Using AWS S3 OTA Application with RAFW Connectivity Devices

This document shows how to use the reference application for over-the-air (OTA) firmware update using AWS IoT S3 Bucket.

Contents

1. Introduction **3**
1.1 Software Requirements 3
1.2 Hardware Requirements 3
2. Set Up and Configure Hardware **4**
3. Reference Application Setup **4**
3.1 Hardware Setup 4
3.2 Program Firmware Image 4
3.3 AWS Account Preparation 4
3.3.1 Create S3 Bucket 5
3.3.2 Create AWS IoT Thing and Certificate 7
3.3.3 Create Policies 7
3.3.4 Create and Configure IAM Role 7
3.3.5 Create and Configure IAM User 8
3.3.6 Create Role Alias 10
3.3.7 Attach Role Alias Policy to Certificate 11
3.3.8 Request Security Token 11
4. AWS S3 OTA Update Example **13**
5. Run S3 OTA Update Reference Application **14**
6. Revision History **15**

Figures

Figure 1. Open AWS S3 console 5
Figure 2. Create S3 bucket 5
Figure 3. Upload firmware file in S3 bucket 6
Figure 4. Add files to S3 bucket 6
Figure 5. Upload added files to S3 bucket 6
Figure 6. Add customer managed policy to IAM role 8
Figure 7. Add AWSIoTFullAccess policy to IAM role 8
Figure 8. Add Customer managed policy to IAM user 9
Figure 9. Add AWSIoTFullAccess policy to IAM user 9
Figure 10. Create IAM user 10
Figure 11. Create role alias 10
Figure 12. Add properties for role alias 11

© 2026 Renesas Electronics

Terms and Definitions

AP	Access point
API	Application Programming Interface
ARM	Advanced RISC Machine
AWS	Amazon Web Services
CA	Certificate Authority
CRC	Cyclic Redundancy Check
EVK	Evaluation Kit
FSP	Flexible Software Package
GCC	GNU Compiler Collection.
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
IoT	Internet of Things
MCU	Microcontroller Unit
NVRAM	Non-Volatile Random-Access Memory
OTA	Over-the-Air
PC	Personal Computer
PEM	Privacy Enhanced Mail
S3	Simple Storage Service
SigV4	AWS Signature Version 4
URL	Uniform Resource Locator
USB	Universal Serial Bus
Wi-Fi®	Wireless Fidelity

1. Introduction

To get started with OTA update feature of AWS IoT using S3 Bucket, begin with a ready-to-use reference application. The provided application project template, `awsiot_wifi_s3_ota_ek_ra6w1` and `awsiot_wifi_s3_ota_ek_ra6w2`, includes pre-configured project settings, middleware, and example code, which helps save time and ensures that the environment is correctly set up for development. This application demonstrates how to implement the OTA update on RA6W1/RA6W2 using the AWS IoT S3 Bucket and AWS Signature Version 4 (SigV4) for authenticated time-bound access. The solution enables devices to securely download firmware images from S3, validate update packages, and perform remote upgrades with minimal manual intervention. The project highlighted end-to-end OTA workflow design, including firmware version management, cloud-to-device orchestration, secure artifact delivery, and device-side update execution, showing how AWS IoT services can be integrated for scalable and secure fleet maintenance. The solution architecture included device connectivity through AWS IoT Core, cloud-side update orchestration, and firmware artifact management in S3. Firmware binaries were uploaded to an S3 bucket and made available to target devices through pre-signed or SigV4-authenticated requests, ensuring that only authorized devices could download the update package. On the device side, it handles the artifact download, integrity verification, version validation, and firmware installation, which is followed by a reboot.

The following are the main components of AWS IoT S3 OTA update reference application:

- EK-RA6W1/RA6W2 EVK
- AWS IoT S3 OTA update application package
- Terminal application
- Router: Connection to internet
- AWS account.

1.1 Software Requirements

The following software tools and packages are required to complete the project:

- Renesas Electronics Flexible Software Package (FSP): Version 2.2.0 or later.
- e^2 studio IDE: Version 2026-04.3 or later.
- GCC Arm Embedded Toolchain: Version 13.3.1.arm-13-24.
- Serial Terminal Application: Tera Term or similar tools for logging and monitoring output.

1.2 Hardware Requirements

- Renesas RA6W1/RA6W2 Evaluation Kit (EVK).
- RA6W1/RA6W2 MCUwith ultra-low power consumption.
- USB Type-C Cable: For programming and debugging.
- Jumper Wire (Female to Female): For making proper motherboard configurations for the application.
- Host PC: For programming, viewing console logs.
- Wi-Fi Access Point (AP): With network connectivity.

2. Set Up and Configure Hardware

To connect hardware:

1. Connect the RA6W1/RA6W2 MCU to the Evaluation Kit (EVK) using the extension socket.
2. Connect the USB Debug port on the EVK to the host PC for programming and debugging using a USB Type-C cable.
3. Set up jumpers as following:
   - UART (RX): P0_00
   - UART (TX): P0_01
   - J102: {2-3} [Current Measurement]
   - J105: {2,3} [Voltage Selector]
   - J106: {2,3} [Voltage Selector]
   - J107: {2,3} [Voltage Selector]
   - J210: {1,2} [Reset]
   - SWCLK(c): SWCLK
   - SWDIO(c): SWDIO

3. Reference Application Setup

To set up reference application you need to set up hardware, program firmware image, and prepare AWS account.

3.1 Hardware Setup

An RA6W1/RA6W2 Evaluation Board (EK-RA6W1/RA6W2) is required to run the reference application. An external access point with internet access is required for the final Wi-Fi connectivity.

