

Custom FOTA User Manual

The target audience of this document are customers of AWS IoT who want to deploy over-the-air firmware updates (FOTA) to devices that have integrated the MQTT File Download Agent.

Customers who have integrated a version of FOTA-capable amazon FreeRTOS on their devices can refer to [other documentations](#) about how to perform FOTA.

Prerequisite

1. The current firmware image on your devices already has the capability to connect to AWS IoT. Most likely, your devices have integrated the [AWS IoT Device SDK for Embedded C](#). In addition, your devices have been registered as things in AWS IoT Core.
2. The current firmware image on your devices already has the MQTT File Download Agent integrated.
3. You have AWS credentials to access the [AWS Console](#).
4. You have installed the [AWS Command Line Interface \(CLI\)](#), and can execute commands.

Step 1 - Create the firmware image

You have developed a new version of firmware and packaged it as an image file. A best practice for firmware update is that you also sign the image digitally, so that the device receiving the image can verify that it has not been tampered with en route. You can use [Code Signing for AWS IoT](#) to sign your files, or you can sign your files with your own code-signing tools.

If you choose to sign the firmware image file digitally, you need to include codes on the device that knows how to verify the signature. How to sign the image file and how to verify it are outside of the scope of this document.

Step 2 - Upload the firmware image file to Amazon S3

Amazon S3 is an AWS service where you can store files.

First, create an S3 bucket.

- In the Console select the [Amazon S3 service](#). Make sure that the **Region** is the same as the one where your devices are provisioned.
- Click on **+ Create bucket**

In bucket name, give it a name, for example, *“iot-fota-bucket”*. In the following screens click **Next** until **Create Bucket** button appears, then click on **Create bucket**.

Second, upload the firmware image file to the bucket. In the Amazon S3 console, search for the bucket you just created and:

- Click on the name of the bucket
- Click on **Upload**
- Select the firmware image file on your local machine, or drag and drop it onto the browser window. For example, let's assume this file has the name of *"fota-image-v1"*. (We will use this file name in the examples below.)

Click **Upload** and the document file should appear in the bucket.

Step 3 - Create an IAM role

You should have installed [AWS CLI](#) on your local machine.

First, create an IAM role that the IoT service principal can assume. Save the following trust policy document on your local machine, name it *iot-role-trust.json*:

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

Execute the following AWS CLI command. This will create an IAM role named *iot-fota-role* using the JSON file above.

```
aws iam create-role --role-name iot-fota-role --assume-
role-policy-document file://iot-role-trust.json
```

Save the following JSON content into another file named *iot-fota-policy.json*.

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

Execute the following AWS CLI commands to create a policy using the JSON file above, and attach it to the role you've created previously. (Please replace the

placeholder account number **123456789012** with your own AWS account number.) When AWS IoT service assumes this role, this policy will grant it access to the files under the named S3 bucket.

```
aws iam create-policy --policy-name iot-fota-policy --  
policy-document file://iot-fota-policy.json  
aws iam attach-role-policy --role-name iot-fota-role --  
policy-arn "arn:aws:iam::123456789012:policy/iot-fota-  
policy"
```

Step 4 - Create a stream

Now, create a stream in AWS IoT that your devices can retrieve the firmware image file from.

Save the following JSON content into a file named **create-stream.json**. (Please replace the placeholder account number **123456789012** with your own AWS account number.)

```
{  
  "streamId": "fota-image-v1-stream",  
  "description": "This stream is used for sending fota-  
image-v1 file to devices.",  
  "files": [  
    {  
      "fileId": any_integer_between_1_to_255,  
      "s3Location": {  
        "bucket": "iot-fota-bucket",  
        "key": "fota-image-v1"  
      }  
    }  
  ],  
  "roleArn": "arn:aws:iam::123456789012:role/iot-fota-  
role"  
}
```

Run the following AWS CLI command to create the stream. This will create a stream in AWS IoT from which devices can retrieve the file. From your device firmware, the stream can be referred to by the value of its streamId (*fota-image-v1-stream*). The file can be referred to by its fileId (number 0).

```
aws iot create-stream --cli-input-json file://create-  
stream.json
```

Step 5 - Create a job

First, create a JSON document to describe what need to be executed for the job, and upload it to the S3 bucket you created before.

Create a file named *iot-fota-job.json* on your local machine, copy and paste the following content into it:

```
{
  "command": "fota",
  "streamId": "fota-image-v1-stream",
  "fileId": any_integer_between_1_to_255,
  "fileSize": your_fota_image_size_in_Bytes
}
```

When you deploy the job, this JSON document will be sent to target devices over a reserved MQTT topic. Your code running on the device will receive it in a callback function registered by using `aws_iot_jobs_subscribe_to_job_messages` [function](#) of the AWS IoT Device SDK for Embedded C. Your code understands that the job is to perform OTA firmware update by interpreting the value of *command* field, then it invokes the MQTT File Download Agent, passing it the values of *streamId*, *fileId* and *fileSize*, then the file download agent will download the file from the stream.

In the [Amazon S3 console](#), search for the bucket named *iot-fota-bucket*, which you've created in step #2.

- Click on the name of the bucket
- Click on Upload
- Select the *iot-fota-job.json* file on your local machine, or drag and drop it onto the browser window.
- Click **Upload** and the document file should appear in the bucket.

The following can be done in AWS CLI as well. However, in order to make it easier to monitor the status of the job, we will use the AWS IoT console to do it.

Open the [AWS IoT Core](#) console.

- Click on **Manage**
- Click on [Jobs](#)
- In the right pane click on the **Create** button

In the following screen click on the button to **create a custom job**. In the next screen, enter a name in the Job ID field, for example, *"iot-fota-job-1"*.

(NOTE: the Job ID must be unique. When manually creating the jobs, you would likely use meaningful names, but in a production setup you would rather use UUIDs for Job ID, and rely on Resource Tags to organize your jobs and make them easily searchable.)

Scroll down, and click on the **Select** link in the **Select devices to update** section. Check the check-boxes next to the individual devices that you want to update. You can also select **Thing Groups** here.

Next, add the job document. Click on the **Select** link in the **Add a job file** section and select the bucket named *iot-fota-bucket*. Select the file named *iot-fota-job.json* by clicking on the **Select** link.

In the **Job Type** section, select whether you want it to be [a snapshot job or a continuous job](#).

- Click **Next**
- Click **Create** (scroll at the bottom of the page if the Create button is not visible)

After clicking **Create**, the new job will appear on the job overview panel. If you now click on *iot-fota-job-1*, you will see that the overall job status is **IN PROGRESS**, and you will see the list of devices that are executing the job. The status of each device can be **Queued**, **In progress** or others, depending on how each device is executing. The job overall IN PROGRESS status means that it is being rolled out to the devices that have been selected as target. The Resource status In progress means that the agent has acknowledged the job and is executing it.

Step 6 - Check the job execution status on devices

If a device that you have selected as target of the job is running, you can check whether it has received the job and is downloading the file.