Service/Domains/(Created domain name): Domain Creating

Figure 3-17 is an example of creating the domain name **test-domain**.
Domain status will be **Loading**. It takes a few minutes to a few tens of minutes to complete the creation.

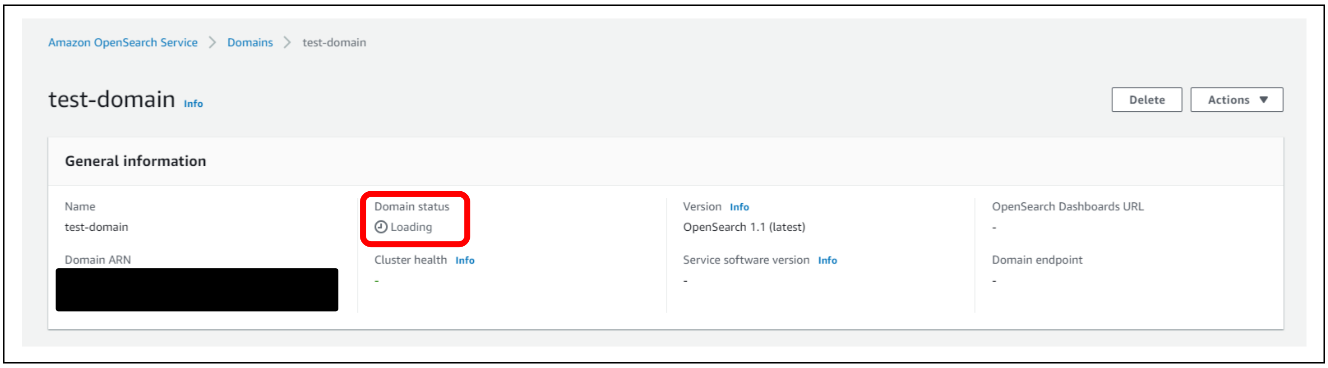

Figure 3-17 Amazon OpenSearch Service/Domains: Domain Creating Screen

10. Amazon OpenSearch Service/Domains/(Created domain name): Domain Creation Completed

The **Domain status** will be **Active** and the URL for OpenSearch Dashboards will be displayed.
Access the URL of OpenSearch Dashboards.

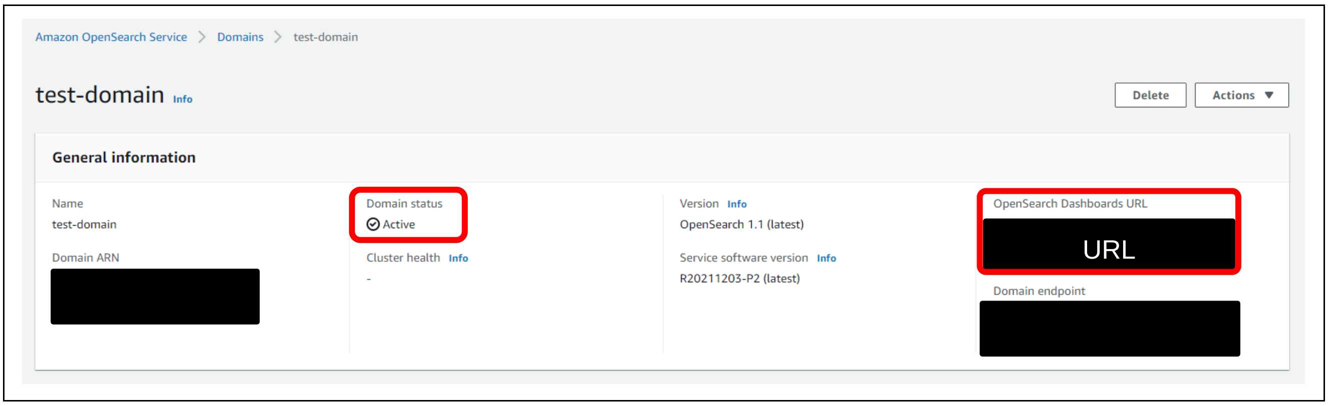

Figure 3-18 Amazon OpenSearch Service/Domains: Domain Creation Completed Screen

(b) OpenSearch Dashboards: Settings

The procedure is shown below.

1. After accessing the URL of OpenSearch Dashboards

   Click **Explore on my own** on the displayed screen.

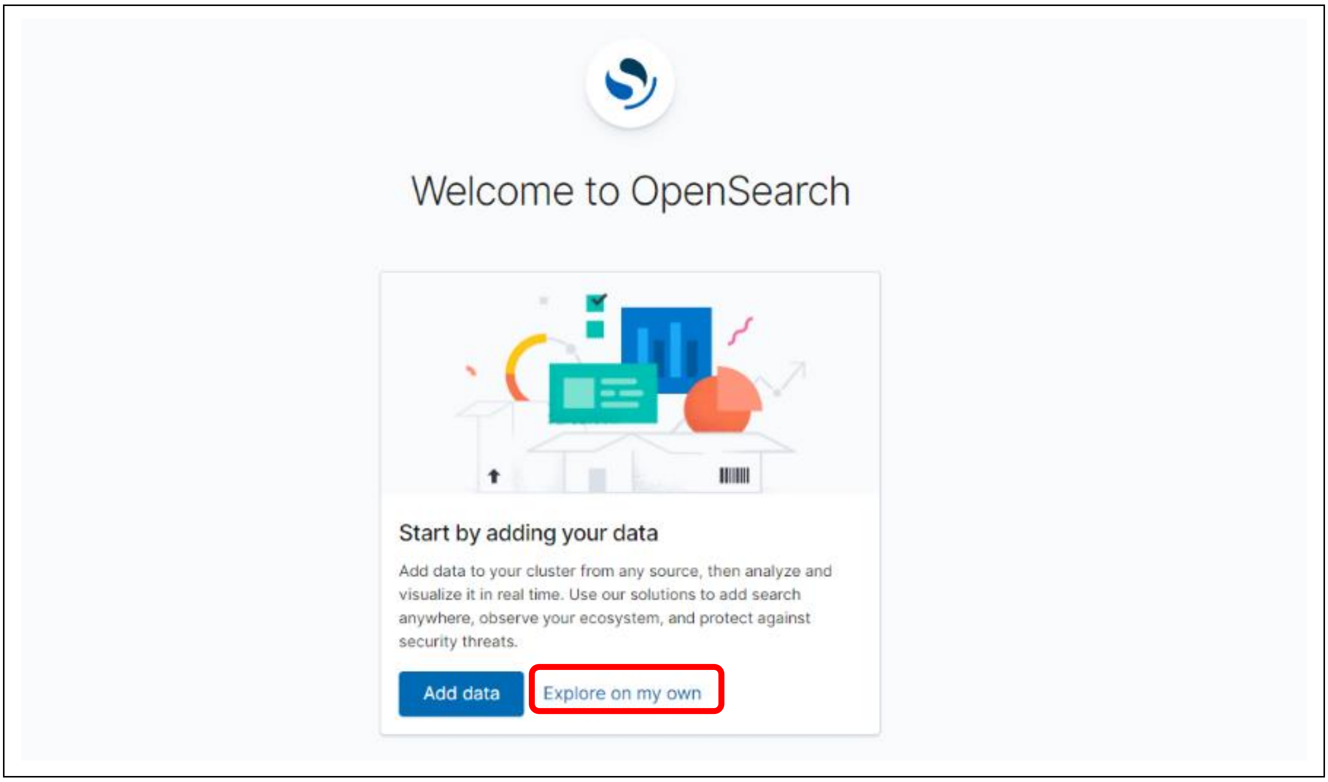

Figure 3-19 Screen of OpenSearch

2. Select " ≡ " → **Dev Tools**

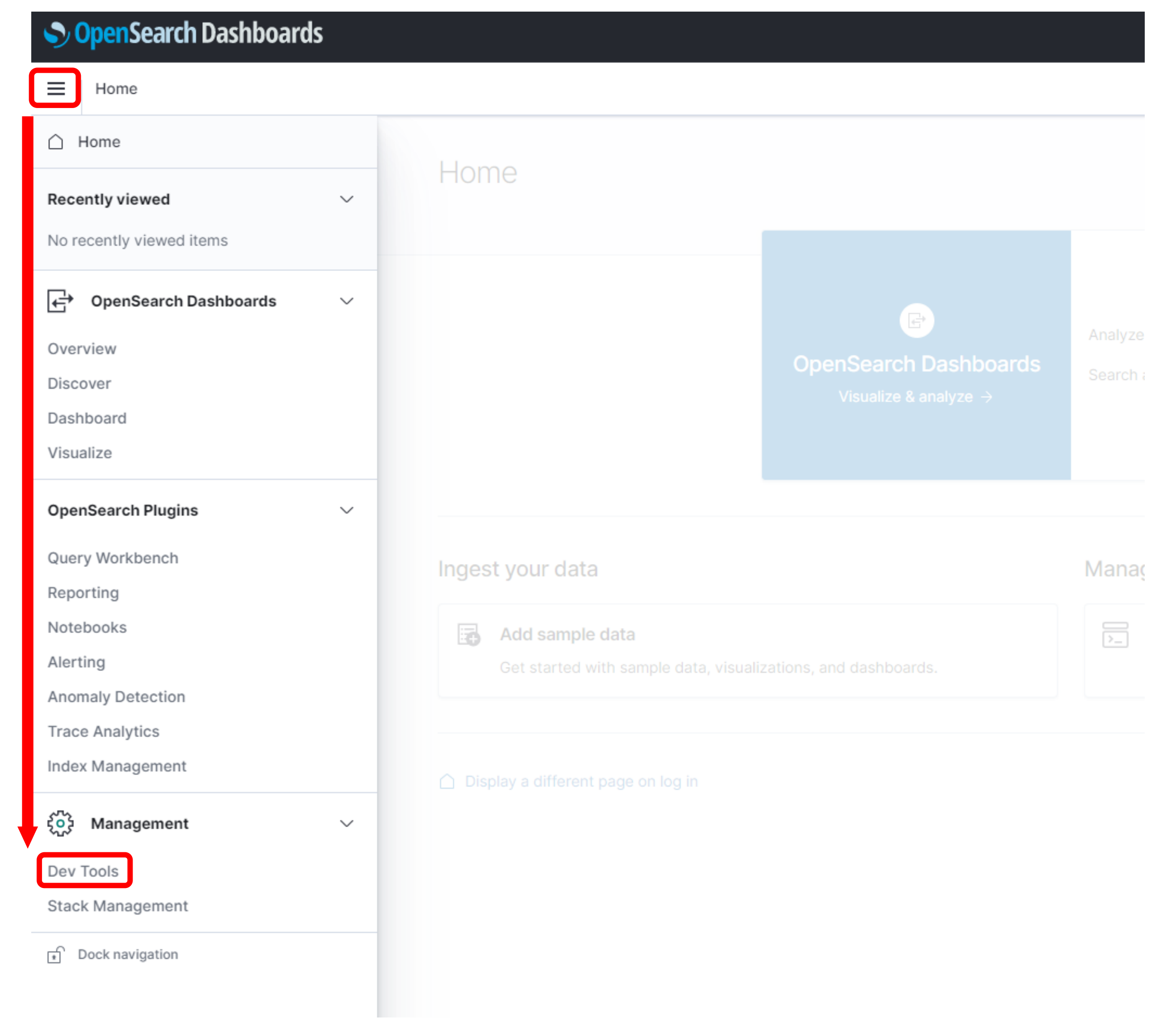

Figure 3-20 Screen of OpenSearch Dashboards

3. Dev Tools:

After transitioning to the Dev Tools screen, click **X** to close **Welcome to Console**.

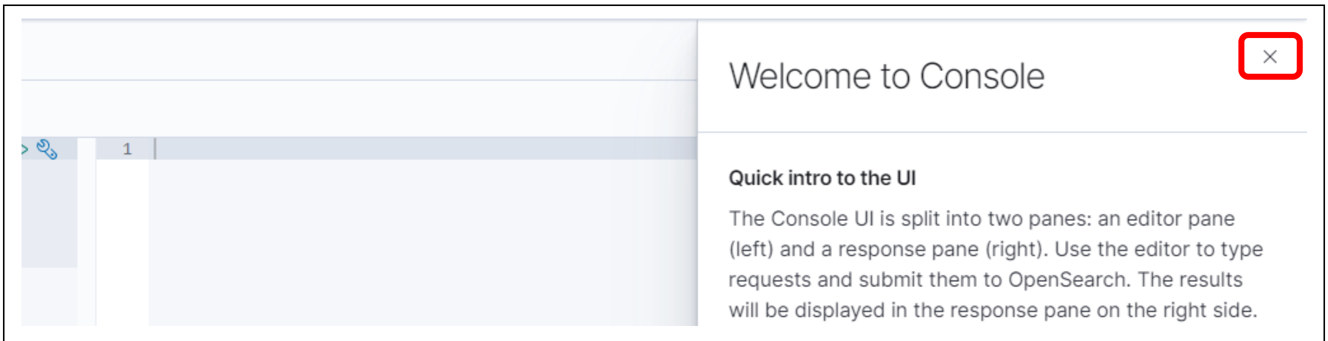

Figure 3-21 Dev Tools: Welcome to Console Screen

4. Dev Tools: Code Entering

Enter the code shown in Figure 3-23 in the left console of Figure 3-22 and click "▷" (Click to send request). The right console of Figure 3-22 shows the response.

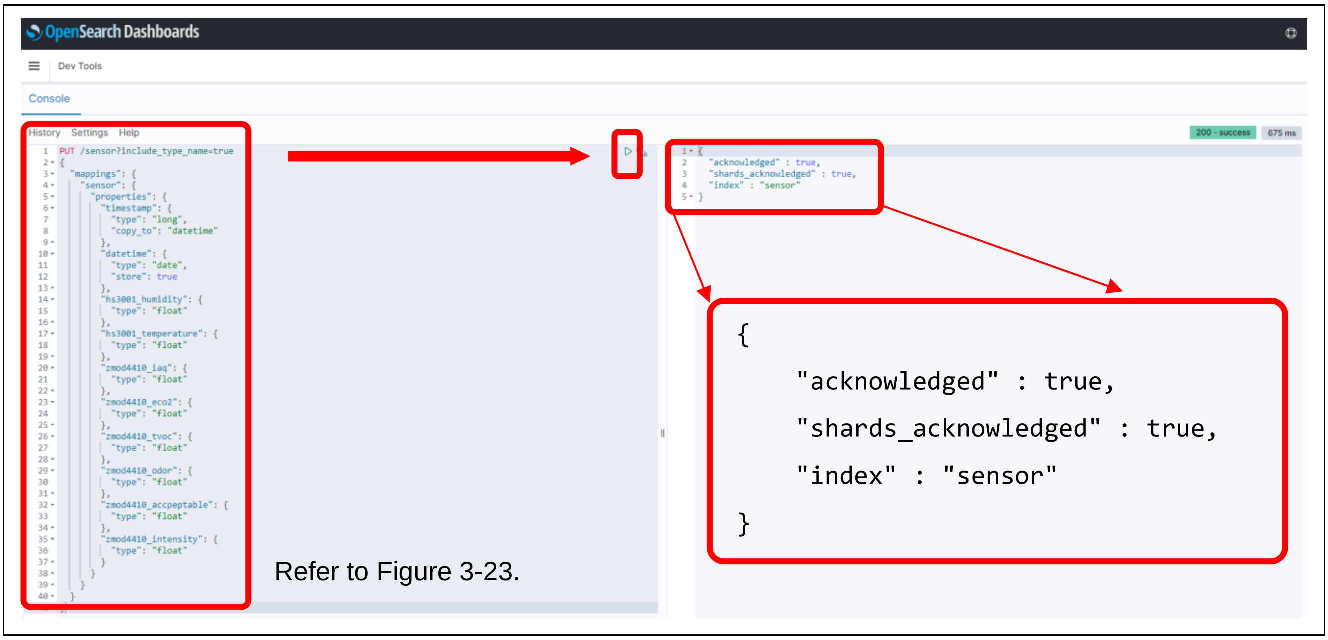

Figure 3-22 Dev Tools: Console Screen

The code you enter in the left console is shown below. **Be careful not to break the last line.**

```
PUT /sensor?include_type_name=true
{
  "mappings": {
    "sensor": {
      "properties": {
        "timestamp": {
          "type": "long",
          "copy_to": "datetime"
        },
        "datetime": {
          "type": "date",
          "store": true
        },
        "hs3001_humidity": {
          "type": "float"
        },
        "hs3001_temperature": {
          "type": "float"
        },
        "zmod4410_iaq": {
          "type": "float"
        },
        "zmod4410_eco2": {
          "type": "float"
        },
        "zmod4410_tvoc": {
          "type": "float"
        },
        "zmod4410_odor": {
          "type": "float"
        },
        "zmod4410_accpeptable": {
          "type": "float"
        },
        "zmod4410_intensity": {
          "type": "float"
        }
      }
    }
  }
}
```

Figure 3-23 Code to Enter in Console

(c) AWS: Settings for AWS IoT Core

The procedure is shown below.

