RAK7258 Supported LoRa Network Servers

AWS IoT Core for LoRaWAN

Execute the following steps to set up your AWS account and permissions:

Set up Roles and Policies in IAM

Add an IAM Role for CUPS Server

Adding an IAM role will allow the Configuration and Update Server (CUPS) to handle the wireless gateway credentials.

This procedure needs to be done only once, but must be performed before a LoRaWAN gateway tries to connect with AWS IoT Core for LoRaWAN.

1. Go to the **IAM Roles** page on the IAM console.
2. Choose **Create role**.
3. On the Create Role page, choose **Another AWS account**.
4. Enter your **Account ID**, then select **Next: Permissions**.
5. In the search box next to the Filter Policies, type ***AWSIoTWirelessGatewayCertManager***.
    - If the search results show the policy named ***AWSIoTWirelessGatewayCertManager***, select it by clicking the checkbox.
    - If the policy does not exist, create one.
        - Go to the **IAM console** .

- Choose **Policies** from the navigation pane.
- Choose **Create Policy**, then select the **JSON** tab to open the policy editor.
- Replace the existing template with trust policy document.

```
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "IoTWirelessGatewayCertManager",
"Effect": "Allow",
"Action": [
"iot:CreateKeysAndCertificate",
"iot:DescribeCertificate",
"iot:ListCertificates",
"iot:RegisterCertificate"
         ],
"Resource": "*"
          }
      ]
}
```

- Choose **Review Policy** to open the Review Page.
- For the Name, type ***AWSIoTWirelessGatewayCertManager***.

> 📝 **NOTE:**
>
> You must enter the name as ***AWSIoTWirelessGatewayCertManager*** and must not use a different name. This is for consistency with future releases.

- For the Description, enter a description of your choice.
- Then choose **Create policy**. You will see a confirmation message showing the policy has been created.

6. Choose **Next: Tags**, then **Next: Review**.

7. In Role name, enter ***IoTWirelessGatewayCertManagerRole***, and then choose to **Create role**.

> 📝 **NOTE:**
>
> You must not use a different name. This is for consistency with future releases.

8. In the confirmation message, choose ***IoTWirelessGatewayCertManagerRole*** to edit the new role.
9. In the **Summary**, choose the **Trust relationships** tab, and then choose **Edit trust relationship**.
10. In the **Policy Document**, change the **Principal** property to represent the IoT Wireless service:

```
"Principal": {
"Service": "iotwireless.amazonaws.com"
},
```

- After changing the Principal property, the complete policy document should look like the following:

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

11. Choose **Update Trust Policy** to save your changes and exit. At this point, you have created the ***IoTWirelessGatewayCertManagerRole*** and you won't need to do this again.

> 📝 **NOTE:**
>
> The examples in this document are intended only for dev environments. All devices in your fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies can vary for your use case. Identify the permission policies that best meet your business and security requirements. For more information, refer to **Example Policies** and **Security Best Practices**

Add IAM Role for Destination to AWS IoT Core for LoRaWAN

Creating a Policy

Creating a policy gives the role permissions to describe the IoT endpoint and publish messages to AWS IoT.

1. Go to the IAM console .
2. Choose **Policies** from the navigation pane.
3. Choose **Create Policy**, then choose the **JSON** tab to open the policy editor. Replace the existing template with this trust policy document:

```
{
"Version": "2012-10-17",
"Statement": [
   {
"Effect": "Allow",
"Action":
[
"iot:DescribeEndpoint",
"iot:Publish"
],
"Resource": "*"
   }
   ]
}
```

4. Choose **Review Policy** to open the Review page.
5. For **Name**, enter a name of your choice.
6. For **Description**, enter a description of your choice.
7. Choose **Create policy**. You will see a confirmation message indicating that the policy has been created.

Creating the Role

1. In the **IAM console**, choose **Roles** from the navigation pane to open the Roles page.
2. Choose **Create Role**.
3. In **Select type of trusted entity**, choose **Another AWS account**.
4. In **Account ID**, enter your AWS account ID, and then choose **Next: Permissions**.
5. Search for the **IAM policy** you just created by entering the policy name in the search bar.
6. In the search results, select the checkbox corresponding to the policy.
7. Choose **Next: Tags**.
8. Choose **Next: Review** to open the Review page.
9. For **Role name**, enter an appropriate name of your choice.
10. For **Description**, enter a description of your choice.
11. Choose **Create role**. You will see a confirmation message indicating that your role has been created.

Updating your Trust Policy

Update your role's trust relationship to grant AWS IoT Core for LoRaWAN permission to assume this IAM role when delivering messages from devices to your account.

1. In the IAM console, choose **Roles** from the navigation pane to open the Roles page.
2. Enter the name of the role you created earlier in the search window, and click on the role name in the search results. This opens up the Summary page.
3. Choose the **Trust relationships table** to navigate to the Trust relationships page.
4. Choose **Edit trust relationship**. The principal AWS role in your trust policy document defaults to root and must be changed. Replace the existing policy with this:

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

5. Choose **Update Trust Policy.** Under Trusted entities, you will see: *The identity provider(s) iotwireless.amazonaws.com*.

Add the Gateway to AWS IoT

Requirements

To complete setting up your gateway, you need the following:

- LoRaWAN region. For example, if the gateway is deployed in a US region, the gateway must support LoRaWAN region US915.
- Gateway LNS-protocols. Currently, the LoRa Basics Station protocol is supported.
- Gateway ID (GatewayEUI) or serial number. This is used to establish the connection between the LNS and the gateway. Consult the documentation for your gateway to locate this value.
- Add minimum software versions required, including Basics Station 2.0.5.