3.2 Program Firmware Image

The firmware image for reference application is generated and flashed onto the EK-RA6W1/RA6W2 using e^2 studio with J-Link support. e^2 studio provides an integrated build and flash environment, eliminating the requirement for external flash programming scripts.

To flash image:

1. Open the e² studio and connect your EK-RA6W1/RA6W2 board to the PC with the USB (using the inbuilt SEGGER J-Link).
2. Load the reference application to e^2 studio. The reference application for AWS IoT Door Lock is in the `application_projects/r19an0519/AWS_S3_OTA_Update_Reference_Application/` path. For more information on how to load the template, see the Load Project Template in e^2 studio section in *RA6W1/RRQ61001 Application Development Guide* or *RA6W2/RRQ61051 Application Development Guide*.
3. Edit the properties of AWS Stack based on the configurations made in Section 6.3.2. For more information on how to edit AWS Stack Properties, see the Load AWS-IoT Reference Application in e² studio section in *RA6W1/RRQ61001 AWS IoT Core Quick Start Guide* or *RA6W2/RRQ61051 AWS IoT Core Quick Start Guide*.
4. Flash the image using debugger. For more information, see the Flash and Debug using e² studio section in *RA6W1/RRQ61001 Application Development Guide* or *RA6W2/RRQ61051 Application Development Guide*.

After flashing is complete, the device starts running the firmware.

3.3 AWS Account Preparation

To set up AWS account you need to create:

- S3 Bucket (Section 3.3.1 Create S3 Bucket)
- AWS IoT thing and certificate (Section 3.3.2 Create AWS IoT Thing and Certificate)
- Policies (Section 3.3.3 Create Policies)
- IAM role (Section 3.3.4 Create and Configure IAM Role)
- IAM user (Section 3.3.5 Create and Configure IAM User)
- Role alias (Section 3.3.6 Create Role Alias and Section 3.3.7 Attach Role Alias Policy to Certificate)
- Security token (Section 3.3.8 Request Security Token)

3.3.1 Create S3 Bucket

1. Login to the AWS account and select **S3**.

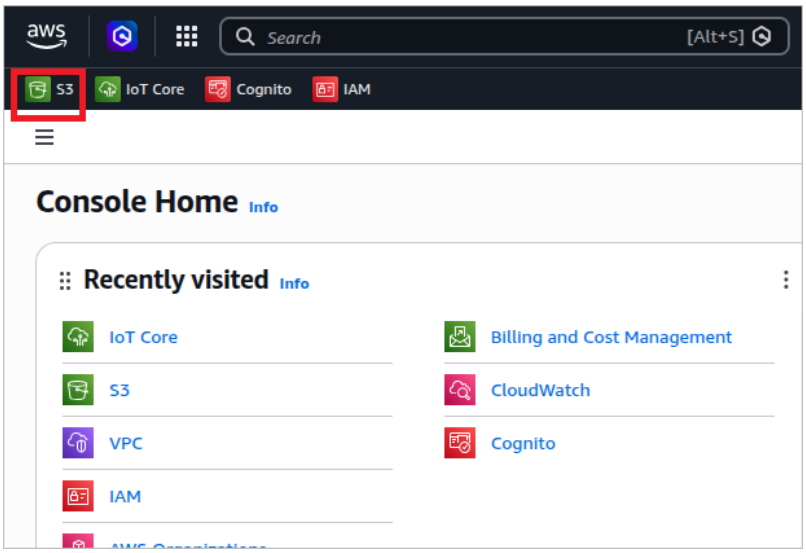


Figure 1. Open AWS S3 console

2. In **S3 Console**, select **Create bucket**.

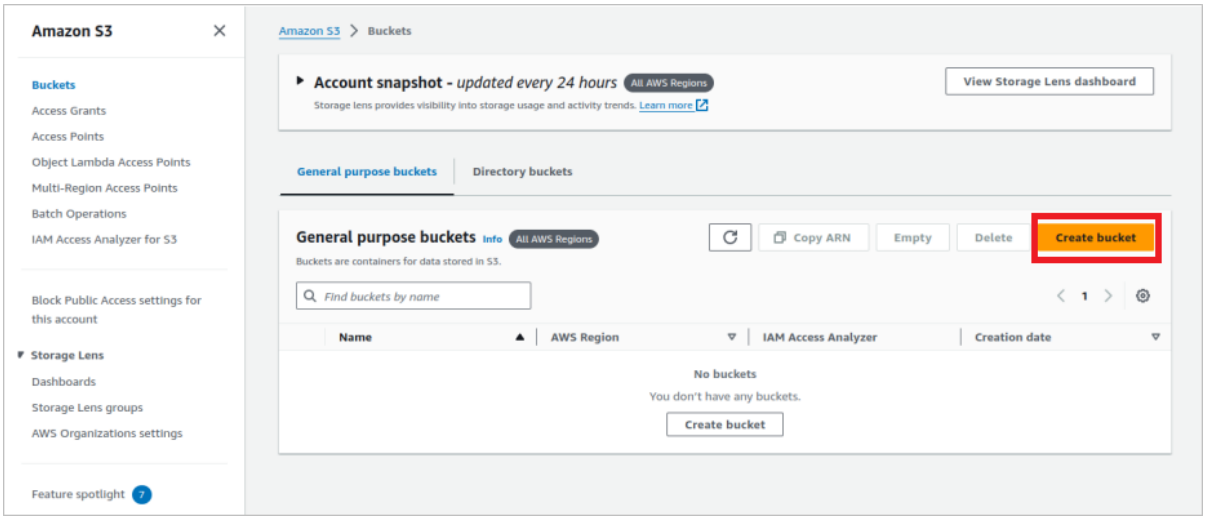


Figure 2. Create S3 bucket

3. Provide a bucket name, and then select **Create bucket**.
4. In the created S3 bucket, you need to upload the firmware file that is later downloaded from S3 to the device, select **Upload**.