1. Select **Services** → **Internet of Things** → **IoT Core**

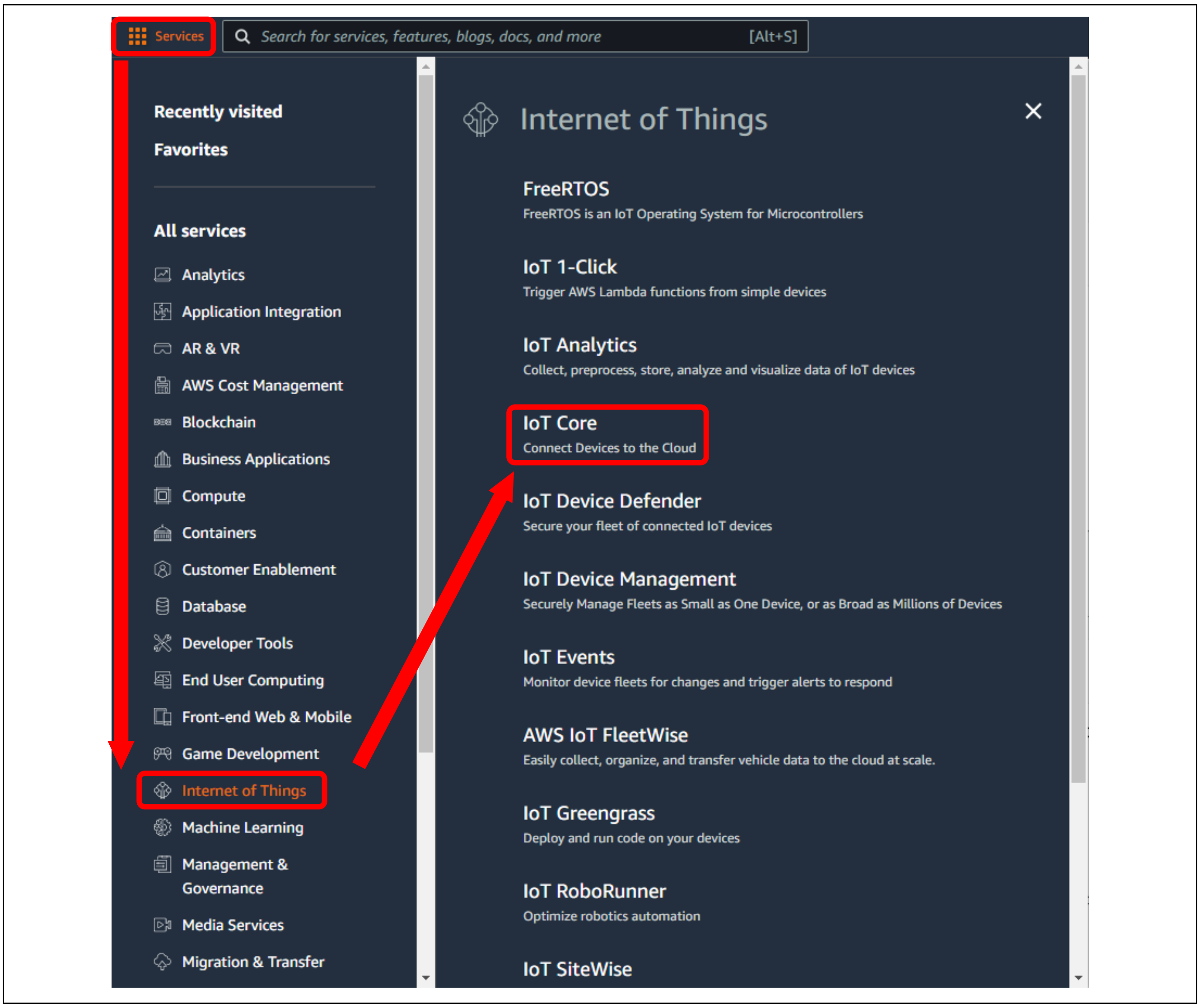

Figure 3-24 Services Screen of AWS

2. AWS IoT/Rules:

Select **Rules** and click **Create a rule**.

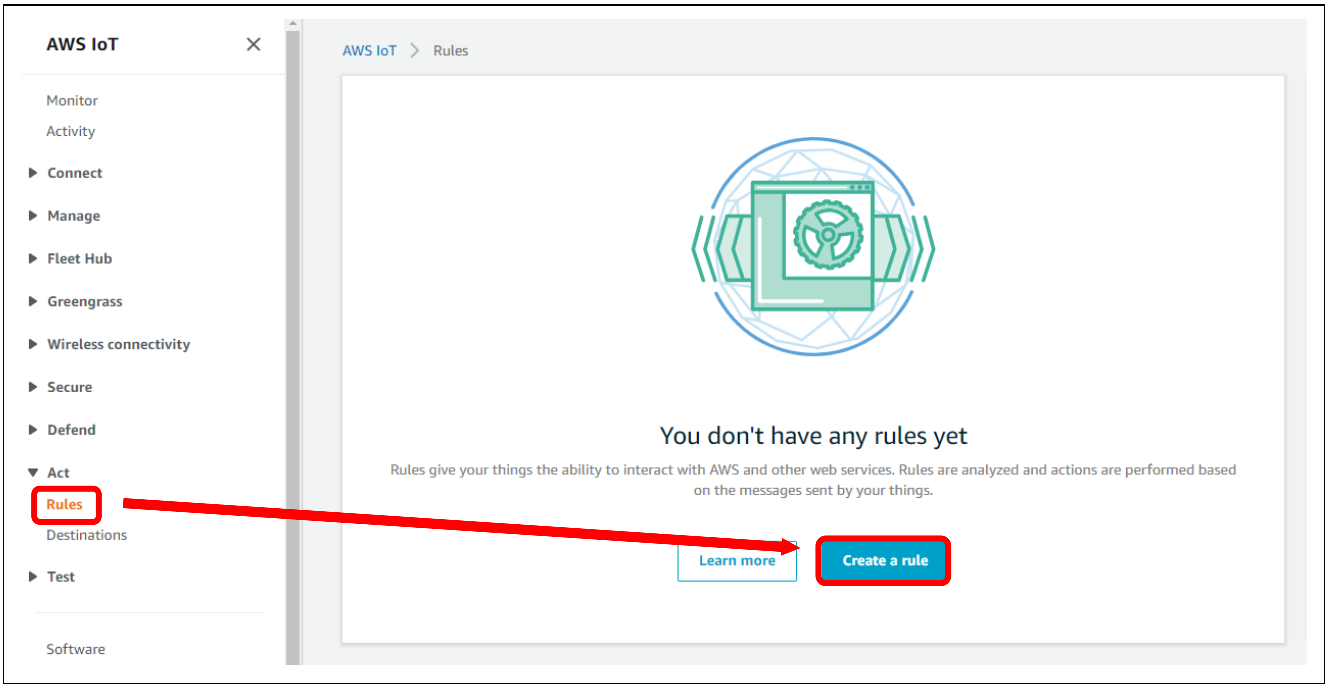

Figure 3-25 AWS IoT/Rules: Rule Creation Screen

3. AWS IoT/Rules/Create a rule: **Name**

Set **Name** to name.
Figure 3-26 is an example of creating the name **test_rule**.
In the following, only the parts that need to be set are described.

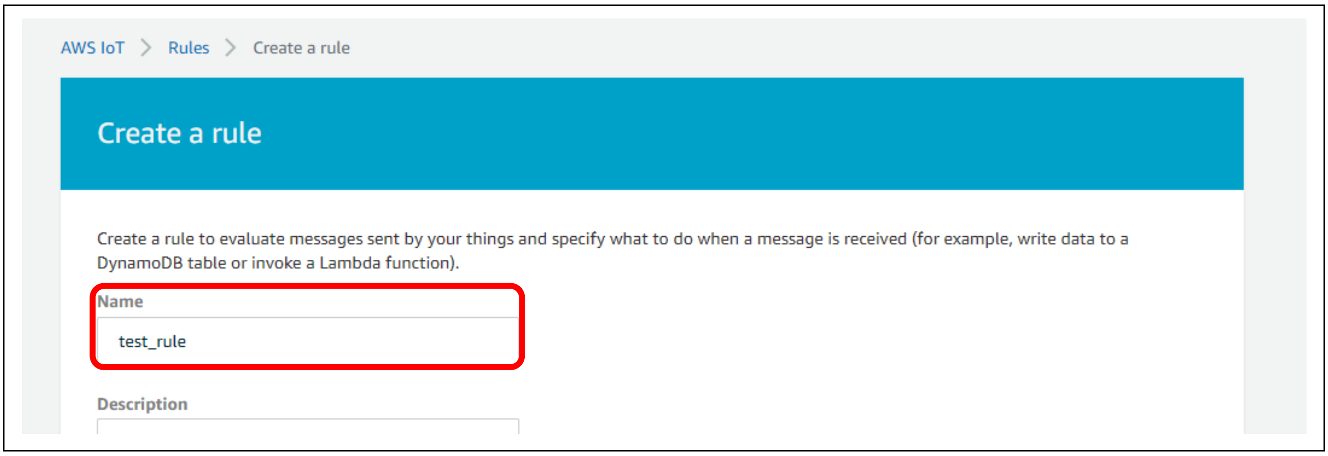

Figure 3-26 AWS IoT/Rules/Create a Rule: Name Screen

4. AWS IoT/Rules/Create a rule: **Rule query statement**

Enter the code below.

```
SELECT *, timestamp() as timestamp FROM 'iotdemo/topic/sensor'
```

Note: Be sure to start a new line after entering the code in the rule query statement.

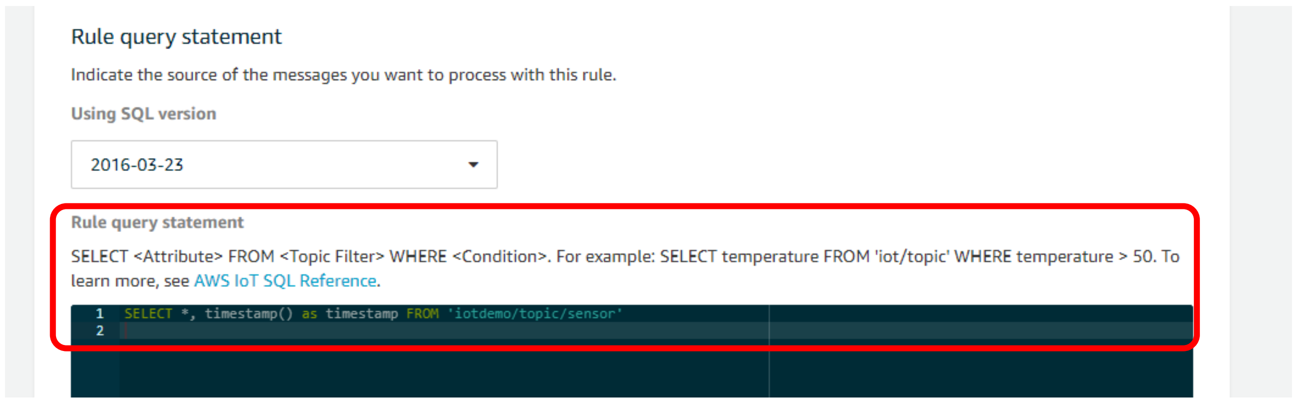

Figure 3-27 AWS IoT/Rules/Create a Rule: Rule Query Statement Screen

5. AWS IoT/Rules/Create a rule: **Set one or more actions** [before Adding Action]

Click Add action. Then, it will transition to the action addition processing.

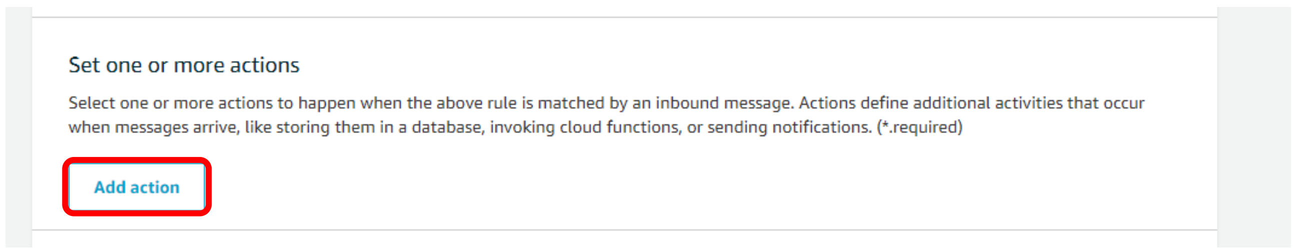

Figure 3-28 AWS IoT/Rules/Create a Rule: Set One or More Actions Screen [before Adding Action]

6. AWS IoT/Rules/Create a rule: Action Selection

When the action selection screen opens, select **Send a message to Amazon OpenSearch Service (the successor to Amazon Elasticsearch Service)**.
Click Configure action. Then, it will transition to the action configuration processing.

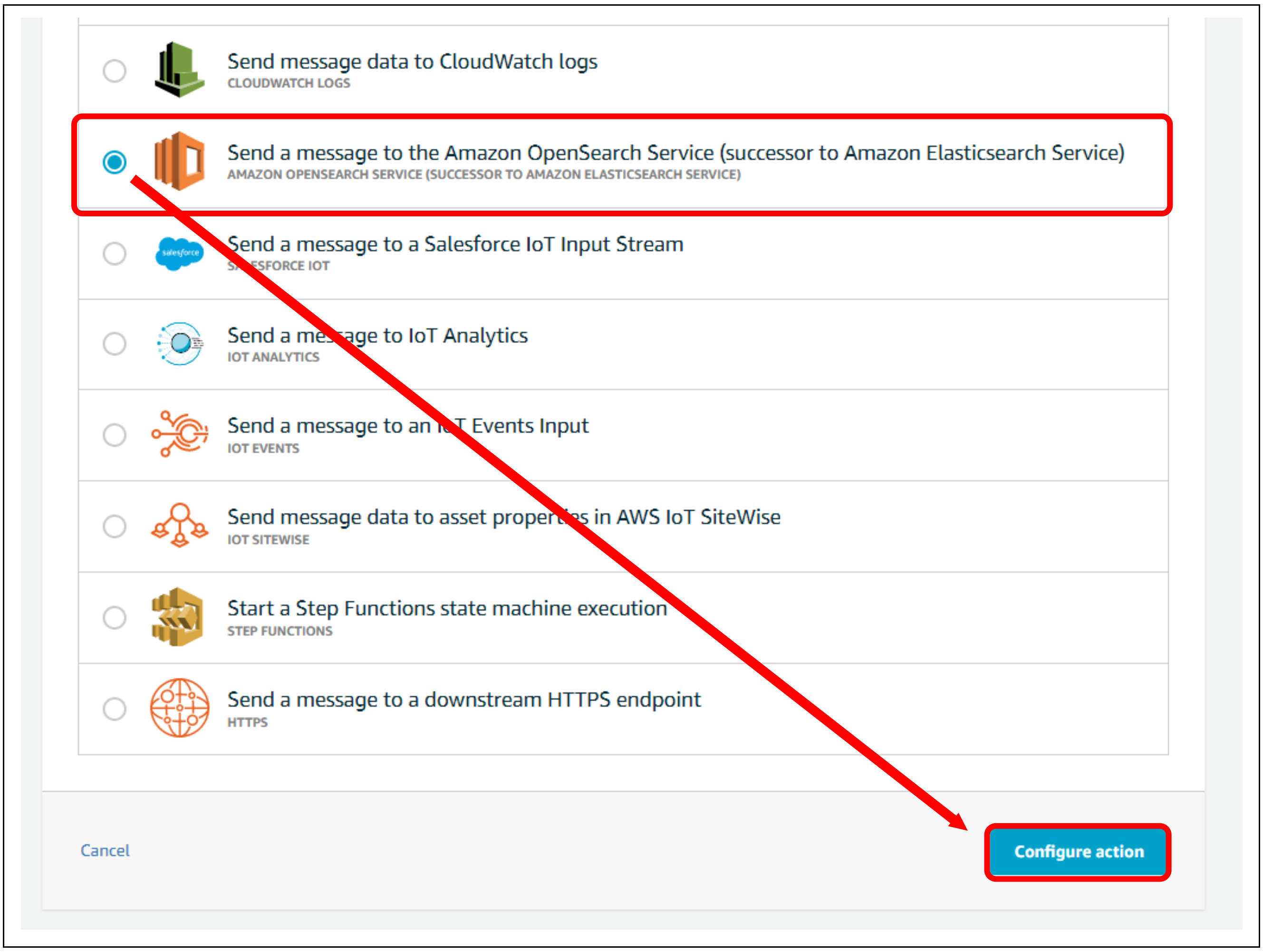

Figure 3-29 AWS IoT/Rules/Create a Rule: Action Selection Screen

7. AWS IoT/Rules: **Domain name, ID, Index** and **Type** of **Configure Action**

The Configure action screen shown in Figure 3-30 opens.
Set ***Domain name** to the domain name created in **9.Amazon OpenSearch Service/Domains/(Created domain name): Domain Creating** in 3.3.4(2)(a), AWS: Settings for Amazon OpenSearch Service.
Also, set each item as follows.