Add the LoRaWAN Gateway

To register the Gateway with AWS IoT Core for LoRaWAN, execute these steps:

1. Go to the **AWS IoT console** .
2. Select **Wireless connectivity** in the navigation panel on the left.
3. Choose **Intro**, and then choose **Get started**. This step is needed to pre-populate the default profiles.
4. Under **Add LoRaWAN gateways and wireless devices**, choose **Add gateway**.
5. In the **Add gateway section**, fill in the **GatewayEUI** and **Frequency band (RF Region)** fields.
6. Enter a descriptive name in the **Name** – optional field. It is recommended that you use the GatewayEUI as the name.
7. Choose **Add gateway**.
8. On the **Configure your Gateway** page, find the section titled **Gateway certificate**.
9. Select **Create certificate**.
10. Once the **Certificate created and associated with your gateway** message is shown, select **Download certificates** to download the certificate (*xxxxx.cert.pem*) and private key (*xxxxxx.private.key*).
11. In the section **Provisioning credentials**, choose **Download server trust certificates** to download the **CUPS (cups.trust)** and **LNS (lns.trust)** server trust certificates.
12. Copy the CUPS and LNS endpoints and save them for use while configuring the gateway.
13. Choose **Submit** to add the gateway.

Add a LoRaWAN Device to AWS IoT

Requirements:

- Locate and note the following specifications about your endpoint device.
    - **LoRaWAN Region**: This must match the gateway LoRaWAN region. The following Frequency bands (RF regions) are supported: o EU868 o US915 o EU433
    - **MAC Version**: This must be one of the following: o V1.0.2 o v1.0.3 o v1.1
    - OTAA v1.0x and OTAA v1.1 are supported.
    - ABP v1.0x and ABP v1.1 are supported.
- Locate and note the following information from your device manufacturer:

- For OTAA v1.0x devices: DevEUI, AppKey, AppEUI
- For OTAA v1.1 devices: DevEUI, AppKey, NwkKey, JoinEUI
- For ABP v1.0x devices: DevEUI, DevAddr, NwkSkey, AppSkey
- For ABP v1.1 devices: DevEUI, DevAddr, NwkSEnckey, FNwkSIntKey, SNwkSIntKey, AppSKey

Verify Profiles

AWS IoT Core for LoRaWAN supports device profiles and service profiles. Device profiles contain the communication and protocol parameter values the device needs to communicate with the network server. Service profiles describe the communication parameters the device needs to communicate with the application server.

Some pre-defined profiles are available for device and service profiles. Before proceeding, verify that these profile settings match the devices you will be setting up to work with AWS IoT Core for LoRaWAN.

1. Navigate to the **AWS IoT console** . In the navigation pane, choose **Wireless connectivity**.
2. In the navigation pane, choose **Profiles**.
3. In the **Device Profiles** section, there are some pre-defined profiles listed.
4. Check each of the profiles to determine if one of them will work for you.
5. If not, select **Add device profile** and set up the parameters as needed. For US 915 as an example, the values are:
   - MacVersion 1.0.3
   - RegParamsRevision RP002-1.0.1
   - MaxEirp 10
   - MaxDutyCycle 10
   - RfRegion US915
   - SupportsJoin true
6. Continue once you have a device profile that will work for you.

7. In the **Service Profiles** section, there are some pre-defined profiles listed. Check each of the profiles to determine if one of them will work for you.

8. If not, select **Add service profile** and set up the parameters as needed. As an example, the default service profile parameters are shown below. However, only the ***AddGwMetadata*** setting can be changed at this time.

    - ULRate 60
    - ULBucketSize 4096
    - DLRate 60
    - DLBucketSize 4096
    - AddGwMetadata true
    - DevStatusReqFreq 24
    - DrMax 15
    - TargetPer 5
    - MinGwDiversity 1

9. Proceed only if you have a device and service profile that will work for you.

Set up a Destination for Device Traffic

Because most LoRaWAN devices don't send data to AWS IoT Core for LoRaWAN in a format that can be consumed by AWS services, traffic must first be sent to a Destination. A Destination represents the AWS IoT rule that processes a device's data for use by AWS services. This AWS IoT rule contains the SQL statement that selects the device's data and the topic rule actions that send the result of the SQL statement to the services that will use it.

For more information on Destinations, refer to the AWS **LoRaWAN Developer Guide** .

A destination consists of a Rule and a Role. To set up the destination, execute the following steps:

1. Navigate to the **AWS IoT console** . In the navigation pane, choose **Wireless connectivity**, and then **Destinations**.
2. Choose **Add Destination**.
3. On the Add destination page, in the **Permissions** section, select the IAM role you had created earlier, from the drop-down.
4. Under **Destination details**, enter ***ProcessLoRa*** as the Destination name, and an appropriate description under **Destination description – optional**.

> 📝 **NOTE:**
>
> The Destination name can be anything. For getting started and consistency, choose ProcessLoRa for the first integration with AWS IoT Core for LoRaWAN.

5. For **Rule name**, enter ***LoRaWANRouting***. Ignore the section **Rules configuration – Optional** for now. The Rule will be set up later in the "Hello World" sample application. See Create the IoT Rule for the destination.
6. Choose **Add Destination**. You will see a message "*Destination added*", indicating the destination has been successfully added.

Register the Device

Now, register an endpoint device with AWS IoT Core for LoRaWAN as follows:

1. Go to the **AWS IoT console** .
2. Select **Wireless connectivity** in the navigation panel on the left.
3. Select **Devices**, then choose **Add wireless device**.
4. On the **Add device** page, select the LoRaWAN specification version in the drop-down under **Wireless device specification**.
5. Under **LoRaWAN specification and wireless device configuration**, enter the **DevEUI** and confirm it in the **Confirm DevEUI** field.
6. Enter the remaining fields as per the OTAA/ABP choice you made above.
7. Enter a name for your device in the **Wireless device name – optional field**.
8. In the **Profiles** section, under **Wireless device profile**, find a drop-down option that corresponds to your device and region.