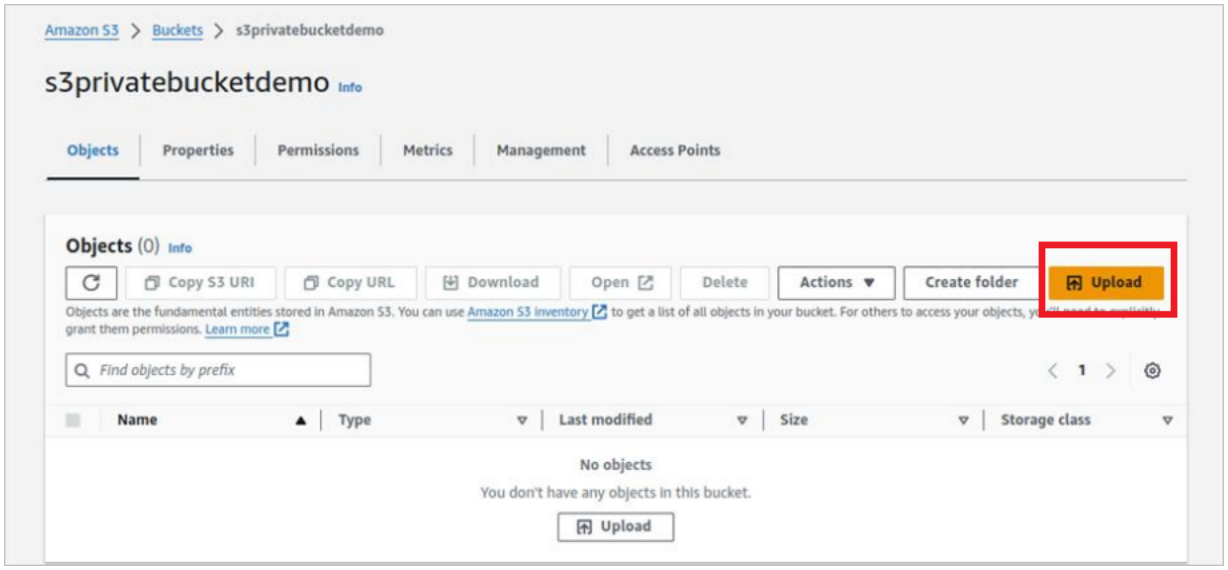


Figure 3. Upload firmware file in S3 bucket

NOTE
While creating a bucket, by default, it is private.

5. In the **Upload**, drag and drop or select **Add files** to upload.

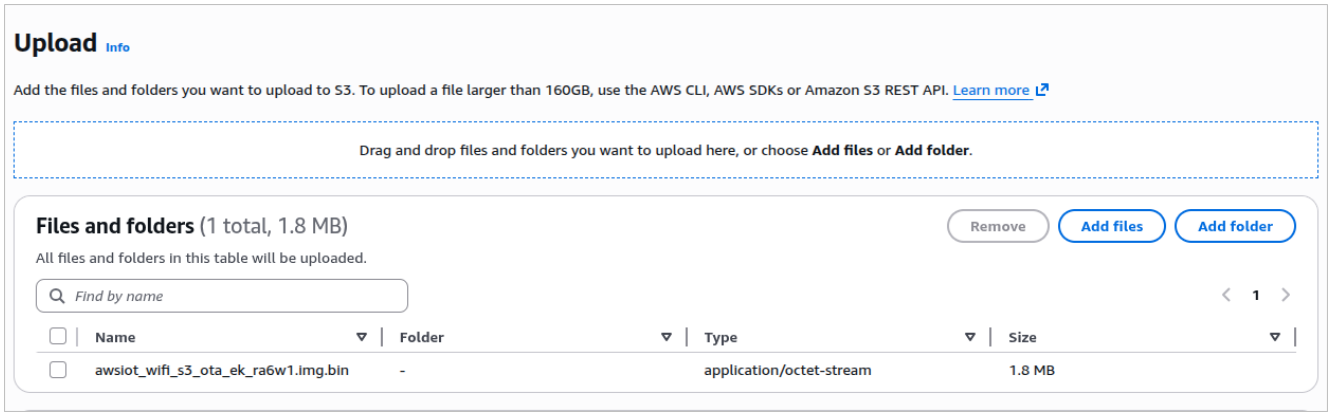


Figure 4. Add files to S3 bucket

6. After files are added, select **Upload**

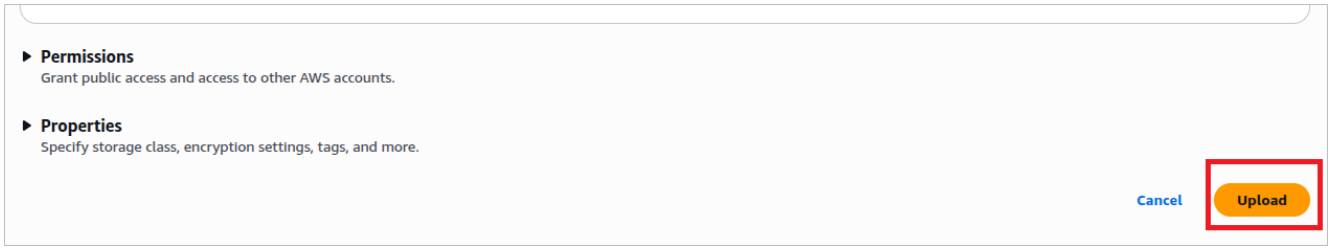


Figure 5. Upload added files to S3 bucket

Now, the S3 bucket is created and has files in it.