***ID : ${newuuid()}**
***Index : sensor**
***Type : sensor**

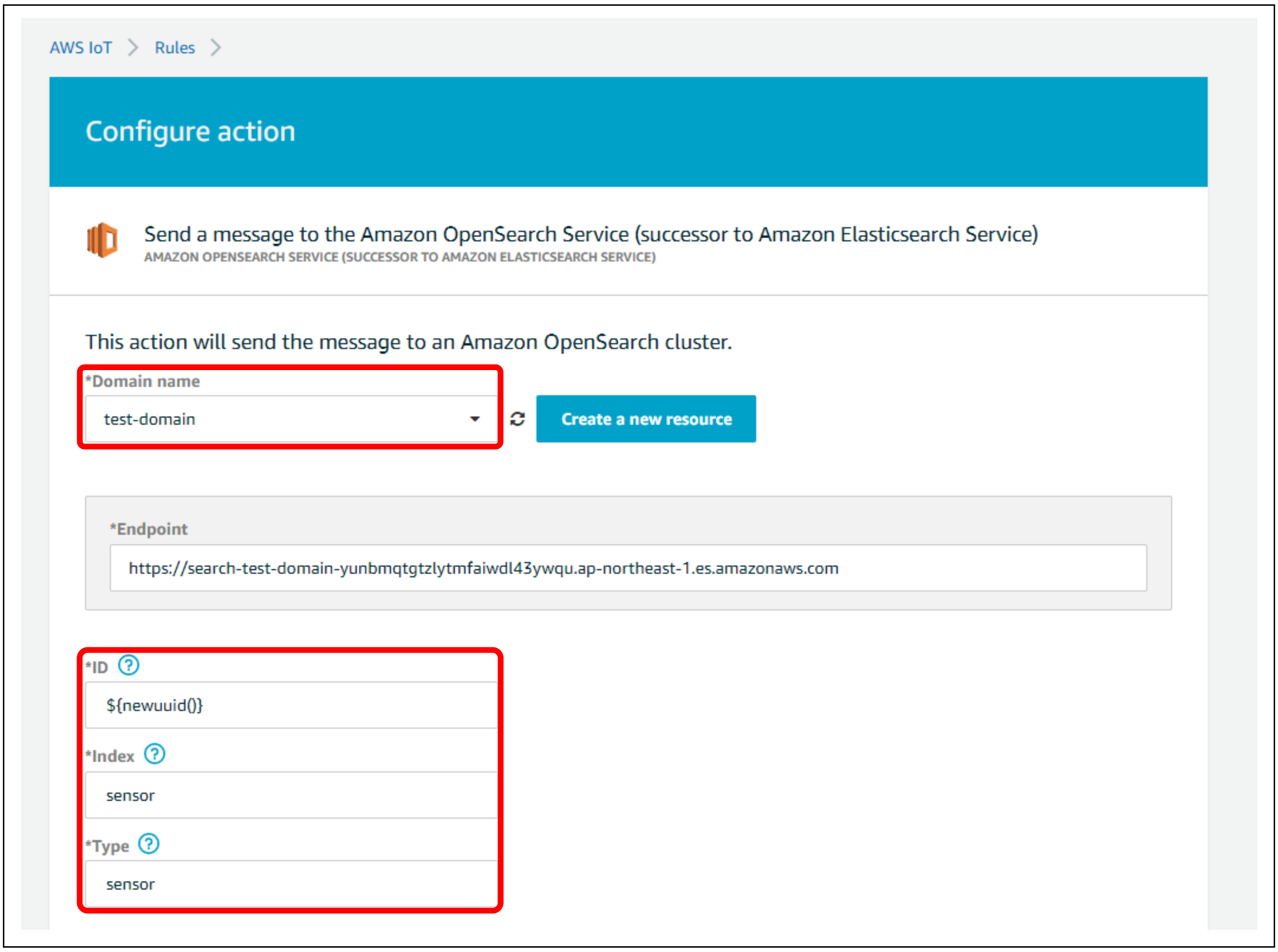

Figure 3-30 AWS IoT/Rules: Domain name, ID, Index and Type Screen of Configure Action Screen

8. AWS IoT/Rules: Role Creation/Selection of **Configure Action** [before Creating Role]

Create a role.
Click **Create Role**. Then, it will transition to the role creation processing.

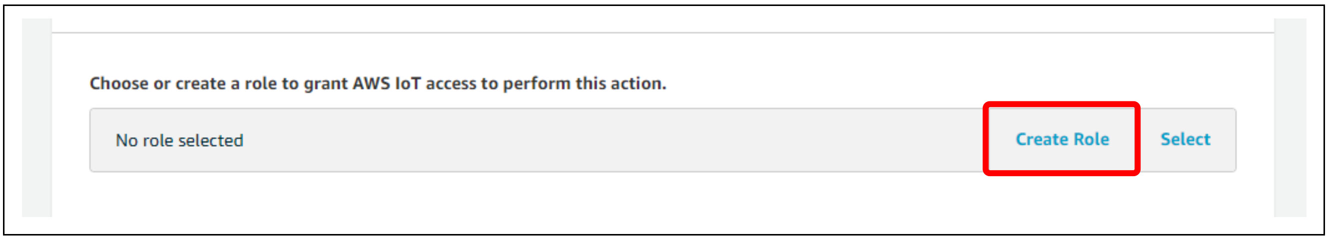

Figure 3-31 AWS IoT/Rules: Role Creation/Selection Screen of Configure Action [before Creating Role]

9. AWS IoT/Rules: **Name** of **Configure a new role**

Set **Name** to a new role name and click **Create Role**.
After the role creation is completed, it will return to the role creation/selection screen of the **Configure Action**, as shown in Figure 3-33.
Figure 3-32 is an example of creating the name **test_role**.

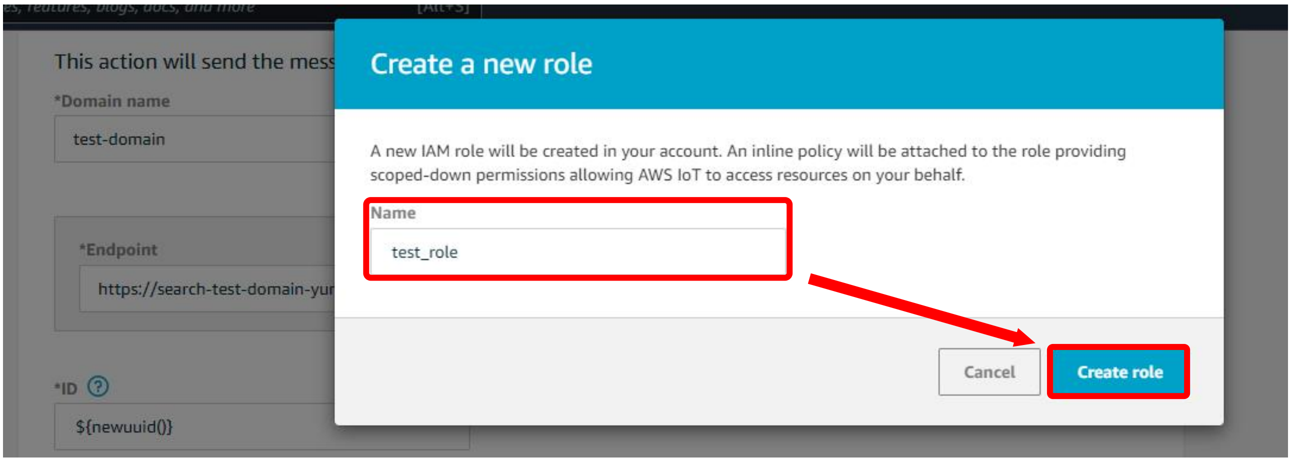

Figure 3-32 AWS IoT/Rules: Role Name Setting

10. AWS IoT/Rules: Role Creation/Selection of **Configure Action** [after Creating Role]

Confirm that the created role has been added.
Click **Add action**.
After the addition is completed, it will return to the **Set one or more actions** screen, as shown in Figure 3-34.

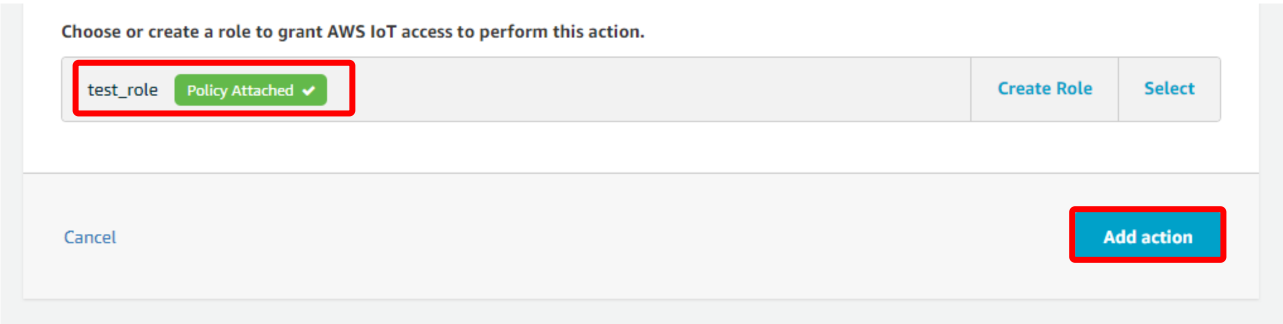

Figure 3-33 AWS IoT/Rules: Role Creation/Selection Screen of Configure Action [after Creating Role]

11. AWS IoT/Rules/Create a rule: **Set one or more actions** [after Adding Action]

The **Set one or more actions** screen is displayed again in order to add the action.
Confirm that the action has been added and click **Create Rule**.

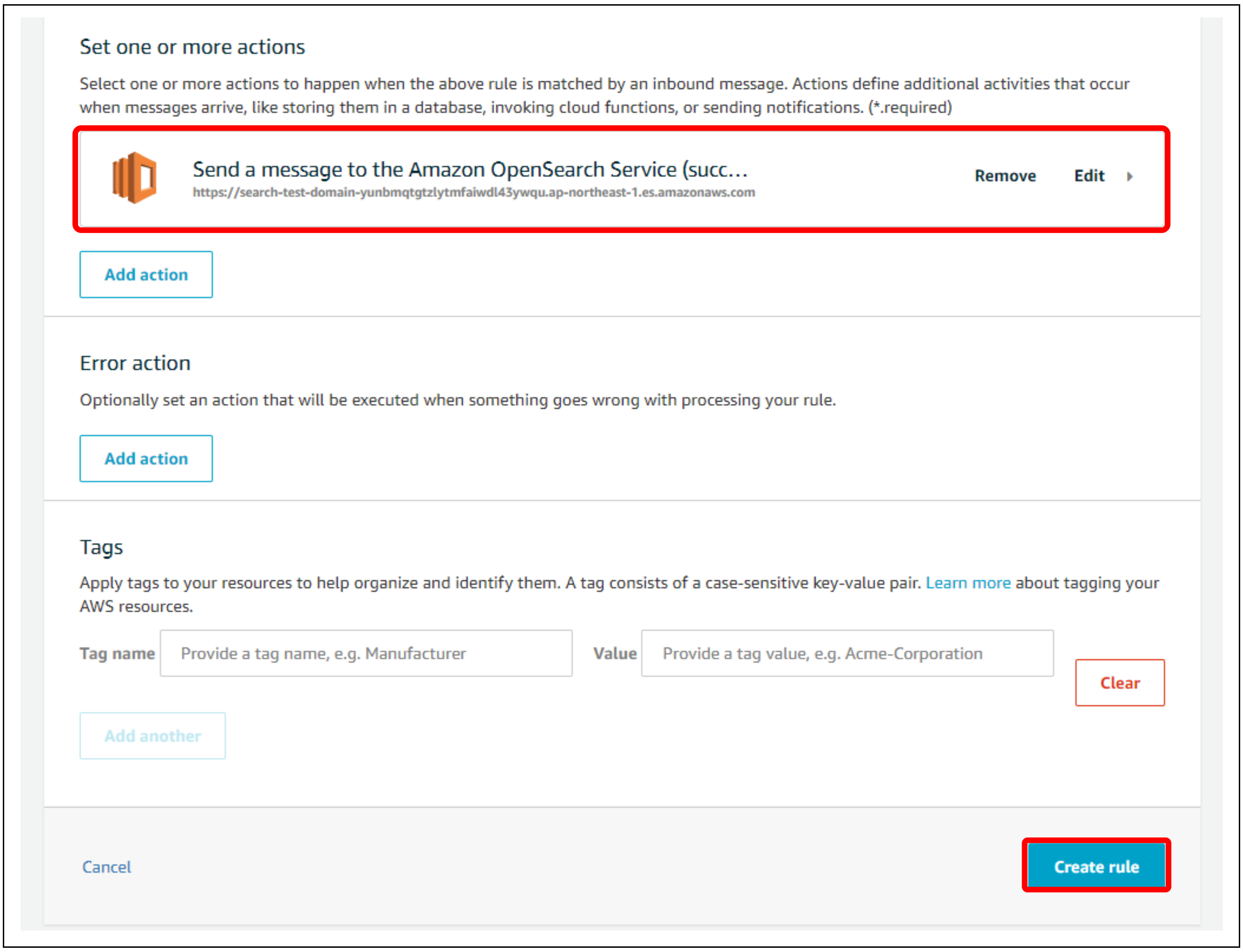

Figure 3-34 AWS IoT/Rules/Create a Rule: Set One or More Actions Screen [after Adding Action]

(d) OpenSearch Dashboards: Data Browsing

The procedure is shown below.

1. After accessing the URL of OpenSearch Dashboards, click **Explore on my own** on the displayed screen.

   For the URL, refer to 3.3.4(2)(a)10, Amazon OpenSearch Service/Domains/(Created domain name): Domain Creation Completed.