> 📝 **NOTE:**
>
> Compare your device details to ensure the device profile is correct. If there are no valid default options, you will have to create a new profile. See the Verify Profiles section.

9. Choose **Next**.
10. Choose the destination you created earlier (*ProcessLoRa*) from the drop-down under **Choose destination**.

11. Choose **Add device**.
12. You will see a message saying "*Wireless device added*", indicating that your device has been set up successfully.

Set up the Gateway

- **Set up the Gateway Hardware**
- **Set up the Gateway Software**

Configure the Gateway Device

1. Using your preferred Web browser, input the aforementioned IP Address and you should see the same Log-in Page shown in the following image. Login the credentials provided below:

- Username: **root**
- Password: **root**

Figure 1: Web User Interface Log-in

2. The firmware version 1.2.0065_Release_r209 on the gateway supports AWS IoT Core for LoRaWAN, and it can be verified on **Status** > **Overview** > **System** > **Firmware Version**.

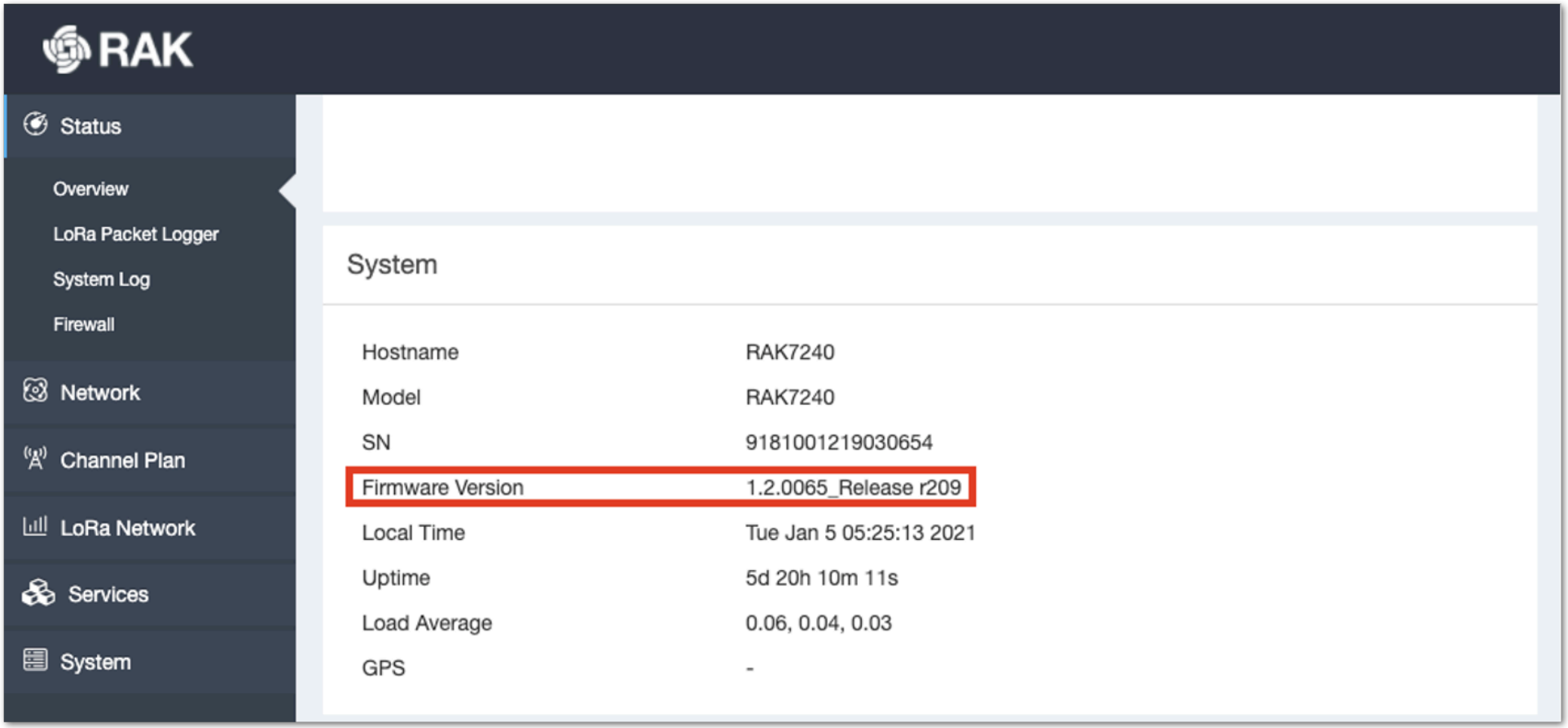


Figure 2: Checking the Firmware Version

3. If the firmware version is prior to 1.2.0065_Release_r209, upgrade the firmware. Navigate to **System** > **Backup/Flash Firmware** > **Flash new firmware image** > **Upgrade the firmware**.