3.3.2 Create AWS IoT Thing and Certificate

1. For creating AWS IoT Thing follow the Register Device section in *RA6W1/RRQ61001 AWS IoT Core Quick Start Guide* or *RA6W2/RRQ61051 AWS IoT Core Quick Start Guide*.
   For **Shadow**, you can select the **No Shadow** option.
2. For creating and configuring certificate to thing, follow the Create and Activate Device Certificate section in *RA6W1/RRQ61001 AWS IoT Core Quick Start Guide* or *RA6W2/RRQ61051 AWS IoT Core Quick Start Guide*.

3.3.3 Create Policies

The access in AWS account is managed by creating policies and attaching them to IAM identities. To create a policy:

1. Go to **IAM** > **Access Management** > **Policies**, and select **Create policy**.
2. In **Policy editor**, select **JSON** and copy the following JSON that is used for performing actions on S3.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "s3:GetObject"
        ],
        "Resource": "arn:aws:s3:::<BUCKET_NAME>/*"
    }
}
```

NOTE
<BUCKET_NAME> is the bucket name previously created.

3. Select **Next**.
4. In **Review and create**, enter the policy name, and then select **Create policy**.

Now the policy is created and can be found in IAM > Access Management > Policies.

3.3.4 Create and Configure IAM Role

An IAM User and an IAM Role are both AWS Identity and Access Management (IAM) identities, but they are used for different purposes.
An IAM User is a permanent identity created in your AWS account for a person or an application. Typically assigned to human users or applications.

1. Go to **IAM** > **Access management** > **Roles**, and select **Create role**.
2. Select **Custom trust policy**and copy the following trust policy (JSON) to grant the credentials provider permission to assume the role.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": {"Service": "credentials.iot.amazonaws.com"},
        "Action": "sts:AssumeRole"
    }
}
```

3. Select **Next**.
4. In **Filter by Type**, select **Customer managed**, and then select the previously created policy from Section 3.3.3 Create Policies.