2. Select “ ≡ ” → **Stack Management**

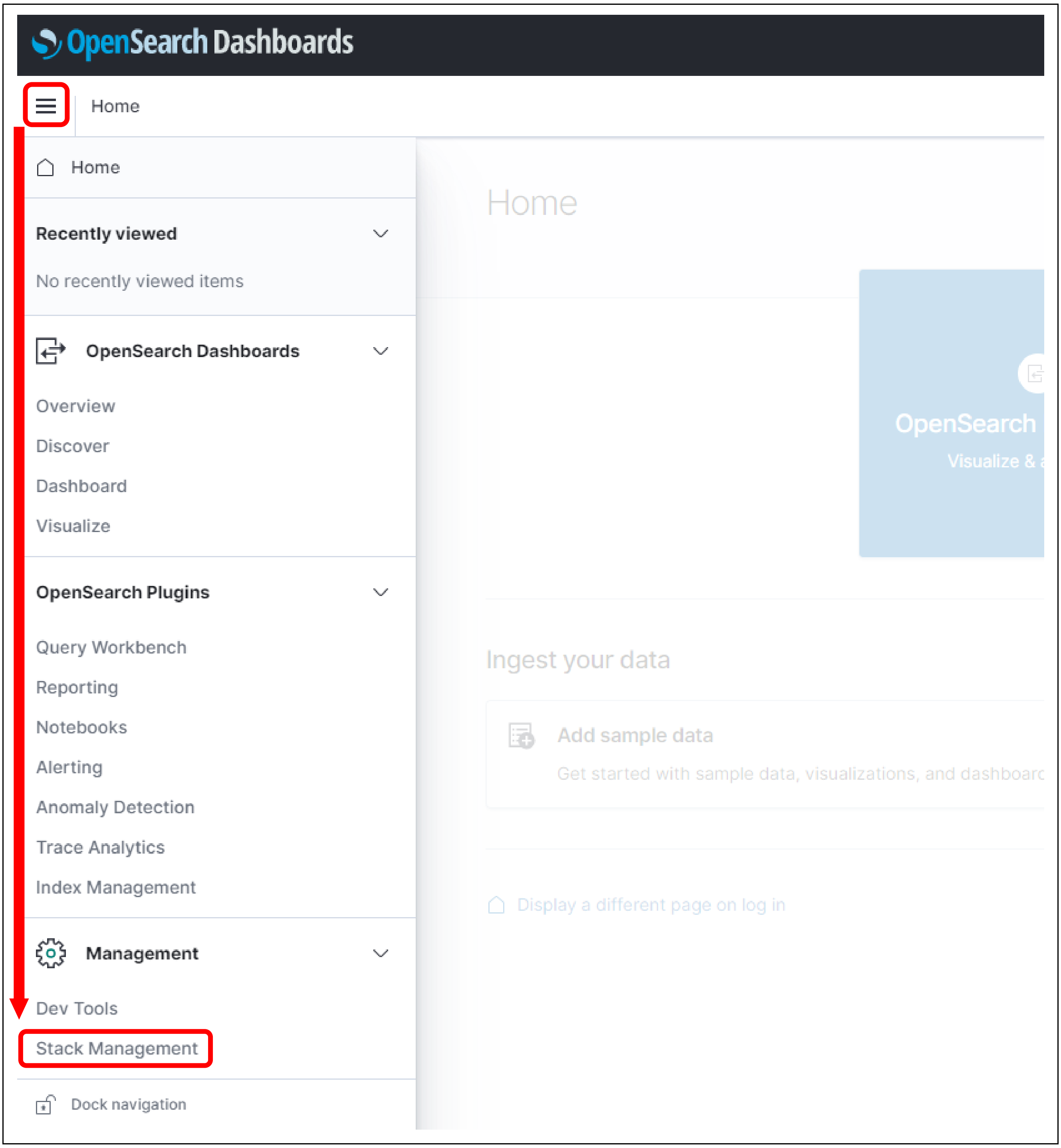

Figure 3-35 Screen of OpenSearch Dashboards

3. Stack Management/Index patterns:

   Select **Index Patterns** and click **Create index pattern**.

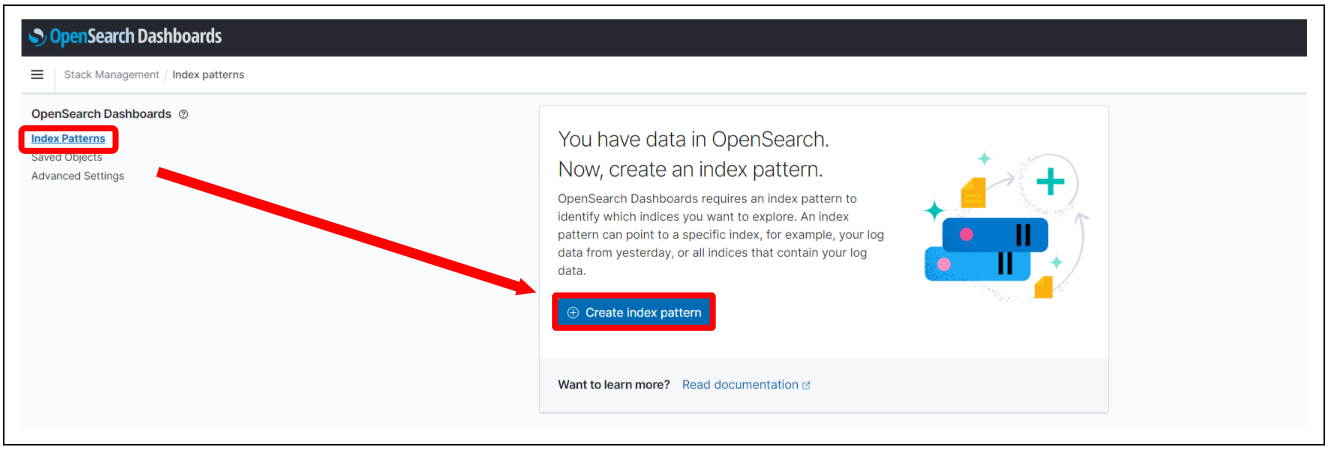

Figure 3-36 Stack Management/Index patterns: Index Patterns Screen

4. Stack Management/Index Patterns/Create index pattern: Step 1 of 2: Define an index pattern

   Set **Index pattern name** to sensor and click **Next step**.

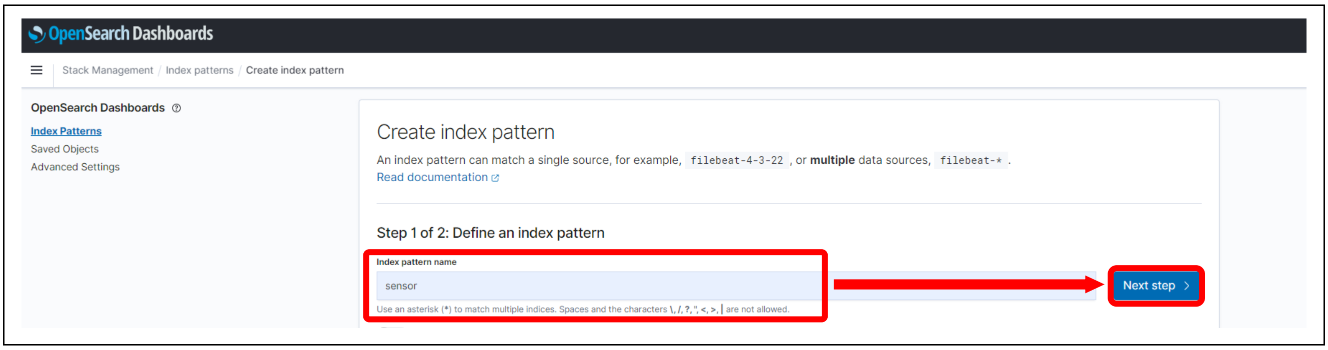

Figure 3-37 Stack Management/Index patterns/Create index pattern: Input Screen of Index Pattern Name

5. Stack Management/Index Patterns/Create index pattern: Step 2 of 2: Configure settings

   Set **Time Field** to **datetime** and click **Create index pattern**.

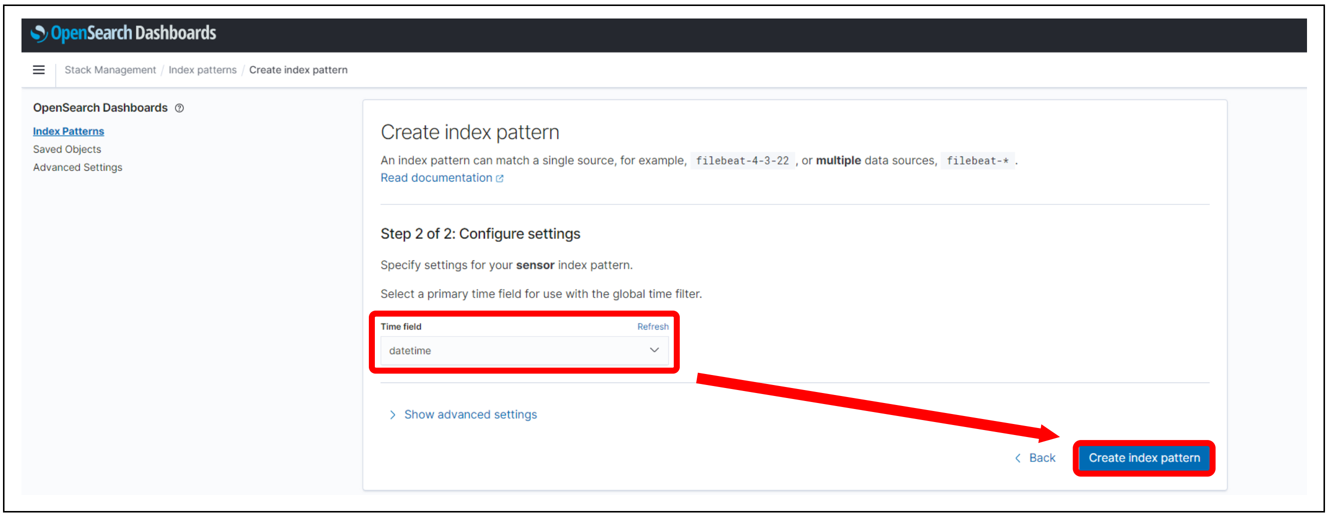

Figure 3-38 Stack Management/Index Patterns/Create index pattern: Time Field Setting Screen

6. Select " ≡ " →**Visualize**

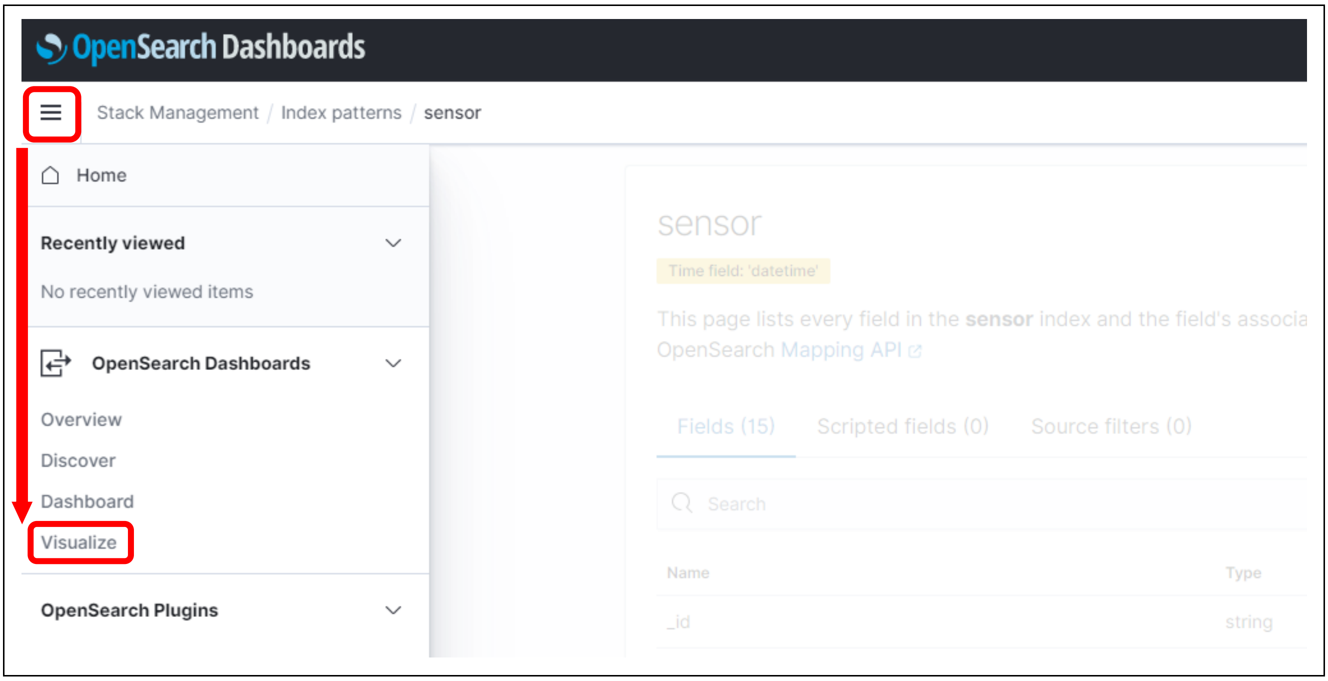

Figure 3-39 Visualize Selection Screen of OpenSearch Dashboards

7. Visualize:

   Click **Create new visualization**.

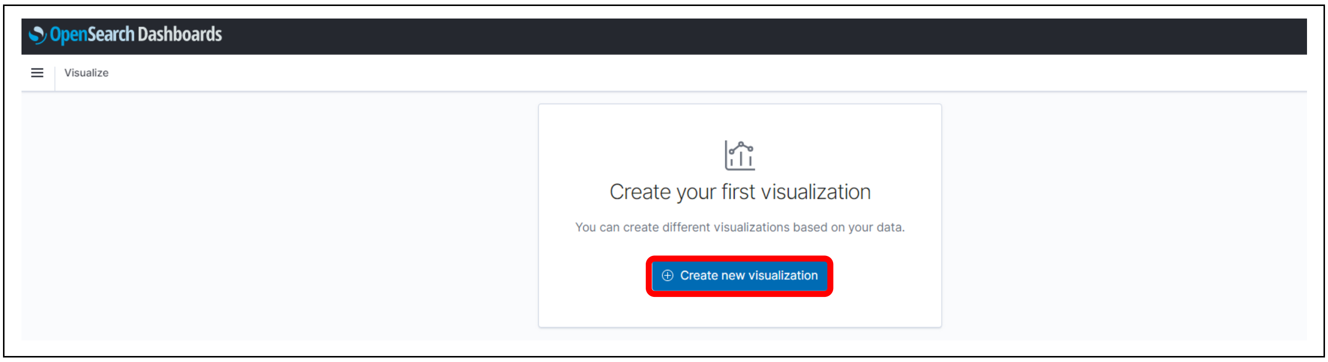

Figure 3-40 Visualize: Create Visualization Selection Screen

8. Visualize: New Visualization

Select **Line**.