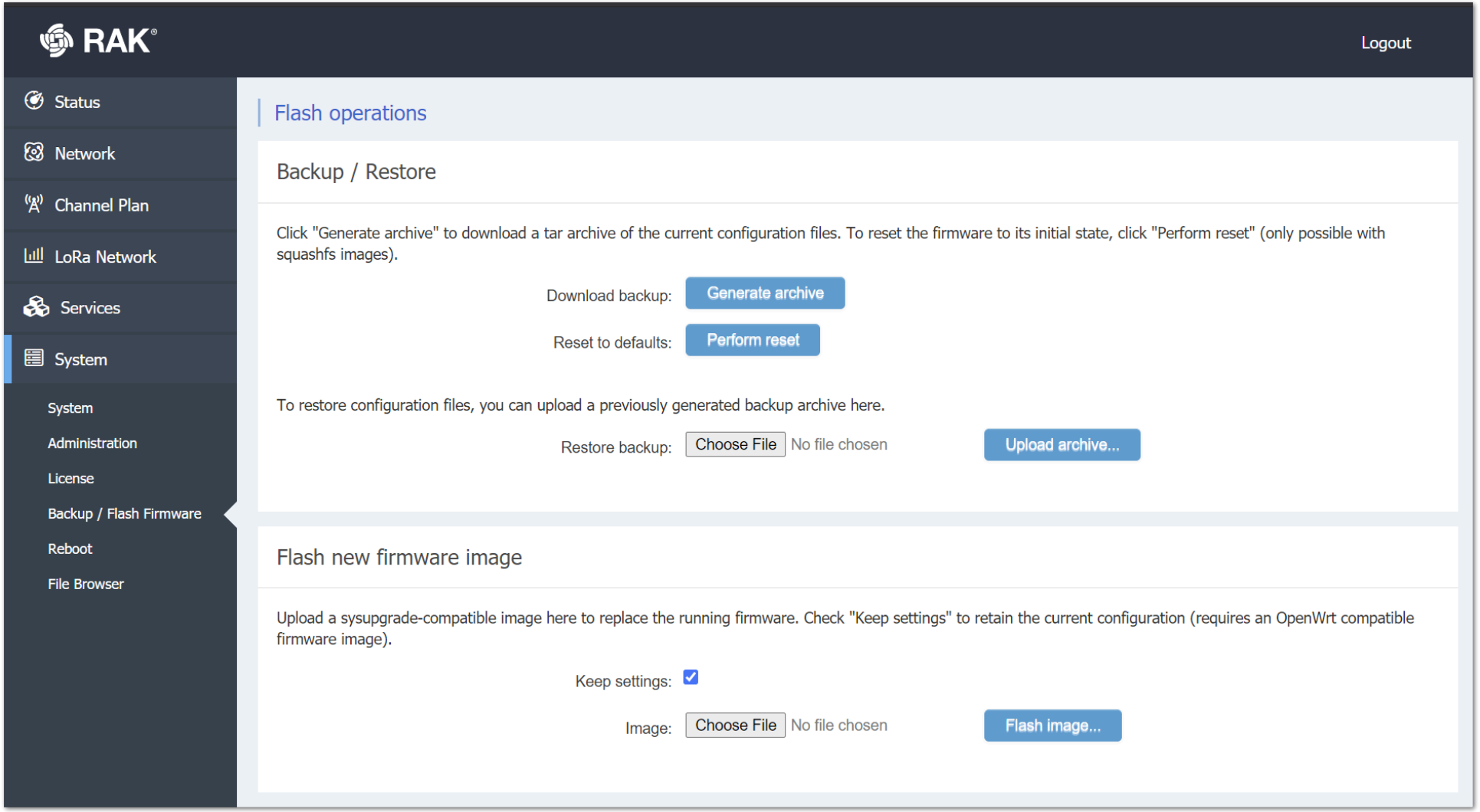


Figure 3: Flashing the firmware

4. Configure Network Mode to Basic Station. Navigate to **LoRa Network** then **Network Settings**.
   - Change the Mode in LoRaWAN Network Settings to Basic Station.
   - Select **LNS Server** from Server, then choose **TLS Server and Client Authentication** from Authentication Mode.