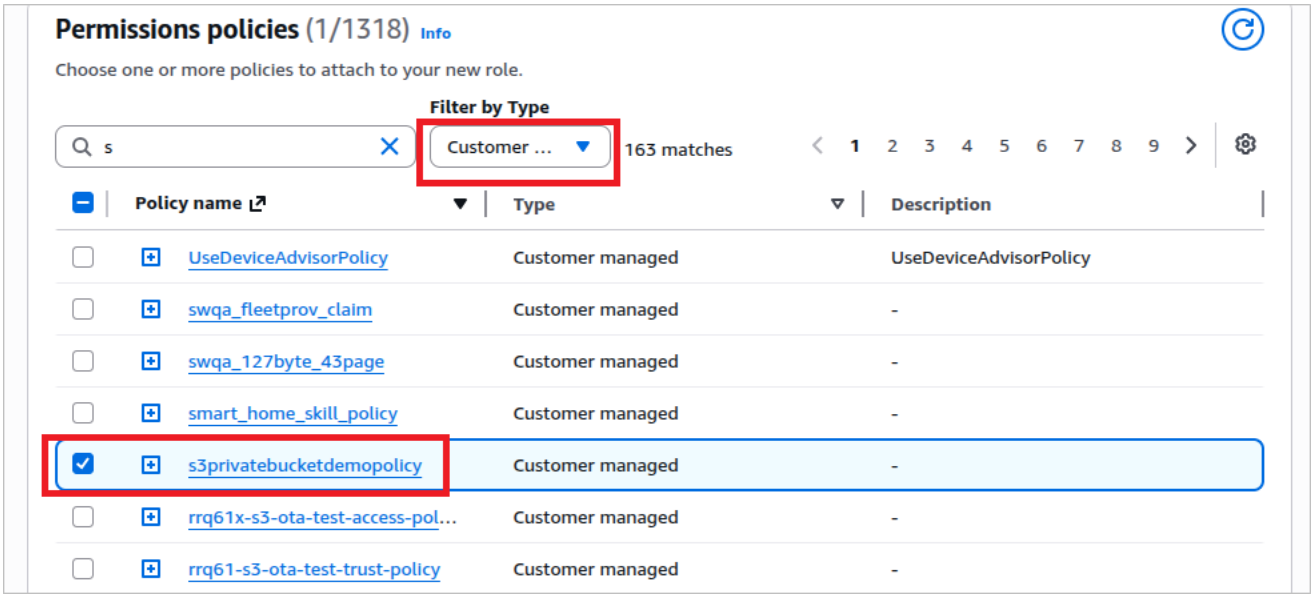


Figure 6. Add customer managed policy to IAM role

5. Switch to the **All types** filter, and select the **AWSIotFullAccess** policy.
6. Select **Next**.

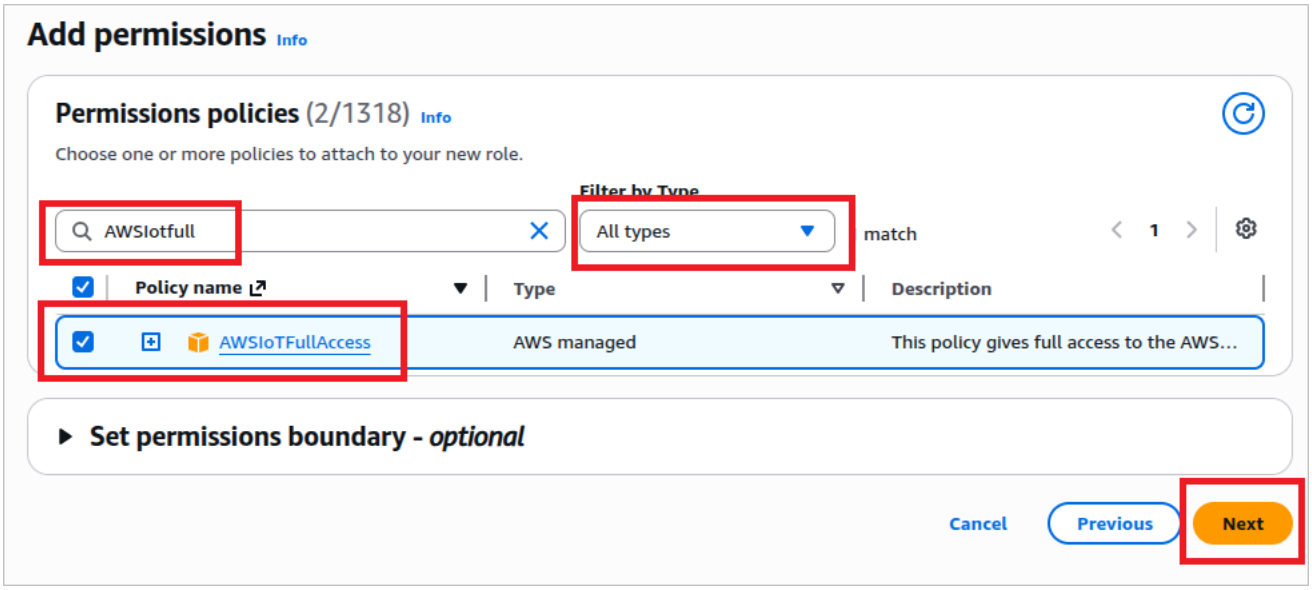


Figure 7. Add AWSIoTFullAccess policy to IAM role

7. Add name for the role and select **Create role**.

The IAM role is created and can be found in IAM > Access Management > Roles.

3.3.5 Create and Configure IAM User

1. Go to **IAM** > **Users** and select **Create User**.
2. Add user name and select **Next**.
3. Select **Attach policies**.
4. Select **Create policy**.
5. The IAM role that you have created must be passed to AWS IoT to create a role alias. To complete this, follow Section 3.3.3 Create Policies with the provided JSON:

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GetRole",
            "iam:PassRole"
       ],
       "Resource": "arn:aws:iam::<your_aws_account_id>:role/<IAM_ROLE_NAME>"
    }
}
```

NOTE

- <IAM_ROLE_NAME>: IAM name created previously.
- <your_aws_account_id>: can be found in right side top of AWS console while selecting account name.

6. In the **Filter by Type** dropdown, select **Customer managed** and select the previously created policy.

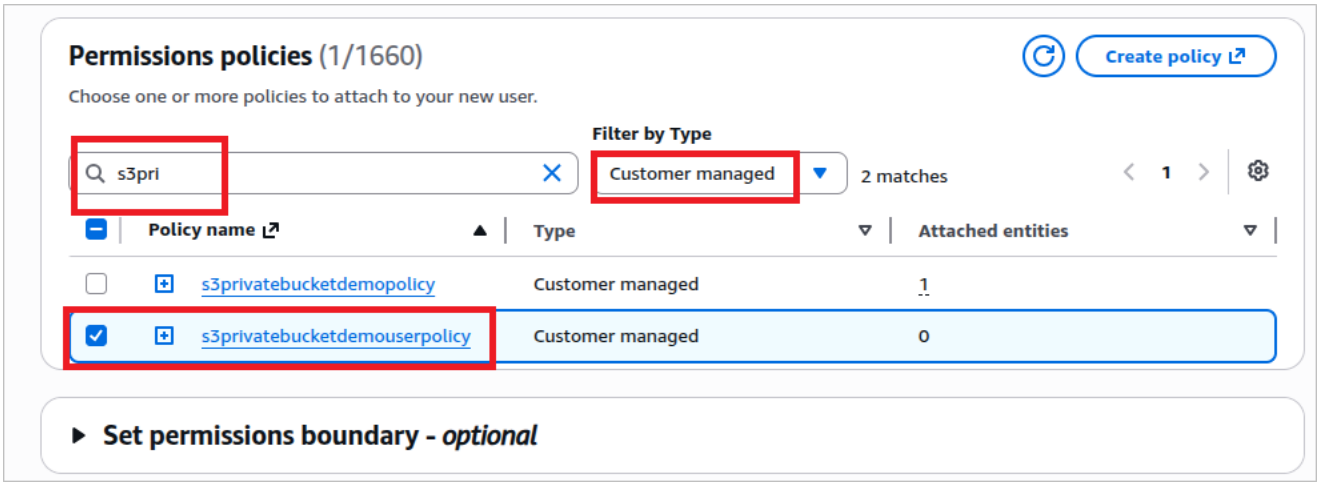


Figure 8. Add Customer managed policy to IAM user

7. Also, select **AWSIotFullAccess** policy from the **AWS managed** filter.
8. Select **Next**.

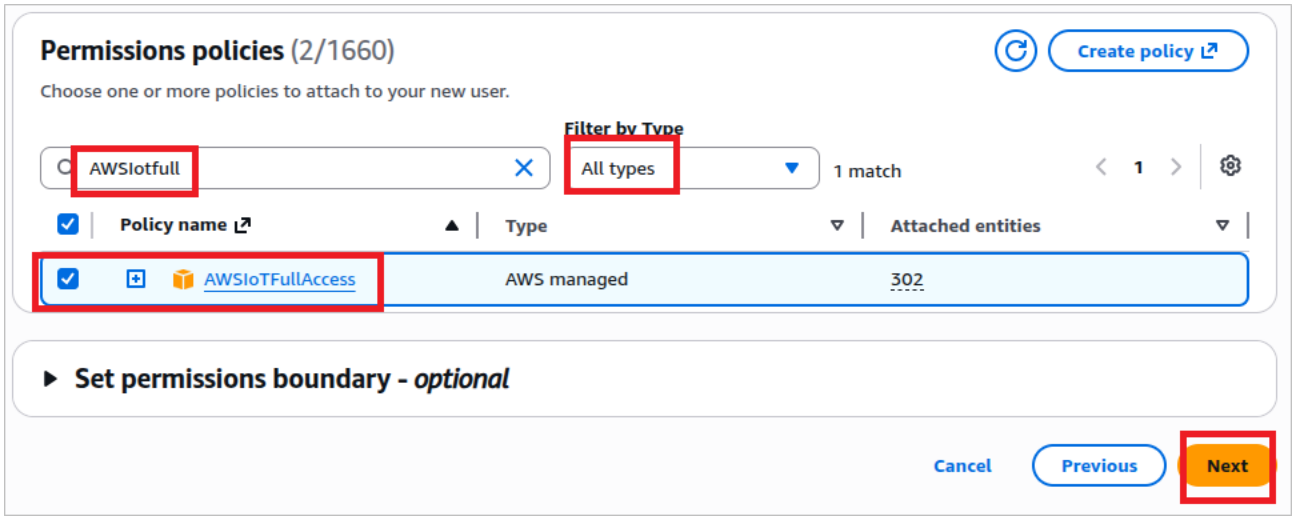


Figure 9. Add AWSIoTFullAccess policy to IAM user

9. Select **Create User**.

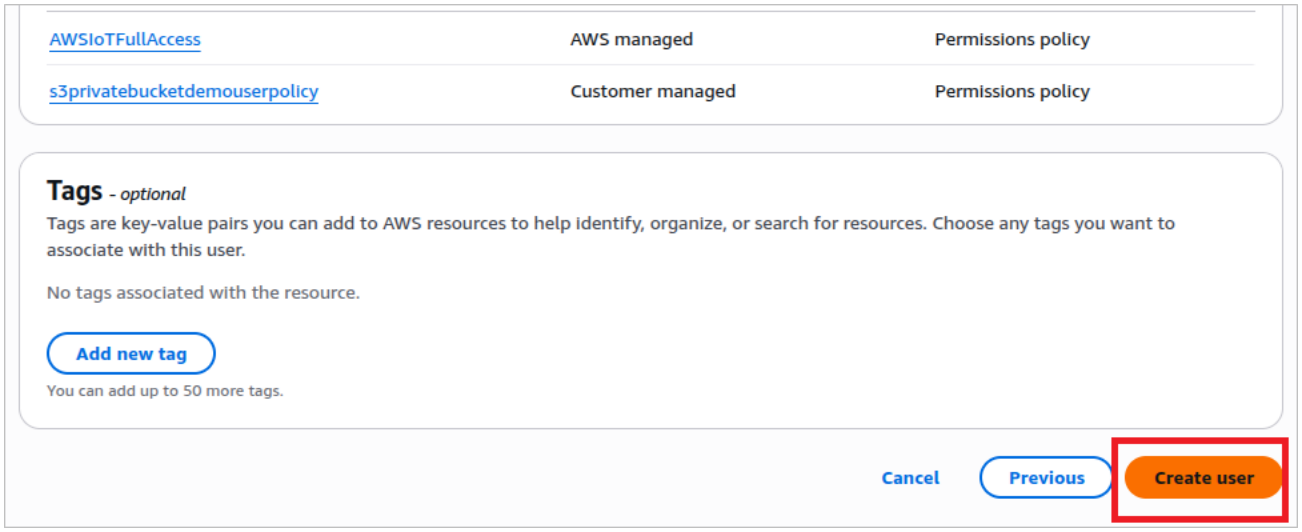


Figure 10. Create IAM user

10. Select **Create user**.
11. Select **Create access key**.
12. Select **CLI**, click the confirmation, and select **Next**.
13. Optionally, give a description tag and select **Create access key**.
14. Download the `.csv` file for storing the Access key and Secret access key.
15. Select **Done**.

3.3.6 Create Role Alias