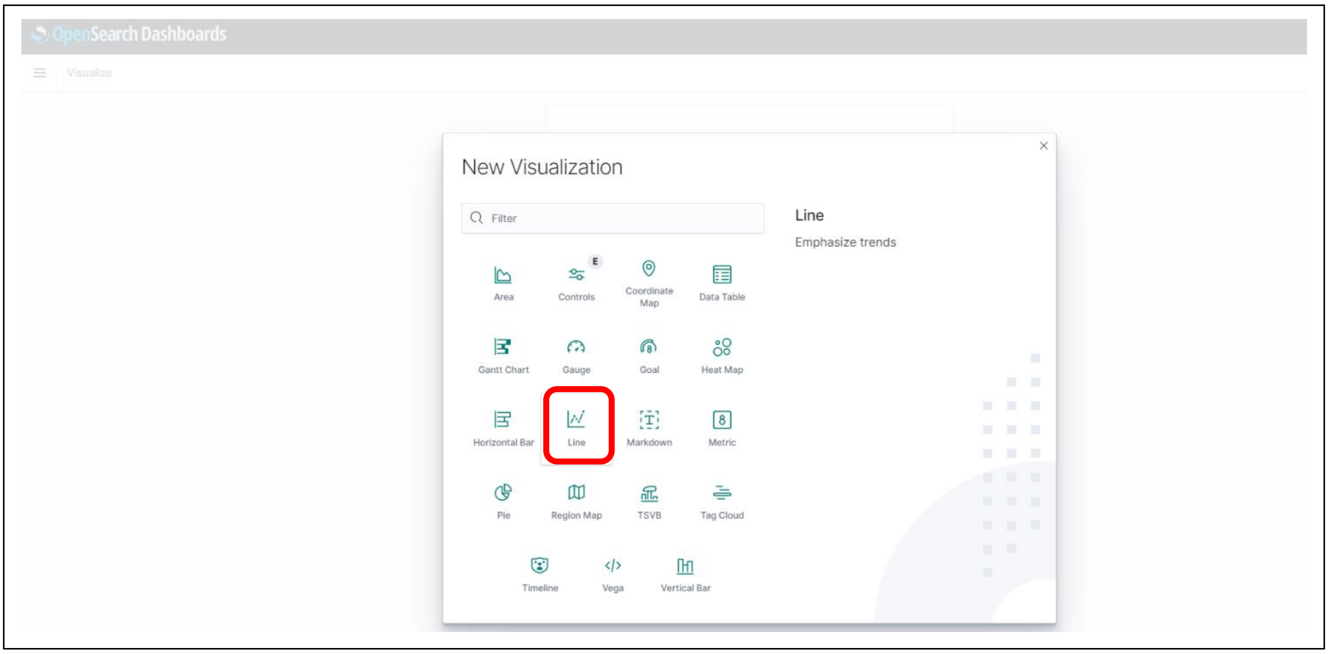

Figure 3-41 Visualize: New Visualization Screen

9. Visualize: New Line/Choose a source

Search for the **Index pattern name** set in 3.3.4(2)(d), 4. Stack Management/Index Patterns/Create index pattern: Step 1 of 2: Define an index pattern.
Select it.

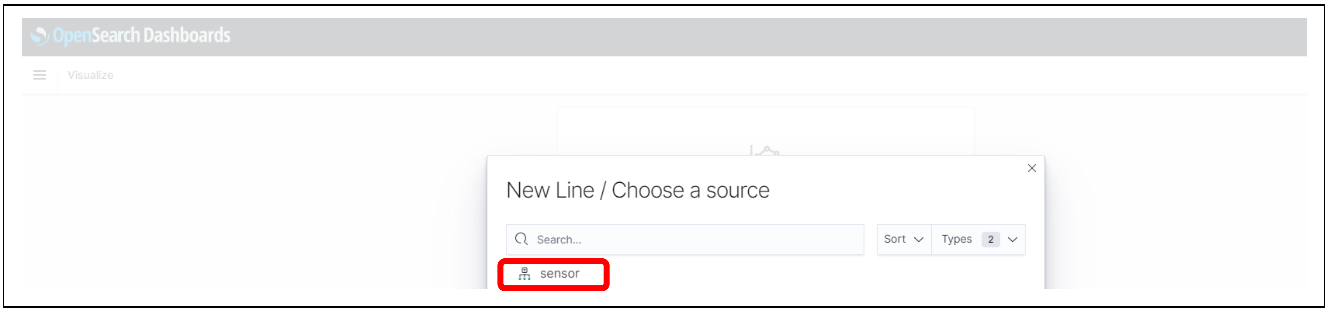

Figure 3-42 Visualize: Index pattern name of New Line/Choose a source

10. Visualize/Create: Refresh Every

Select the **calendar icon (📅)**, set **Refresh every** and click **Start**.
The graph will be updated at the set time.

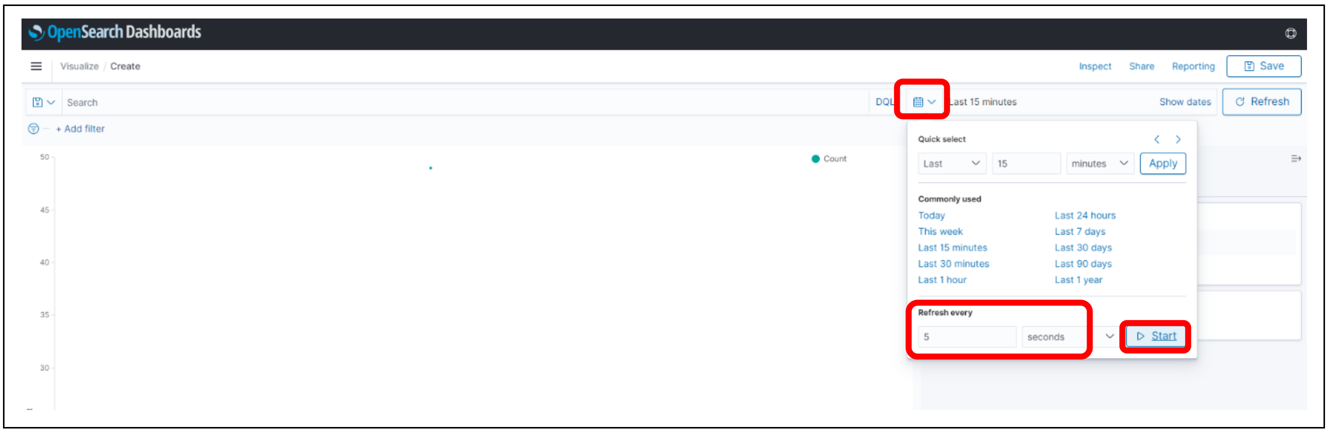

Figure 3-43 Visualize/Create: Refresh Every Time Setting Screen

11. Visualize/Create: Metrics (Aggregation, Field) of Data tab

Select the **Data** tab and set the contents of the **Y-axis** as follows.

- **Aggregation : Average**
- **Field : hs3001_humidity**

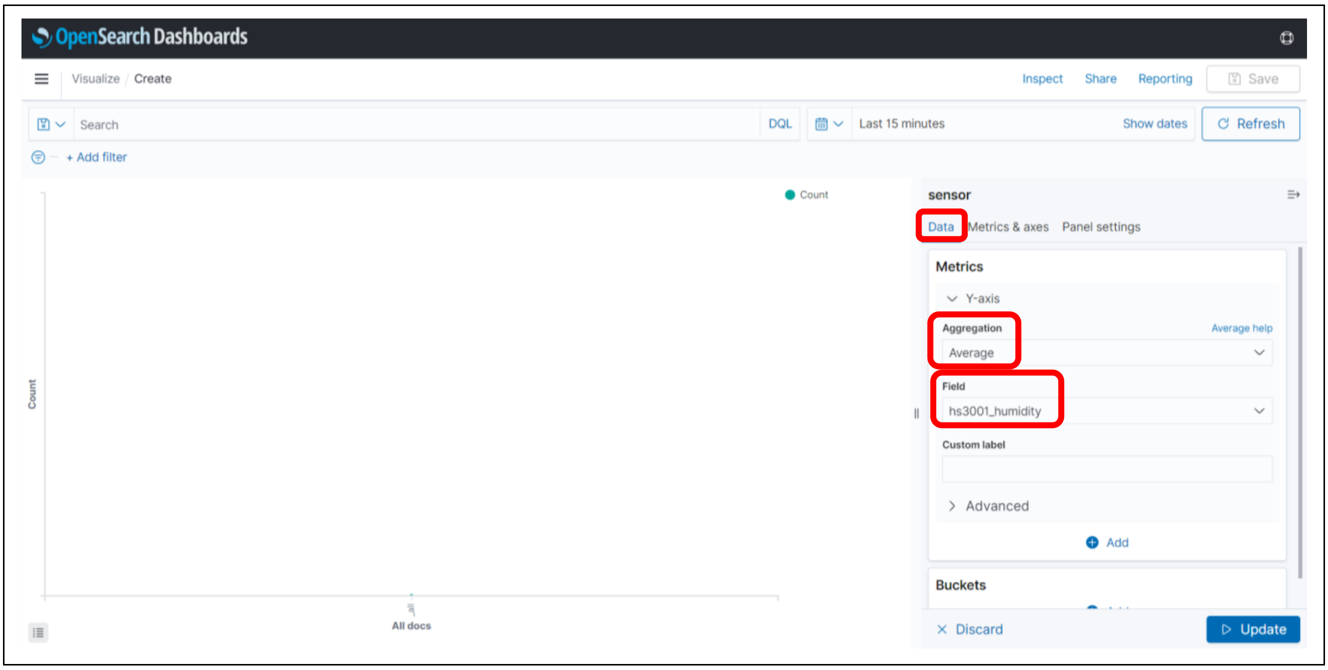

Figure 3-44 Visualize/Create: Metrics Setting Screen 1 of Data Tab

12. Visualize/Create: Metrics (Aggregation, Field) of Data tab

Click **Add** and Add the contents to Y-axis as follows. Set the content for each field.

- **Aggregation** : **Average**
- **Field** : **hs3001_temperature**
- : **zmod4410_iaq**
- : **zmod4410_eco2**
- : **zmod4410_tvoc**
- : **zmod4410_odor**
- : **zmod4410_accpeptable**
- : **zmod4410_intensity**

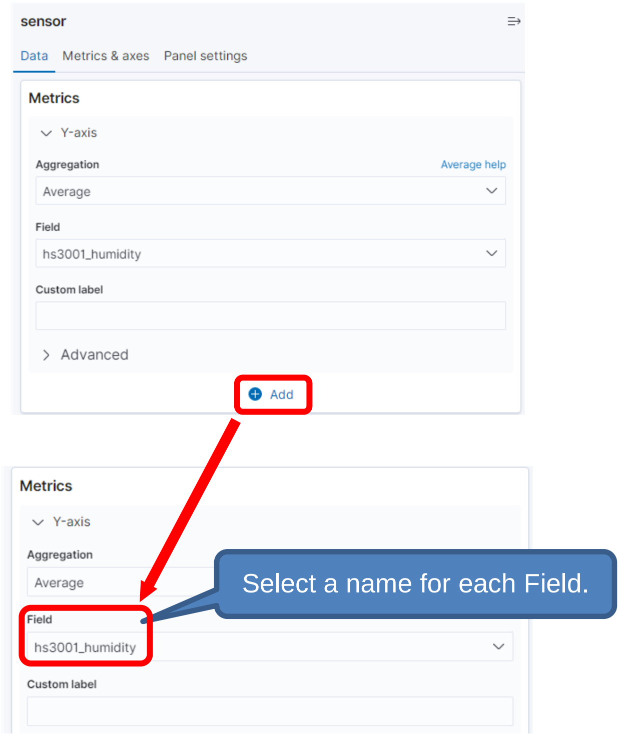

Figure 3-45 Visualize/Create: Metrics Setting Screen 2 of Data Tab

13. Visualize/Create: Buckets of Data tab

Click **Add** and select **X-axis**.

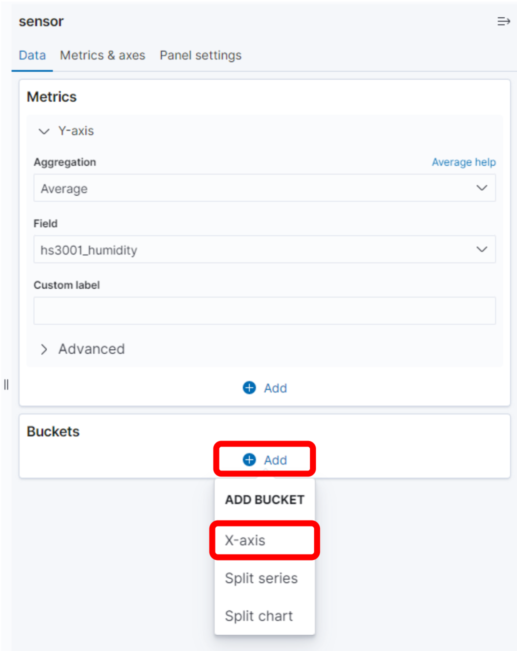

Figure 3-46 Visualize/Create: Buckets Addition Screen of Data Tab

14. Visualize/Create: Buckets of Data tab

Set the contents of the **X-axis** as follows.

- **Aggregation : Date Histogram**
- **Field : datetime**
- **Minimum interval : Second**

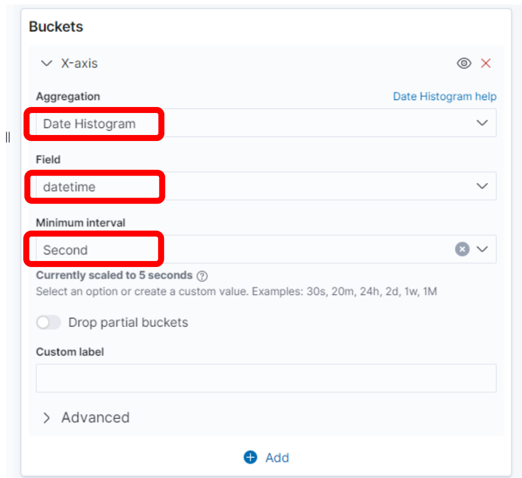

Figure 3-47 Visualize/Create: Buckets Setting Screen of Data Tab

15. Visualize/Create: Update

Click Update. Then, the settings of the Y axis and X axis so far will be reflected.

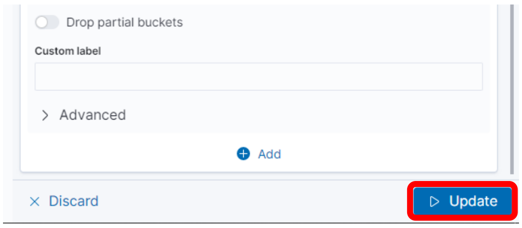

Figure 3-48 Visualize/Create: Update Setting Screen

16. Visualize/Create:

The data is plotted.

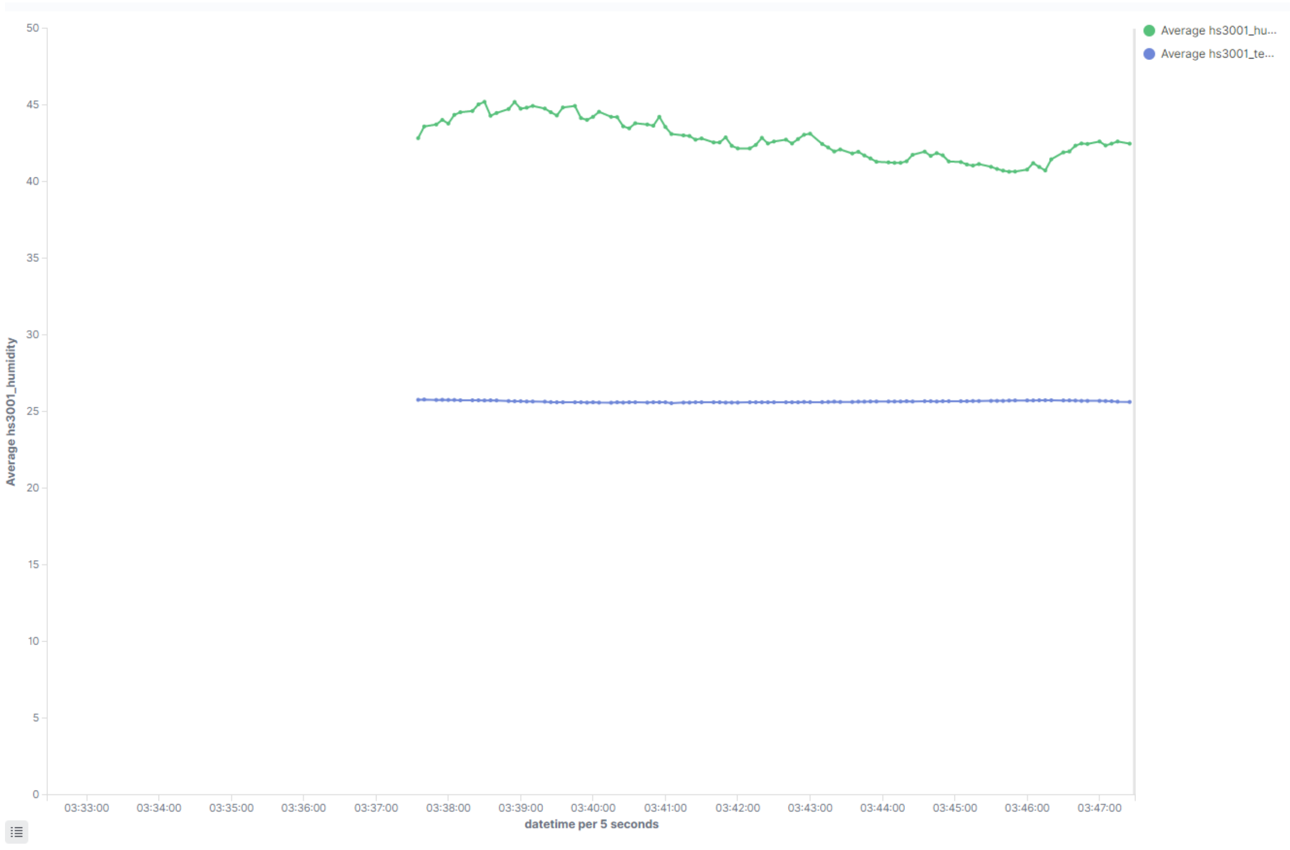

Figure 3-49 Visualize/Create: Data Plotting

3.4 Running the Demo

The demo consists of the following three stages.