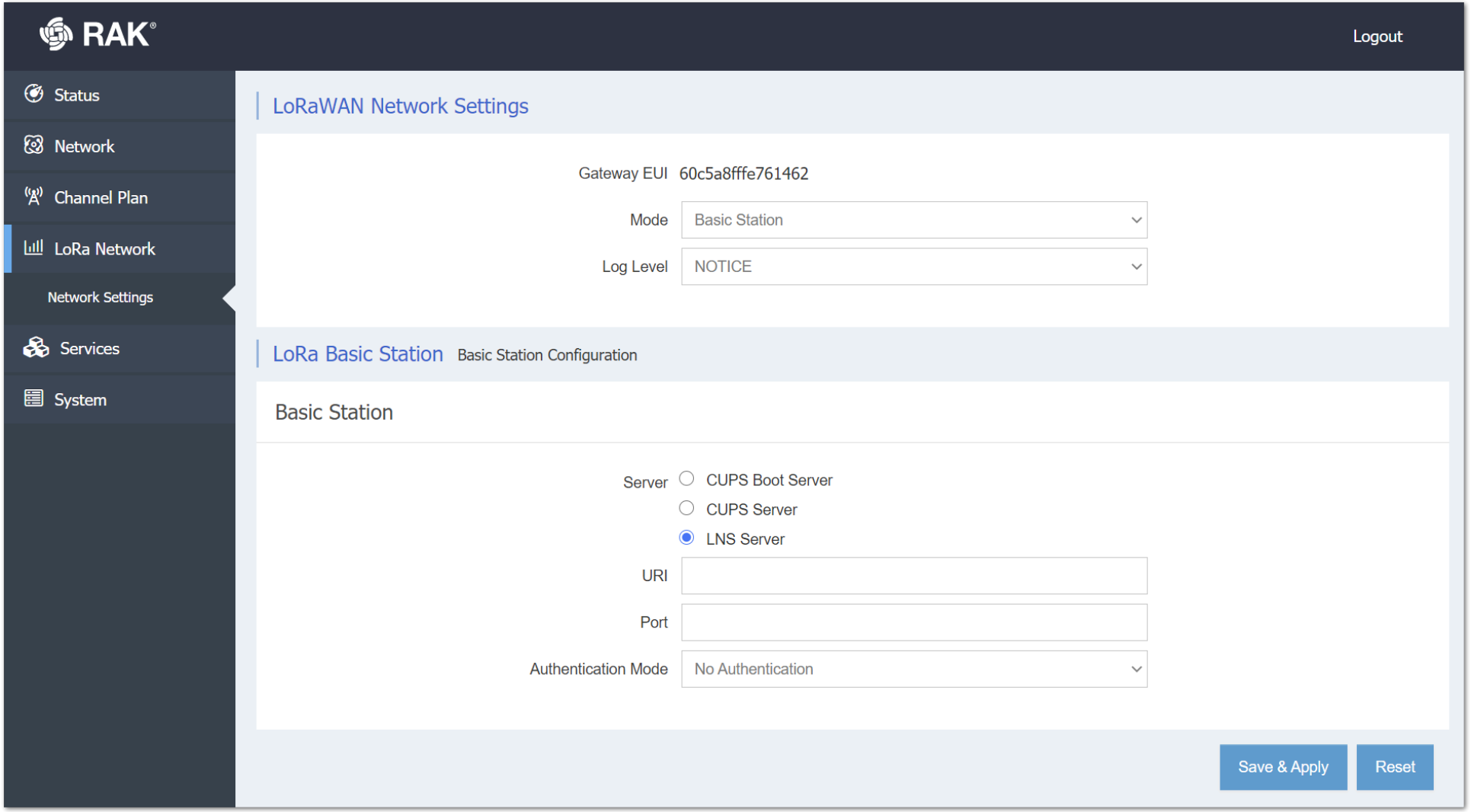


Figure 4: Configuring Network Mode to Basic Station

5. Configure URI, Port, and Authentication Mode.

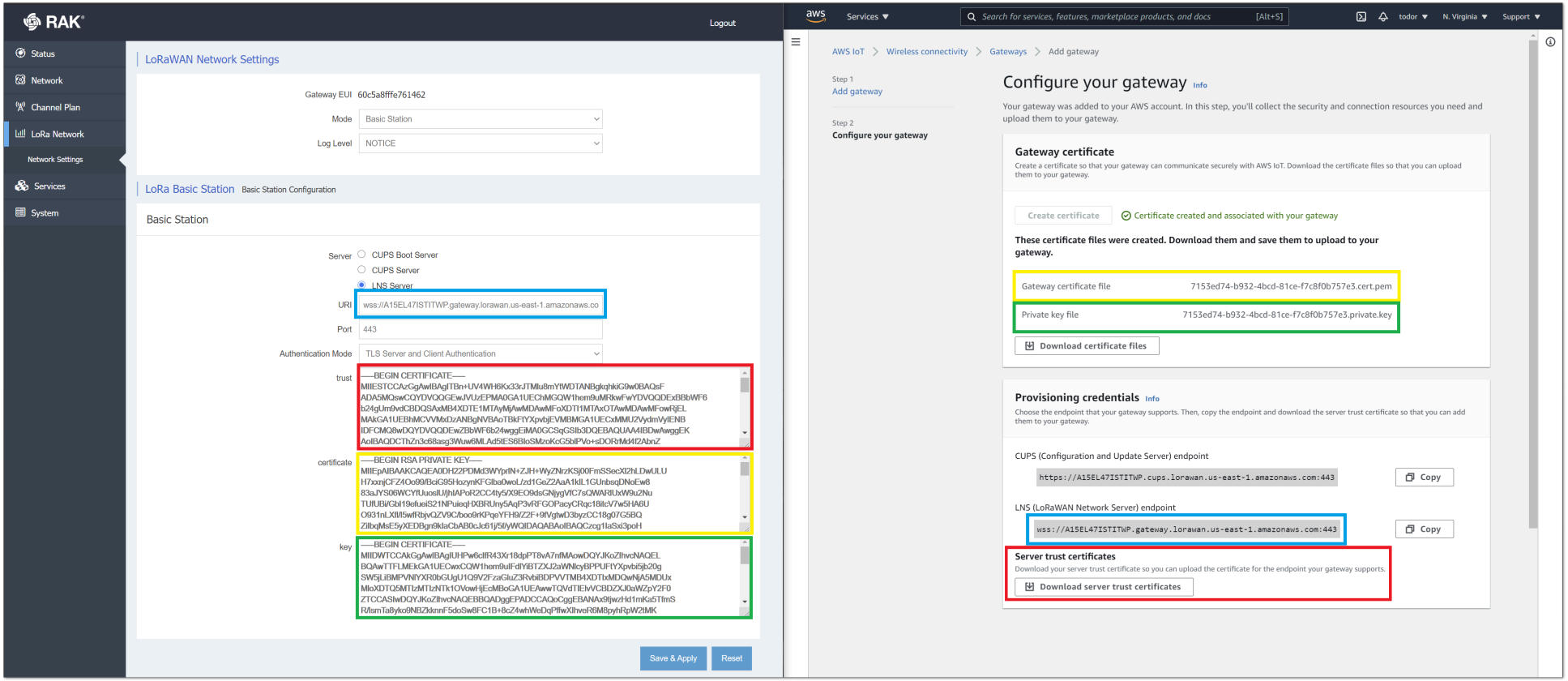
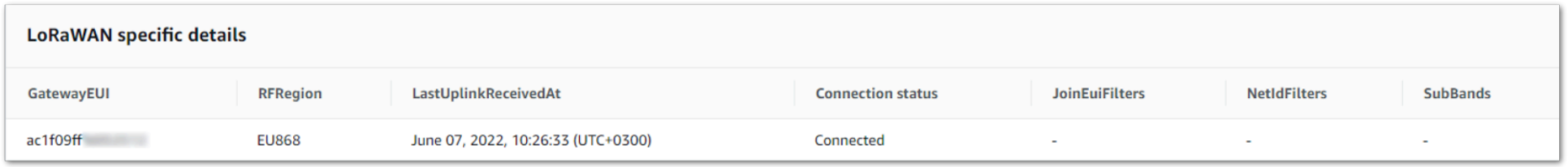


Figure 5: Configuring URI, Port, and Authentication Mode

6. Verifying Operation. Check if the gateway is online in AWS IoT console.



LoRaWAN specific details

GatewayEUI	RFRegion	LastUplinkReceivedAt	Connection status	JoinEuiFilters	NetIdFilters	SubBands
ac1f09ff	EU868	June 07, 2022, 10:26:33 (UTC+0300)	Connected	-	-	-

Figure 6: Verifying Operation

Add End Devices

This section shows an example of how to join the AWS IoT LoRaWAN server.