With the IAM role configured, next, create a role alias with AWS IoT.

1. Go to **IOT** > **Security** > **Role** > **Role aliases**.
2. Select **Create role alias**.

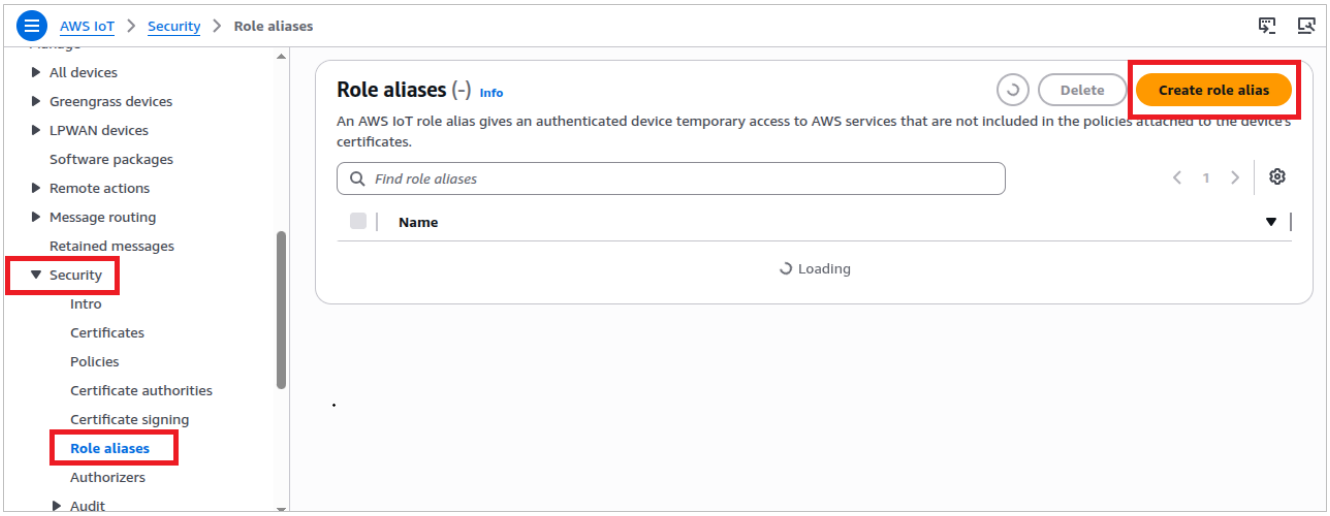


Figure 11. Create role alias

3. Add the alias name, credential duration, and select the created role from Section 3.3.4 Create and Configure IAM Role, and then select **Create**.

NOTE
CredentialDurationSeconds is an optional attribute specifying the validity (in seconds) of the security token. The minimum value is 900 seconds (15 minutes), and the maximum value is 3600 seconds (60 minutes).

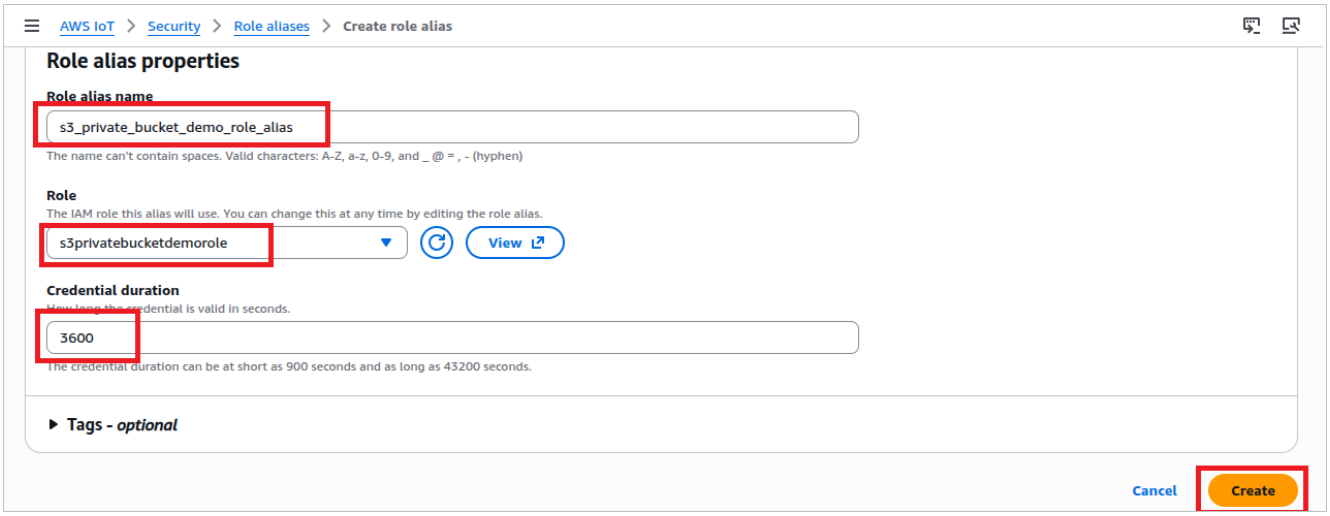


Figure 12. Add properties for role alias

3.3.7 Attach Role Alias Policy to Certificate

1. Create custom policy using given below JSON following steps in the Create and Attach AWS IoT Policy section in *RA6W1/RRQ61001 AWS IoT Core Quick Start Guide* or *RA6W2/RRQ61051 AWS IoT Core Quick Start Guide*.

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
        "Effect": "Allow",
        "Action": "iot:AssumeRoleWithCertificate",
        "Resource": "arn:aws:iot:<Region>:<your_aws_account_id>:rolealias/<name_of_role_alias>"
     }
    ]
}
```