1. Uploading to cloud and visualization of HS3001 sensor data
2. Execution of 2nd OTA update on TB-RX23W board
3. Uploading to cloud and visualization of HS3001 sensor data and ZMOD4410 sensor data

The detailed steps to actually run the demo are described below.

3.4.1 Uploading to Cloud and Visualization of HS3001 Sensor Data

Set up the initial state of the demo (the initial state before 2nd OTA update).

The procedure is as follows.

(1) Step 1: Program Initial Firmware to Each Board, Configure PC Settings, and Configure AWS Settings

Refer to 3.3, Software Environment.

(2) Step 2: Make Connections between Boards and PC

Refer to 3.2, Hardware Environment.

(3) Step 3: Use Terminal Emulator Software to Check Log Output

Refer to 3.3.3, Terminal Emulator Software Settings.

Figure 3-50 shows a log window of output from the RSK+RX65N-2MB board. It can be confirmed that the data of the HS3001 sensor is output. Below that, you can check the log of sending sensor data to AWS by MQTT communication.

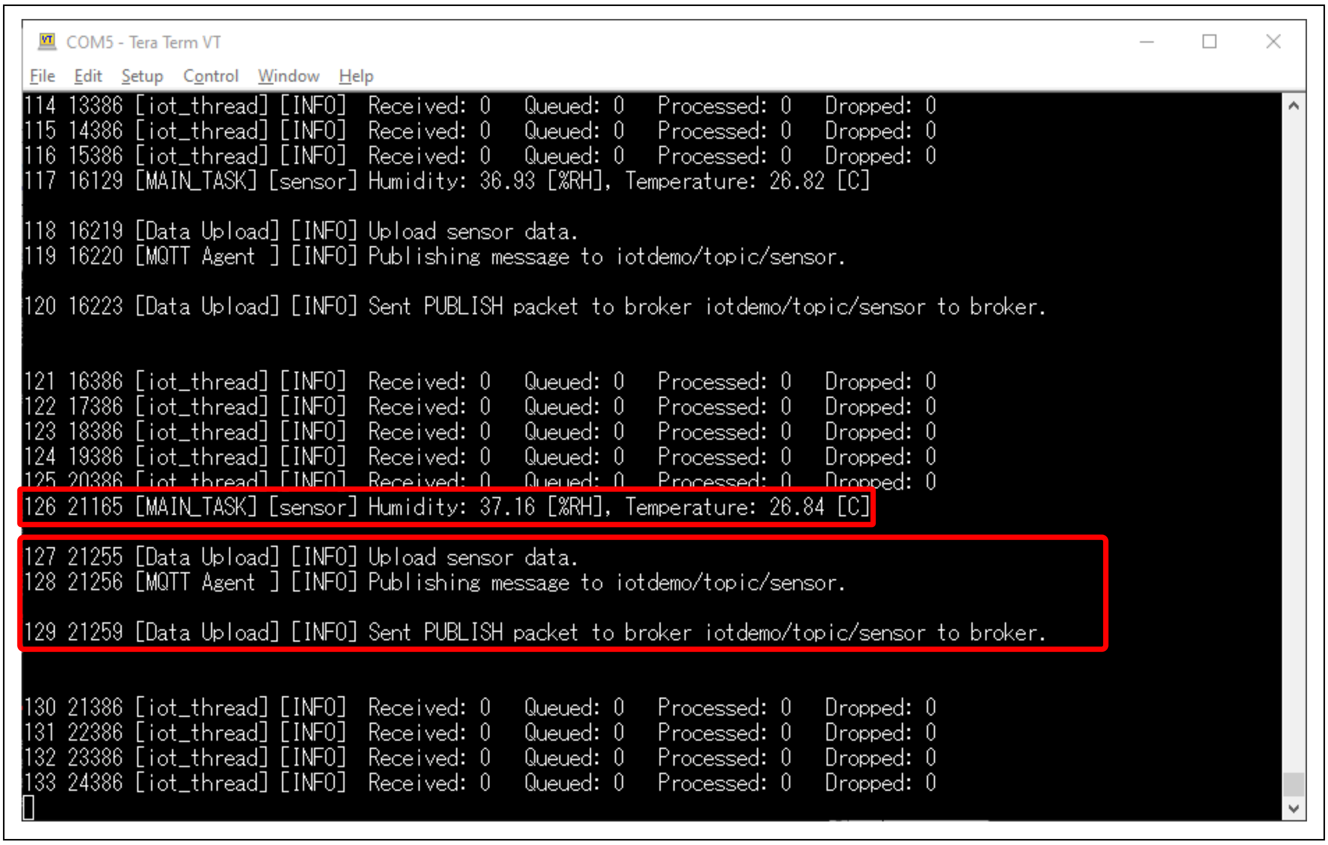

Figure 3-50 RSK+RX65N-2MB Board Log Window

Figure 3-51 shows the initial log of output from the TB-RX23W board.

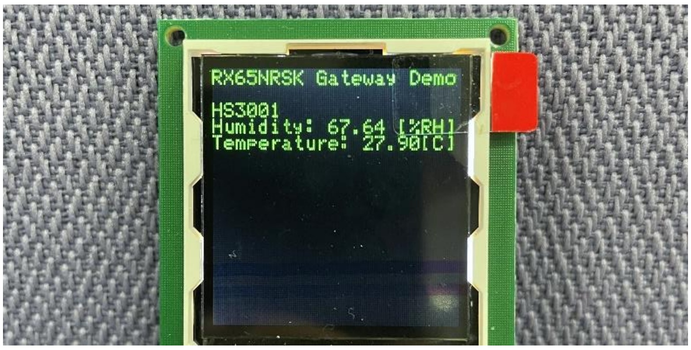

Figure 3-51 Initial Screen Display of Pmod LCD

Figure 3-52 shows the log screen of the TB-RX23W board. Only HS3001 sensor data is displayed.

Confirm that LED1 is blinking and LED0 is off.

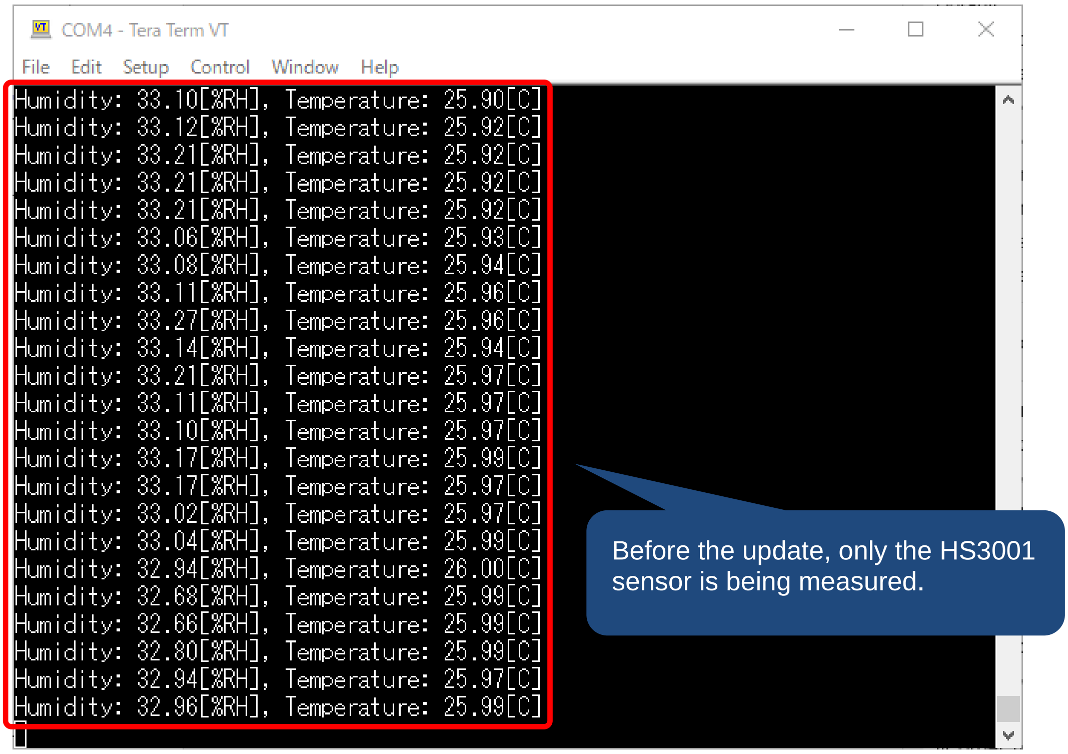

Figure 3-52 TB-RX23W Board Log Window

(4) Step 4: Confirm that Sensor Data Is Plotted on Graph in AWS OpenSearch Dashboards

Confirm that only the HS3001 sensor data is plotted on the graph, as shown in Figure 3-53.

This is the initial state before 2nd OTA update.

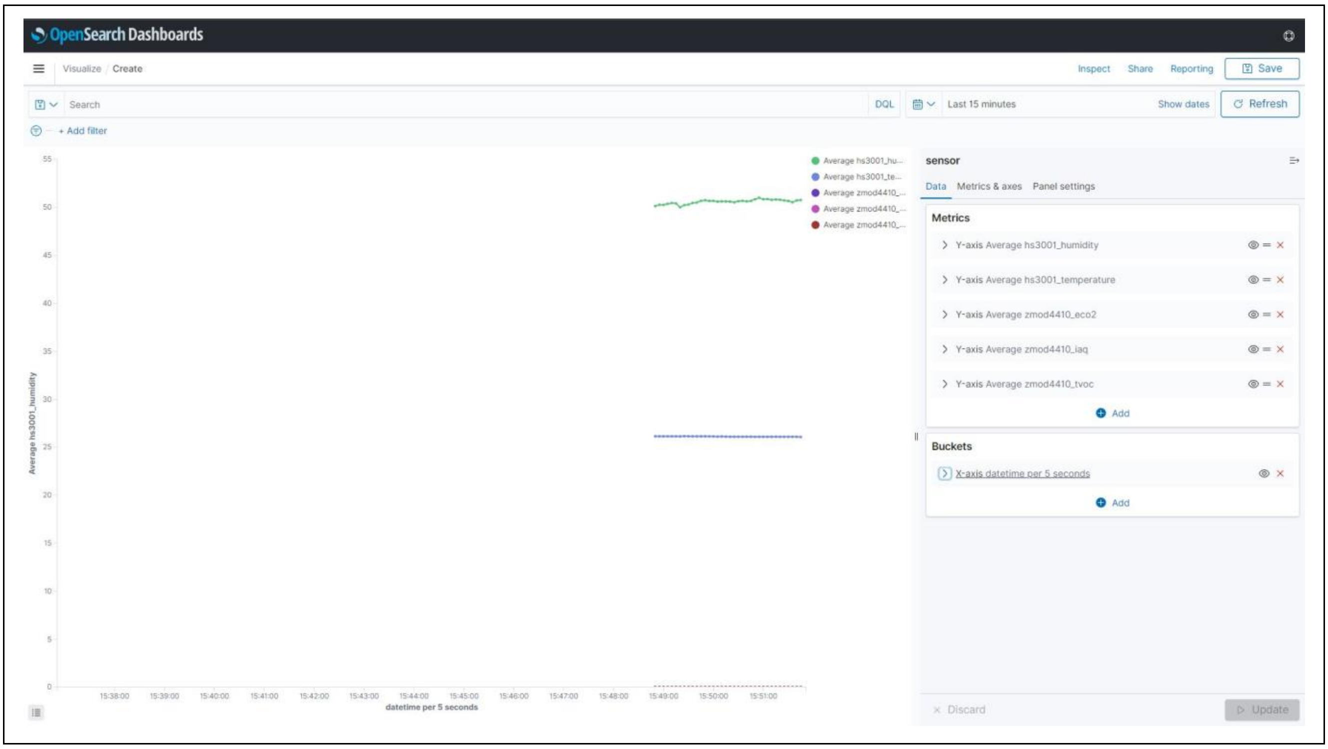

Figure 3-53 OpenSearch Dashboards Graph Display before 2nd OTA Update

3.4.2 Execution of 2nd OTA Update of Firmware on TB-RX23W Board

The procedure to perform 2nd OTA update to update the firmware to enable the TB-RX23W board to acquire measurement data from the ZMOD4410 sensor in addition to the HS3001 sensor is described below.

The procedure is described below.

(1) Step 1: Generate Firmware for Update

Change the following macro setting from "0" to "1" and build the project.

54th line of rx23w_hs3000_demo_project/src/main.c

```
#define USE_ZMOD4XXX                                    (1)
```

(2) Step 2: Generate Firmware for Update using Renesas Secure Flash Programmer

Launch **Renesas Secure Flash Programmer.exe** and select the **Update Firm** tab. Then enter the settings listed in Table 3-6 and click **Generate**.

Table 3-6 Renesas Secure Flash Programmer Settings for Generating Firmware for Updating TB-RX23W Board

Select MCU	RX23W(ROM 512KB)/Secure Bootloader=64KB
Select Firmware Verification Type	sig-sha256-ecdsa
Private Key Path (PEM Format)	Path to private key generated previously
Bank0 User program Firmware Sequence Number	1
Bank0 File Path (Motorola Format)	Path to MOT file of the previously generated firmware for updating the TB-RX23W board

When the message **Generate succeeded** appears in the yellow frame below in the Renesas Secure Flash Programmer, the file **userprog.rsu** have been concatenated to generate.

This completes the procedure for generating the firmware for updating the TB-RX23W board.