1. Add Device Profile.

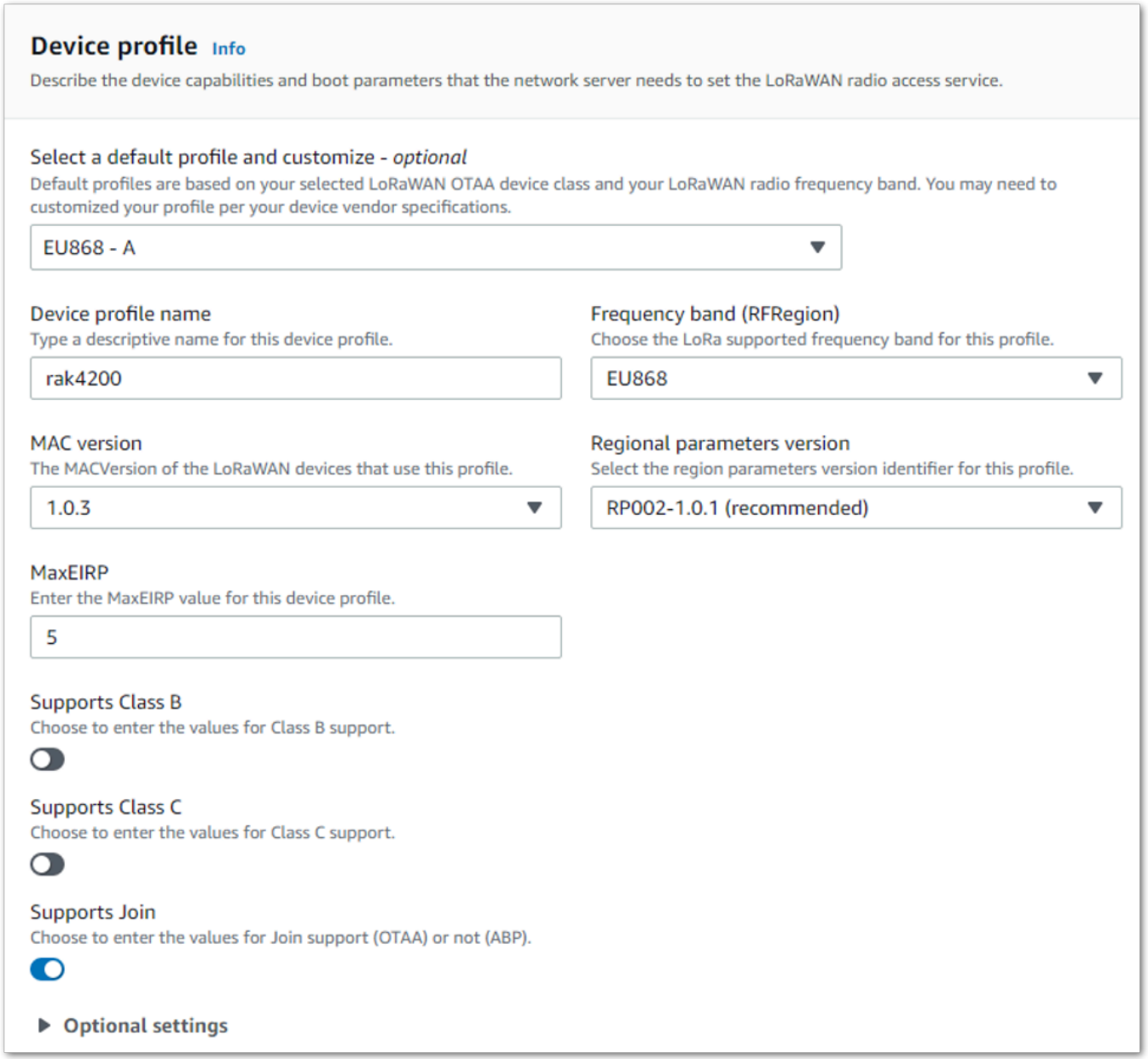


Figure 7: Adding the Device Profile

2. Add Service Profile.

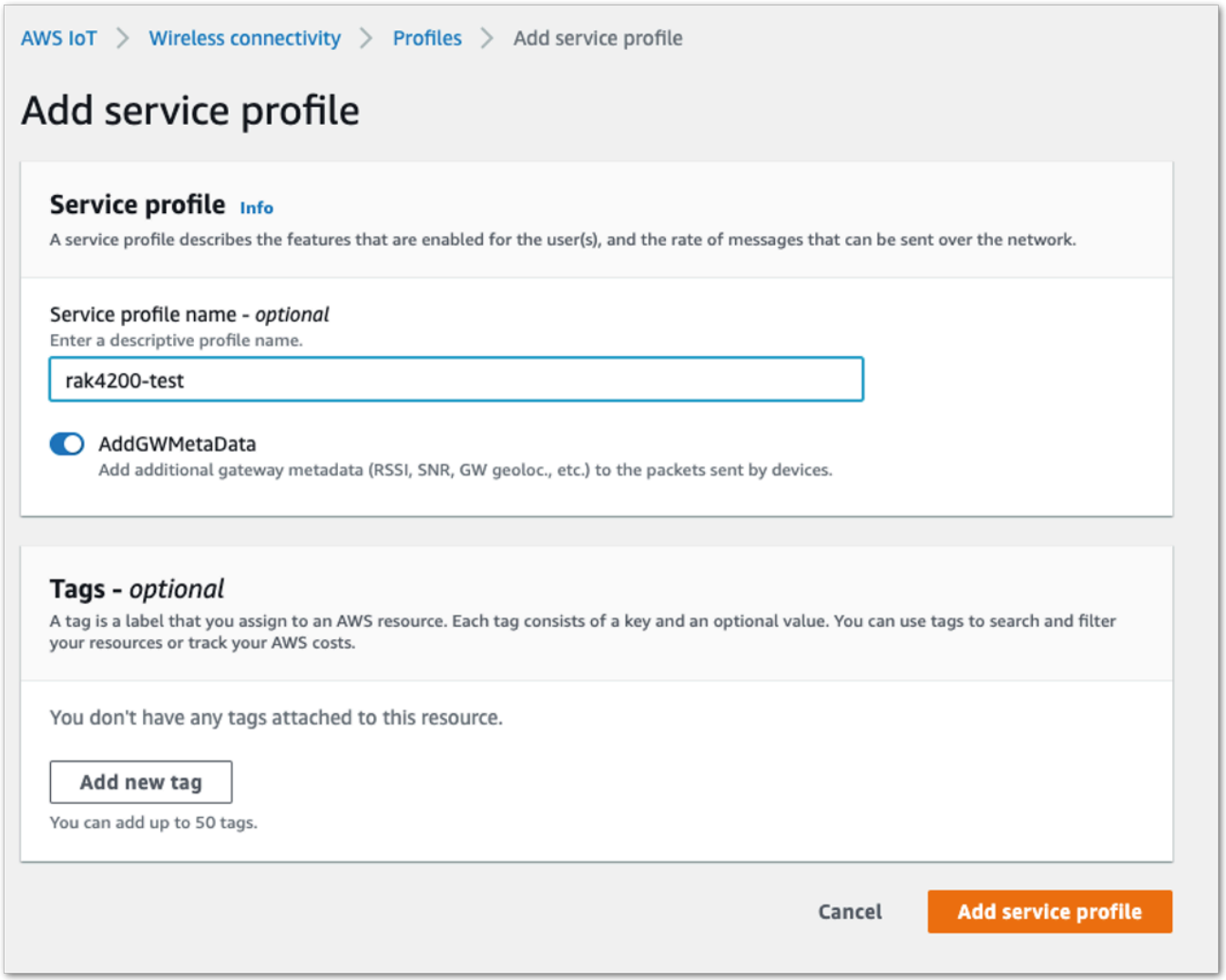


Figure 8: Adding the Service Profile

3. Add Destination.

Before adding the destination, follow the Add IAM role for Destination to AWS IoT Core for LoRaWAN section to configure IAM policy and role.