NOTE

- < name_of_role_alias>: Name of the role alias created.
- <your_aws_account_id>: can be found in right side top of AWS console while selecting account name.
- <Region>: Region which is selected in AWS console.

2. Attach the above created policy to certificate by following steps in Create and Attach AWS IoT Policy section in *RA6W1/RRQ61001 AWS IoT Core Quick Start Guide* or *RA6W2/RRQ61051 AWS IoT Core Quick Start Guide*.

3.3.8 Request Security Token

1. To create or get describeEndpoint, install the AWS CLI by following the steps in: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
2. After installing the AWS CLI, configure it by running the following command:

```
aws configure
```

Provide the `aws_access_key_id` and `aws_secret_access_key` (generated in Section 3.3.5 Create and Configure IAM User) with the region and output format.

3. Run the following command for getting a security token after setting the AWS CLI.

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

4. The following JSON object is the sample output of the describe-endpoint command. It contains the endpointAddress that is used to request a security token.

```
{
    "endpoint Address":
    "your_aws_account_specific_prefix.credentials.iot.your region.amazonaws.com"
}
```

4. AWS S3 OTA Update Example

This application provides an over-the-air (OTA) firmware update mechanism that automatically checks device configuration, securely downloads new firmware from AWS S3, validates it, and prepares the system to boot into the updated image.

The application flow is the following:

- The application first reads the `ota_flag` value from NVRAM.
- If `ota_flag` is set to 1, it reads `ota_url`.
- When both the OTA flag and URL are available, the OTA update thread starts.
- At the start of the OTA process, the application updates NVRAM:
  - `ota_result` is set to 1 to indicate the OTA process has begun.
  - `ota_flag` is set to 0xff to prevent the process from being triggered again unintentionally.
- The OTA thread then tries to connect to the HTTP credentials provider server.
- If the connection fails, it waits for a timeout period and retries.
- The process stops retrying when either the maximum timeout or maximum retry count is reached.
- If all attempts fail, the OTA process is marked as unsuccessful.
- After a successful connection, the application uses AWS-provisioned certificate-based credentials to contact the credentials provider endpoint.
- Application requests temporary security credentials. These temporary credentials are used to securely access the firmware image stored in AWS S3.
- After the credentials are received, the connection to the credentials provider is closed.
- The application opens a secure connection to the S3 server using the CA certificate and the configured S3 URL.
- Application sends a SigV4-signed HTTP request to download the firmware object from S3. The temporary AWS credentials from the credentials provider are used to sign the request.
- If the signature is valid and the credentials have permission, S3 allows the firmware download.
- If the download fails or the connection is interrupted, the application disconnects and retries as needed.
- When the firmware download completes successfully, the application sets `ota_result` to 0 in NVRAM. This indicates that the image was downloaded successfully.
- The application continues to the firmware update flow. Before activating the new firmware, the application validates the downloaded image completely. Validation includes:
  - Confirming the firmware was fully downloaded
  - Verifying the firmware version
  - Checking compatibility with the target device
  - Validating the CRC for integrity.
- If all validation checks pass, the application toggles the boot index so that the system boots from the new firmware image.

5. Run S3 OTA Update Reference Application

1. Import the AWS S3 OTA Update Example project into e^2 studio.
2. In AWS IoT port layer stack (`rm_awsiot_w`), configure the following:
   a. AWSIOT_W Thing Name: Device thing name configured in the AWS account.
   b. AWSIOT_W ROOT CA Certificate: AWS IoT RootCA1 which can be downloaded from AWS account.
   c. AWSIOT_W Client Certificate: AWS IoT Certificate in PEM which can be downloaded from AWS account.
   d. AWSIOT_W Client Private Key: AWS IoT Private Key in PEM which can be downloaded from AWS account.
   Use the default configurations to test with the demo account.
3. In the `config_s3_http.h` file, configure the following (the path is `src/freertos_plus_demo_helpers`):
   a. democonfigIOT_CREDENTIAL_PROVIDER_ENDPOINT: Endpoint for the AWS IoT credential provider.
   b. democonfigIOT_CREDENTIAL_PROVIDER_ROLE: Role alias name for accessing the credential provider, configured in the AWS account.
   c. democonfigS3_BUCKET_NAME: Name of bucket in AWS S3 from where file must be downloaded.
   d. democonfigS3_BUCKET_REGION: AWS Region where the bucket resides.
   e. democonfigS3_OBJECT_NAME: Name of file that must be downloaded from AWS S3.
   Use the default configurations to test with demo account.
4. Generate and build the Example project.
5. Connect the RA6W1/RA6W2 MCU motherboard debug port to the host PC with a type C USB cable.
6. Debug or flash the image to the RA6W1/RA6W2 board.
7. After flashing, press the Reset button.
8. Use any terminal application for logging.
9. During the initial bootup, set up the Wi-Fi details for the internet connection.
10. Write the S3 Bucket URL (`ota_url`) and OTA flag (`ota_flag`) into the NVRAM using CLI commands:
    a. `nvram setenv appcfg ota_flag 1`
       1 denotes the OTA update is ready.
    b. `nvram setenv appcfg ota_url <URL>`
       The URL for demo is aws-s3-ota-bucket.s3.ap-northeast-2.amazonaws.com.
11. Reboot the device using the cli command `reboot` or by using the Reset button.
12. After rebooting, the RA6W1/RA6W2 connects to the configured AP.
13. After connection to AP is successful, RA6W1/RA6W2 checks if the OTA flag is set ready and OTA URL is available. If it is available, RA6W1/RA6W2 starts the OTA update demo and starts to download the image file from the S3 bucket. If the configuration is not available, an error message appears.
14. The downloaded image is written to flash in the address specific for an updated image.
15. After the whole image is downloaded, the application calls for OTA to renew the API. This reads the image from flash and checks the image version from the version header in the OTA image. If the version check passes, the boot index is updated and the device is rebooted.
    Finally, the device boots up in the updated image.

6. Revision History

Revision	Date	Description
1.02	Sep 21,2026	Updated content for FSP version 2.2.0, Reference document names updated.
1.01	July 21, 2026	Updated content for FSP version 2.1.0.
1.00	June 8, 2026	First version.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

CorporateHeadquarters

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-us/

© 2026 Renesas Electronics Corporation. All rights reserved.