(3) Step 3: Create OTA job to execute 2nd OTA Update

(a) Sign in to the AWS Management Console and select Services Internet of Things IoT Core in the upper left corner

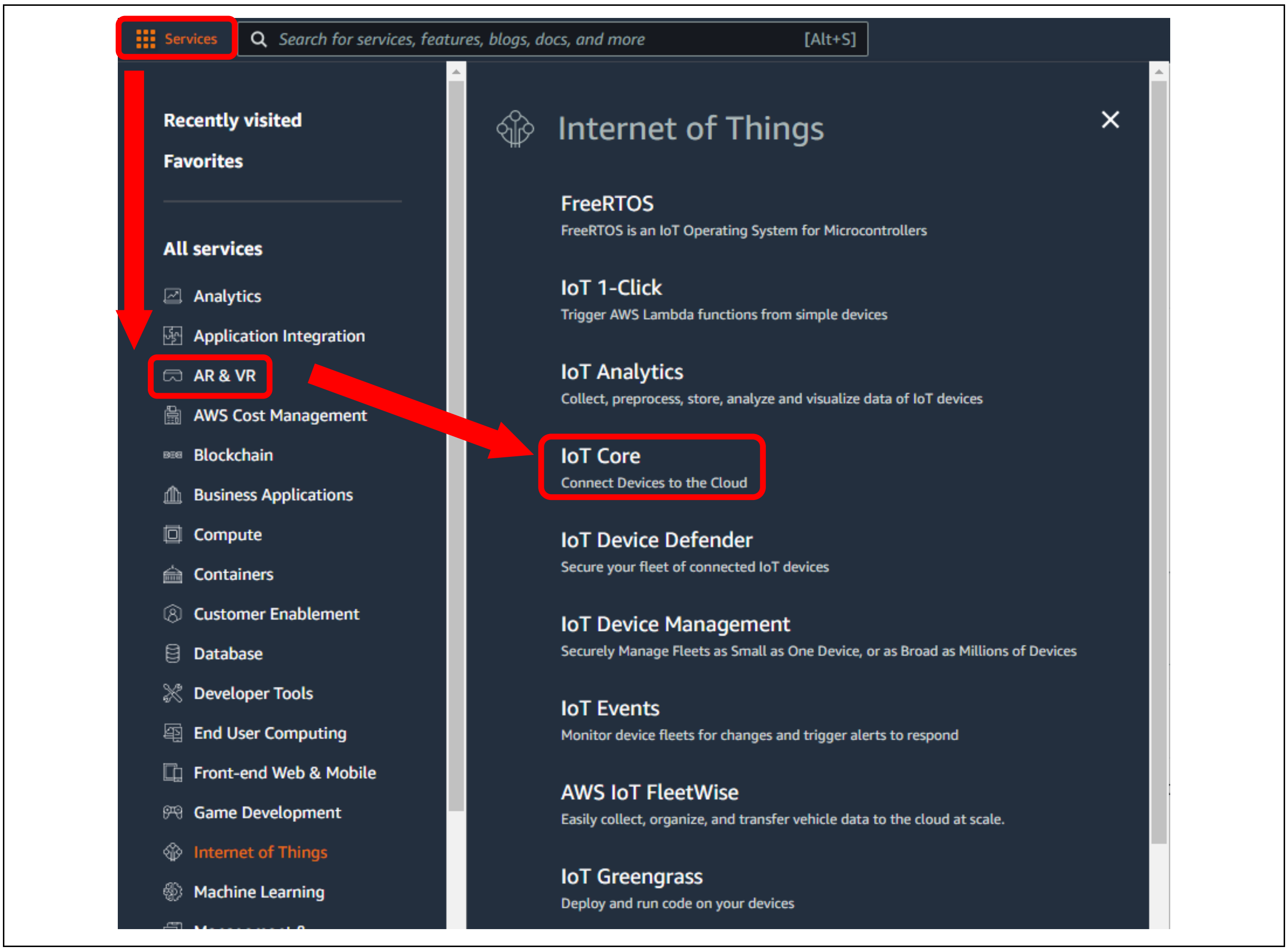

Figure 3-54 AWS services Screen

(b) From the left menu of AWS IoT Core, select Manage → Jobs and click "Create job"

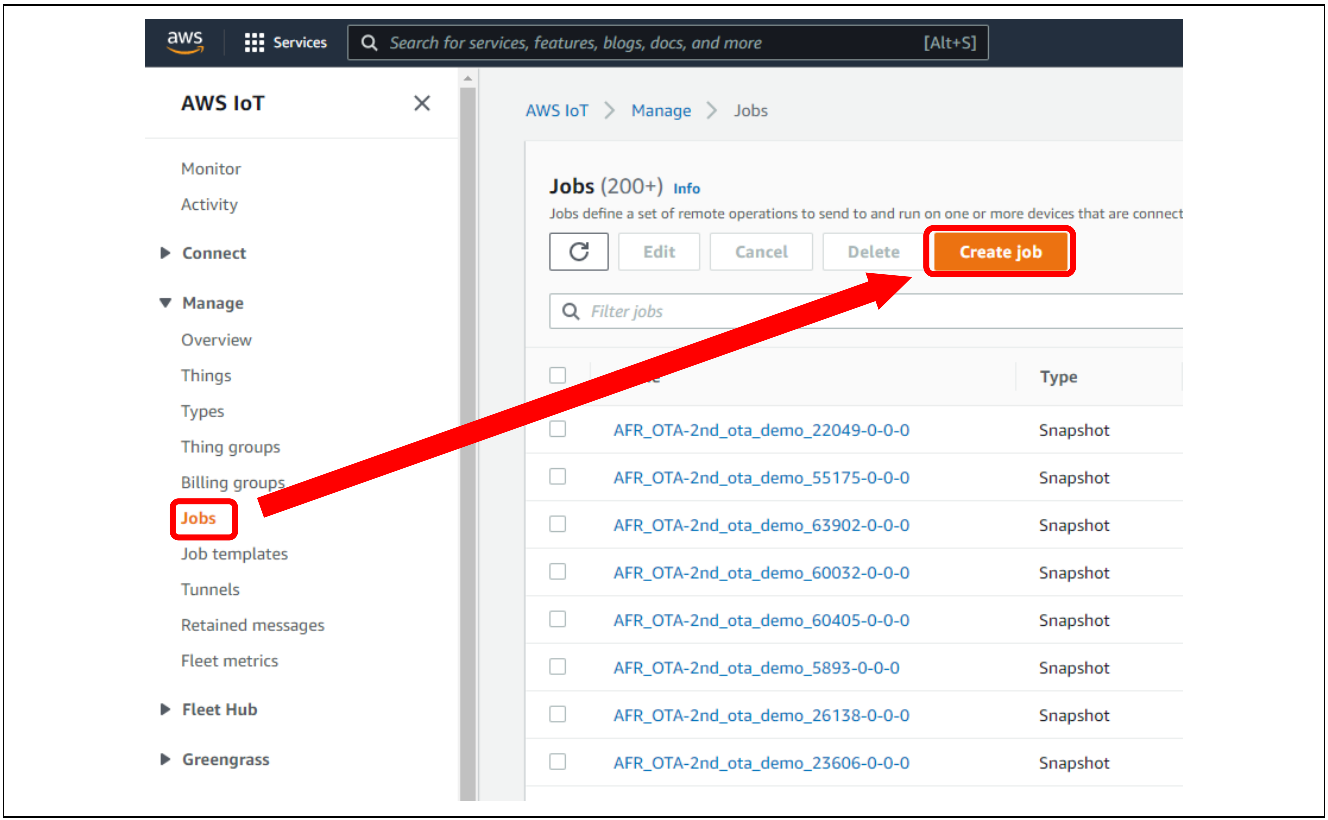

Figure 3-55 AWS IoT Core Screen

(c) On the "Create job" screen, select "Create FreeRTOS OTA update job" and click "Next"

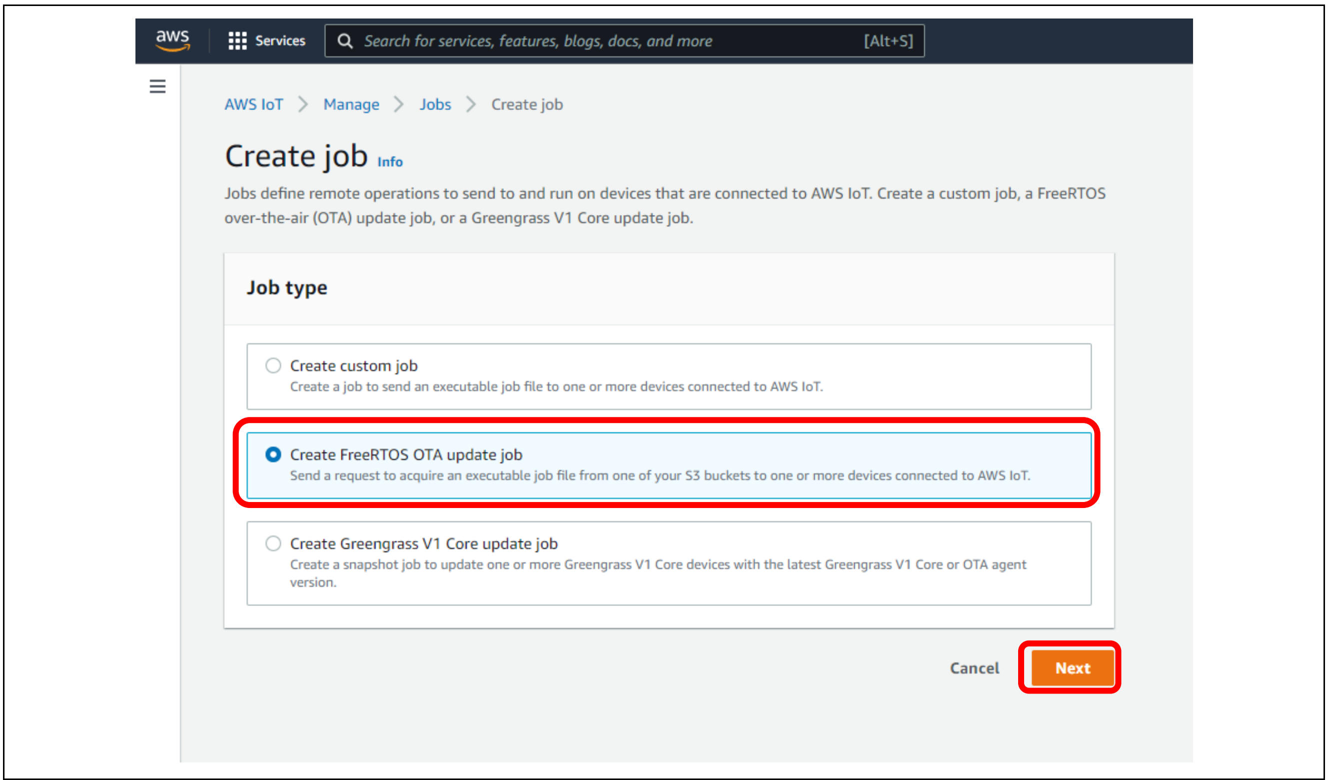

Figure 3-56 Create Job Screen

(d) On the "OTA job properties" screen, enter a "Job name" and click "Next"

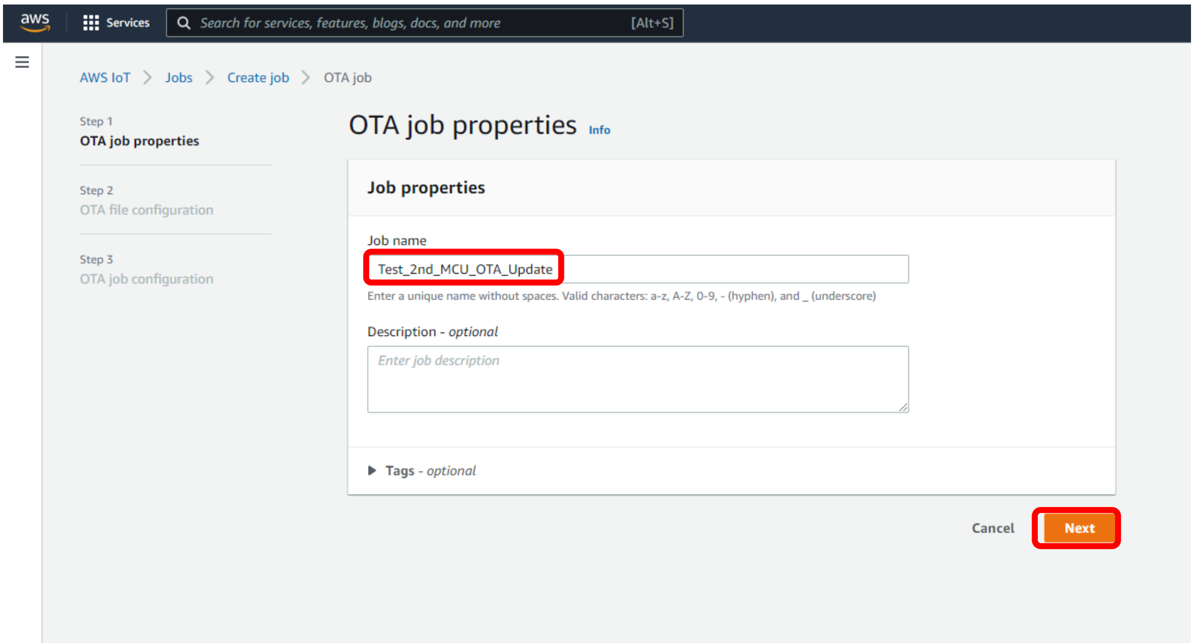

Figure 3-57 OTA Job Properties Entry Screen

(e) Enter the following items on the "OTA file configuration" screen

1. In the "Devices to update" field, enter the name of the thing set in 3.3.4(1).
2. In "Select the protocol for file transfer," select MQTT.
3. In "Sign and choose your file," select "Sign a new file for me."
4. In "Code signing profile," select the code signing profile set in 3.3.4(1).

   Note: To be precise, any profile can be specified for the code signing profile. The code signing profile specified here is not used for code signing verification of the firmware of the TB-RX23W board, because code signature is written in the file when the firmware for updating the TB-RX23W board is created by Renesas Secure Flash Programmer.
5. In "File," select "Upload a new file."
6. In "File to upload," click "Choose file" and select the firmware(.rsu format) for updating the TB-RX23W board created in 3.4.2(2).
7. In "S3 URL", click "Browse S3" and select the Amazon S3 bucket set in 3.3.4(1).
8. In "Path name of file on device," enter any string of characters.
9. In "File type," enter 1.
10. In "Role", select the service role for OTA update set in 3.3.4(1).

After entering the above, click "Next."

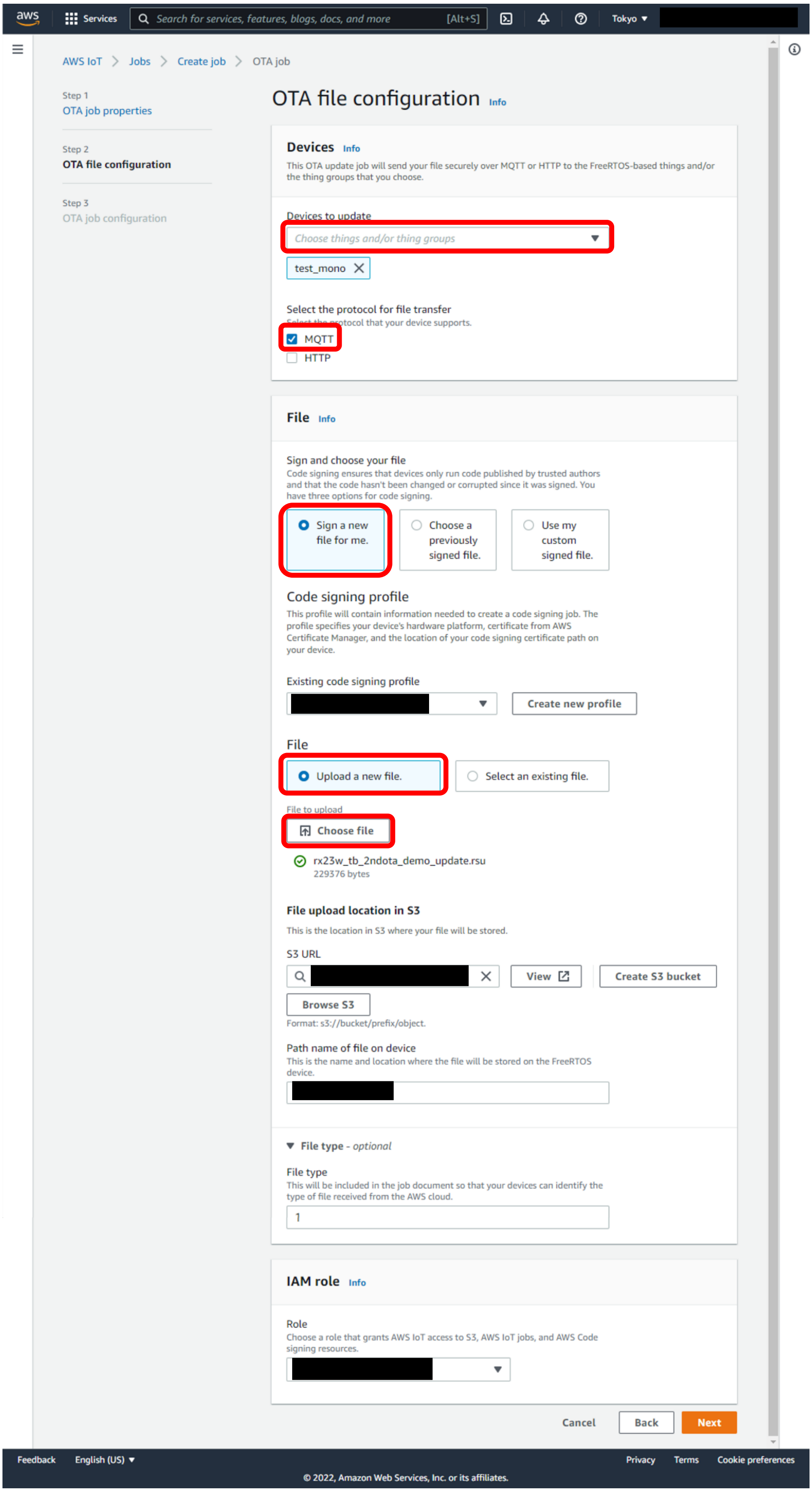

Figure 3-58 OTA File Configuration Screen

(f) Click "Create Job" on the "OTA job settings" screen as it is not necessary to make any changes.