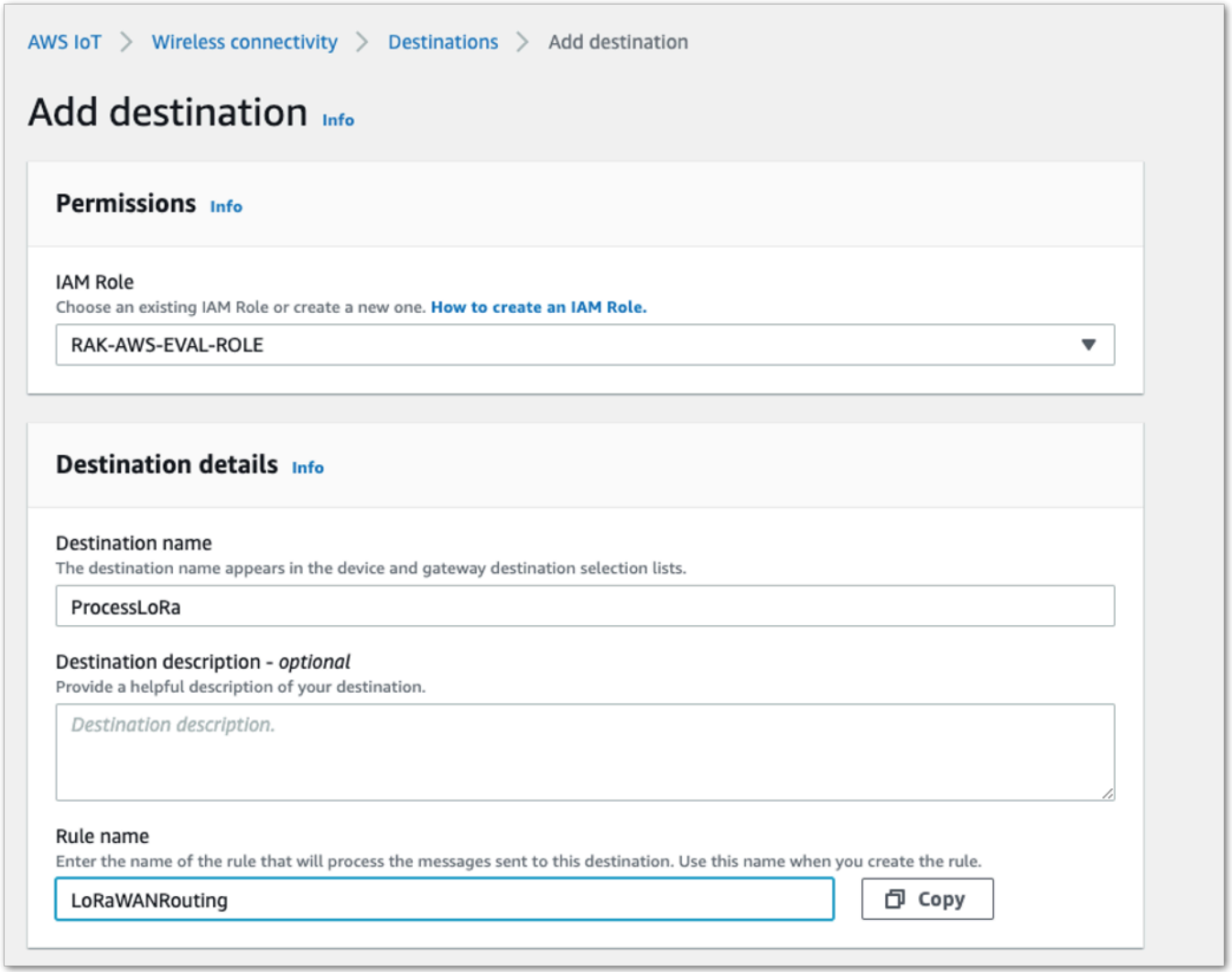


Figure 9: Adding Destination

4. Add Device.

Before adding a device to AWS IoT, retrieve the **DevEui**, **AppEui**, and **AppKey** from the end Device's console. You can use AT command `at+get_config=lora:status` to obtain the information.