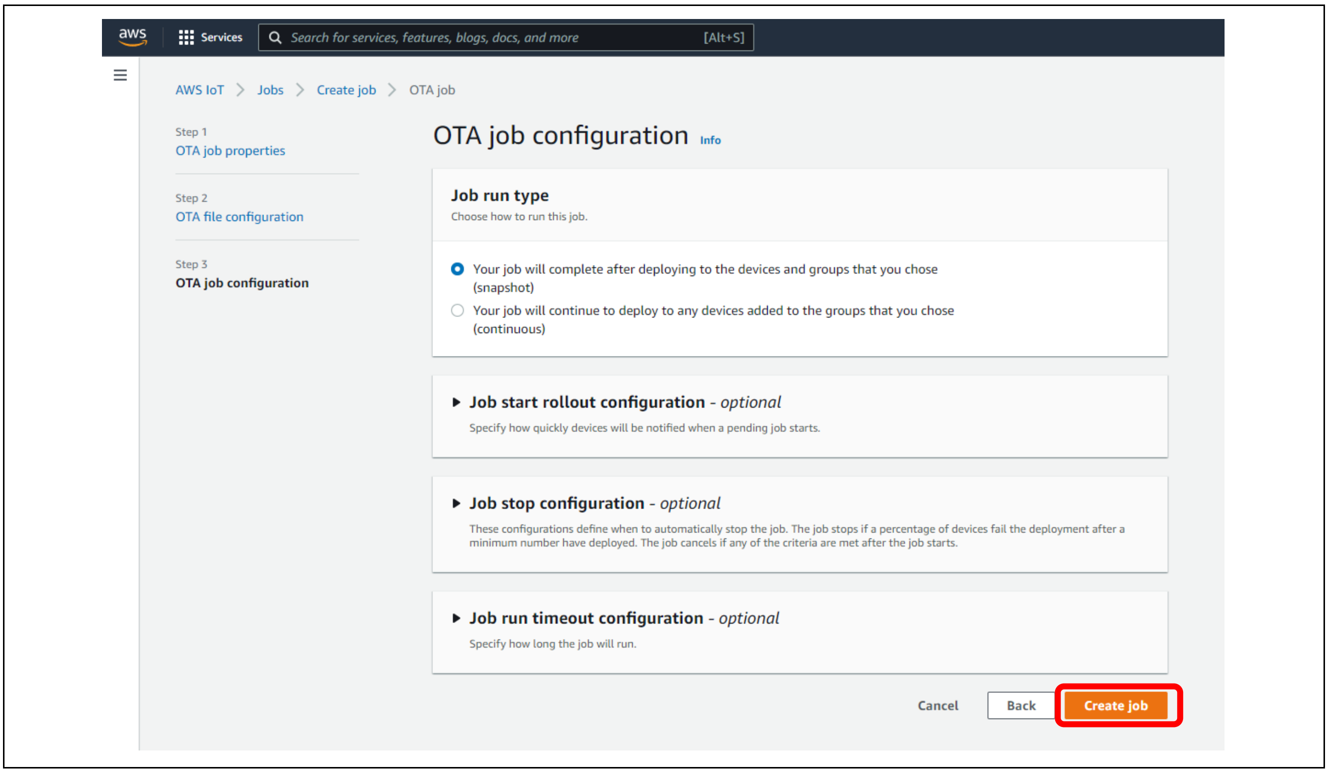

Figure 3-59 OTA Job Settings Screen

An OTA job is created for the 2nd OTA update and the OTA job is delivered to the specified device.

(4) Step 4: Confirm that 2nd OTA Update Is Complete

The 2nd OTA update begins a few seconds after the commands are executed. Progress logs are output to the serial consoles of the RSK+RX65N-2MB and TB-RX23W boards.

About 1 minute after the start of the OTA update, the TB-RX23W board will be reset once and all LEDs will stop blinking.

After another few seconds, it will start blinking again, and if the two LEDs (LED0 and LED1) are blinking simultaneously, the 2nd OTA update has completed successfully.

Figure 3-60 shows the log screen of the RSK+RX65N-2MB board after the update. You can see that the data for the ZMOD4410 sensor is displayed in addition to the HS3001 sensor.

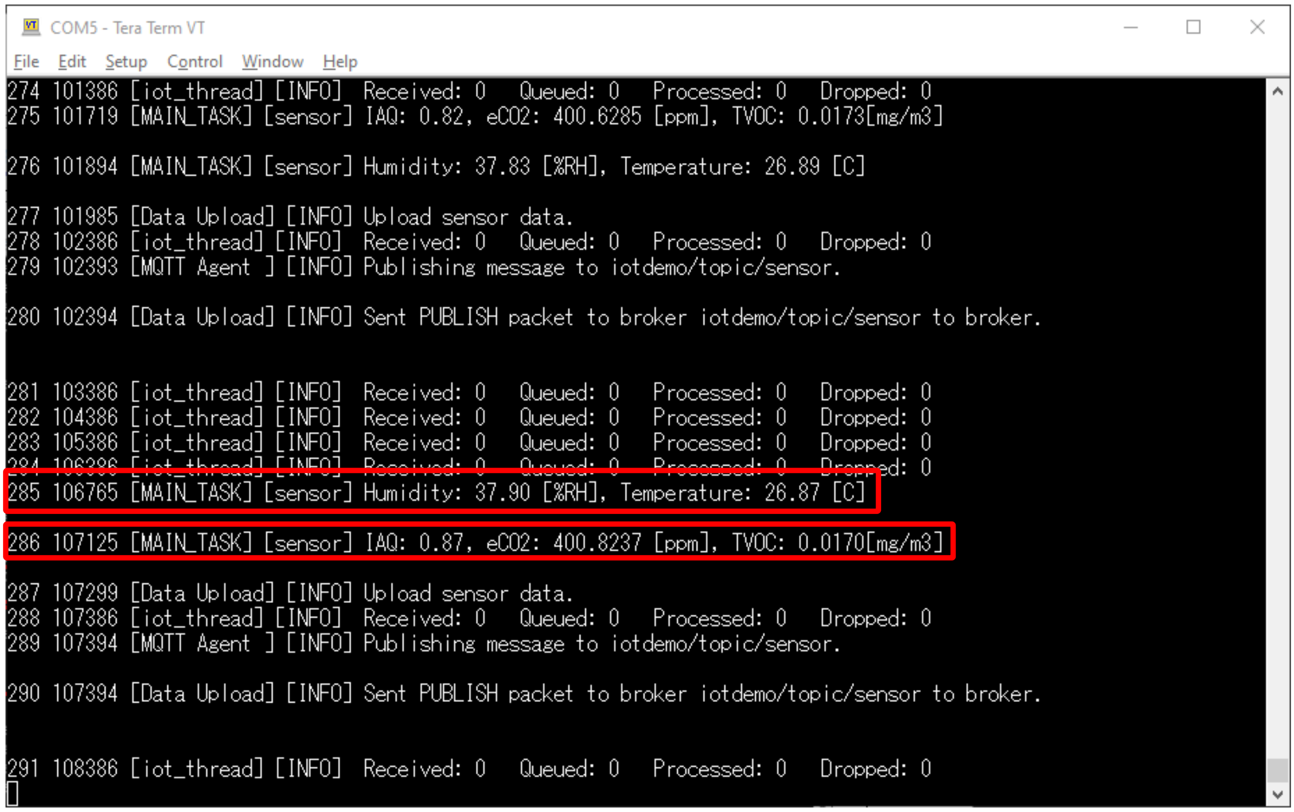

Figure 3-60 RSK+RX65N-2MB Board Log Window after Firmware Update

Figure 3-61 also shows the screen display of the Pmod LCD board connected to the updated RSK+RX65N-2MB board. Similarly, you can see that the data of the ZMOD4410 sensor is displayed.

Figure 3-61 Screen Display of Pmod LCD after Firmware Update

Figure 3-62 shows the log window of the TB-RX23W board.

After the TB-RX23W board firmware update has completed successfully, measurement data can be acquired from both the HS3001 sensor and ZMOD4410 sensor.

The firmware is operating properly if measurement data from the HS3001 and ZMOD4410 sensors is displayed on the serial console of the TB-RX23W board, as shown in Figure 3-62. It takes a few minutes before acquisition of ZMOD4410 measurement data starts.

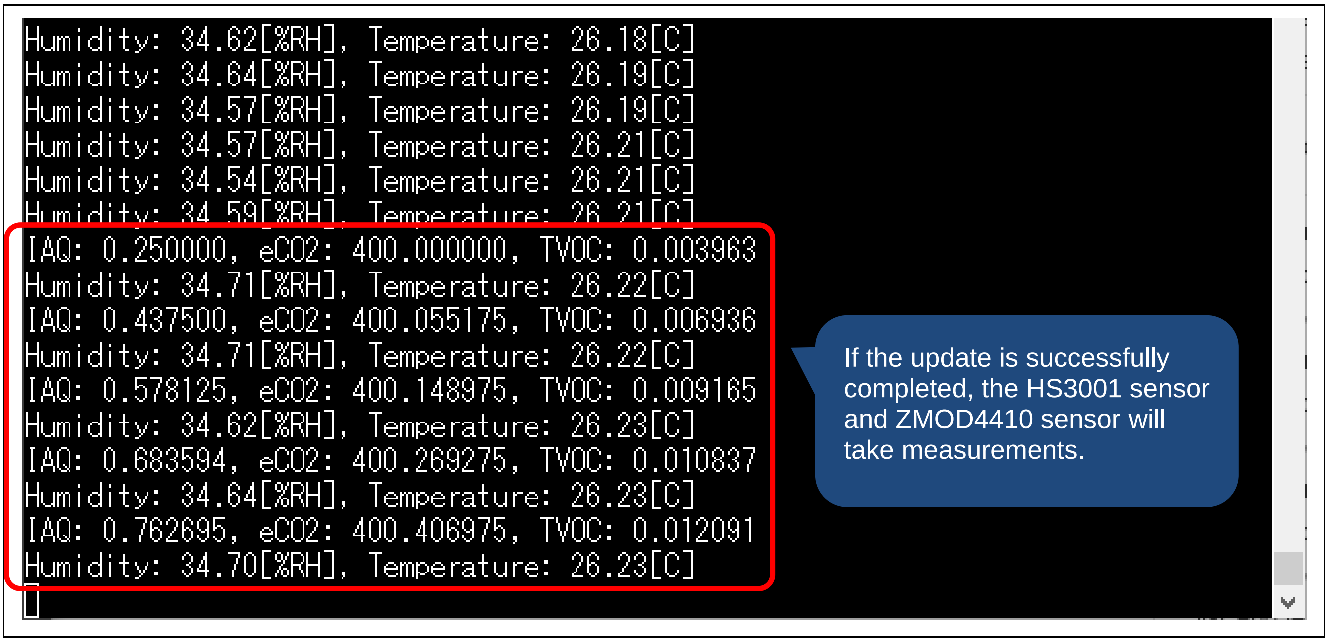

Figure 3-62 TB-RX23W Board Log Window after Firmware Update

Note: If the 2nd OTA update process stops prematurely due to a communication failure between the RSK+RX65N-2MB board and the TB-RX23W board, the 2nd OTA update process is discarded after a certain period of time and returns to normal operation. At that time, the logging of the TB-RX23W board to the terminal software on the PC may become garbled. In that case, please re-launch the terminal software.

3.4.3 Uploading to Cloud and Visualization of Data of HS3001 Sensor and ZMOD4410 Sensor

Confirm that both the HS3001 sensor data and the ZMOD4410 sensor data are plotted on the graph in AWS OpenSearch Dashboards, as shown in Figure 3-63.

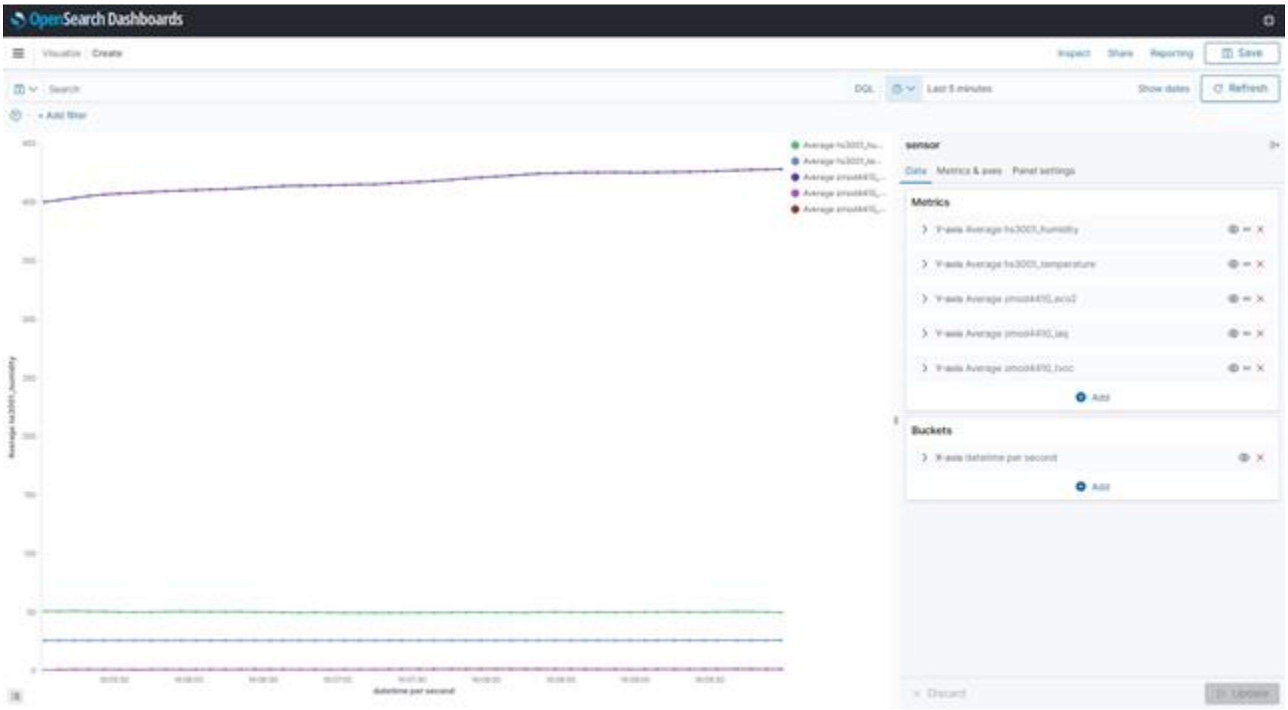

Figure 3-63 Graph Display of OpenSearch Dashboards after 2nd OTA update

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.01	Jan. 24, 2022	—	First edition issued.
1.10	Mar. 31, 2022	—	Supported AWS IoT Over-the-air Update Library v3.0.0.
		5 - 8	1.2.1 Added .settings folder to the folder structure of each project.
		5 - 8	1.2.1 Revised package and folder structure for RX65N project.
		9	1.3 Updated IDE environment to e2studio 2022-01, Toolchain to CC-RX V3.04.00 and FreeRTOS for RX65N project to Version 2021.07.
		9	1.4 Updated code size.
		19 - 20	3.3.1 Updated initial firmware creation method due to RX65N project changes.
		47	3.4.1 Updated screenshot of output log due to RX65N project changes.
		51 - 55	3.4.2 Changed the method of executing an update.
		56	3.4.2 Updated screenshot of output log due to RX65N project changes.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.