For more AT commands, refer to the **RAK4200 AT Command Manual**.

```
at+get_config=lora:status\r\n
OK Work Mode: LoRaWAN
Region: EU868
Send_interval: 600s
Auto send status: false.
MulticastEnable: true.
Multi_Dev_Addr: 260111FD
Multi_Apps_Key: F13DDFA2619B10411F02F042E1C0F356
Multi_Nwks_Key: 1D1991F5377C675879C39B6908D437A6
Join_mode: OTAA
DevEui: 0000000000000888
AppEui: 0000000000000888
AppKey: 00000000000008880000000000000888
Class: C
Joined Network:false
IsConfirm: unconfirm
AdrEnable: true
EnableRepeaterSupport: false
RX2_CHANNEL_FREQUENCY: 869525000, RX2_CHANNEL_DR:0
RX_WINDOW_DURATION: 3000ms
RECEIVE_DELAY_1: 1000ms
RECEIVE_DELAY_2: 2000ms
JOIN_ACCEPT_DELAY_1: 5000ms
JOIN_ACCEPT_DELAY_2: 6000ms
Current Datarate: 4
Primeval Datarate: 4
ChannelsTxPower: 0
UpLinkCounter: 0
DownLinkCounter: 0
```

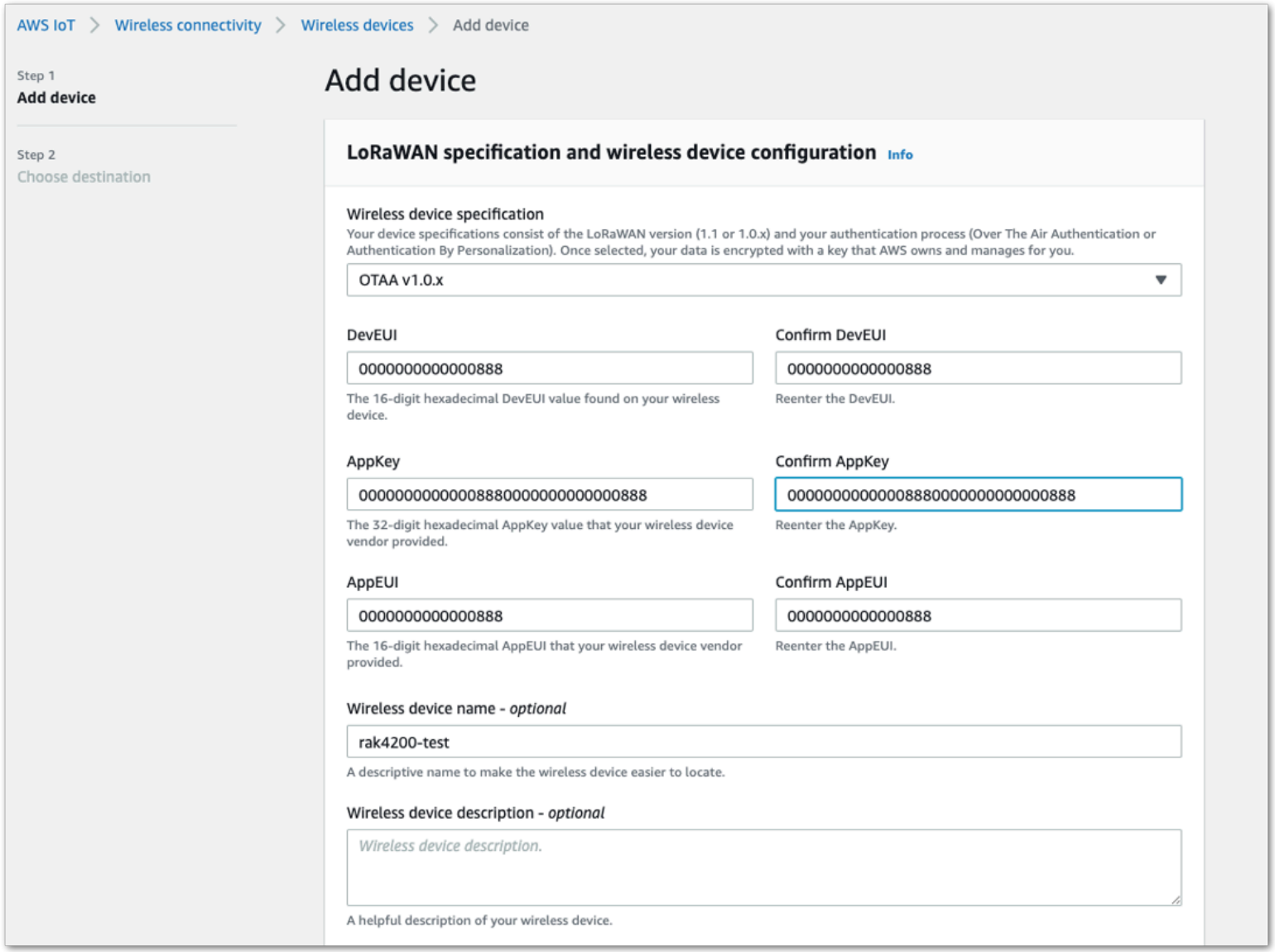


Figure 10: LoRaWAN specifications and wireless device configuration

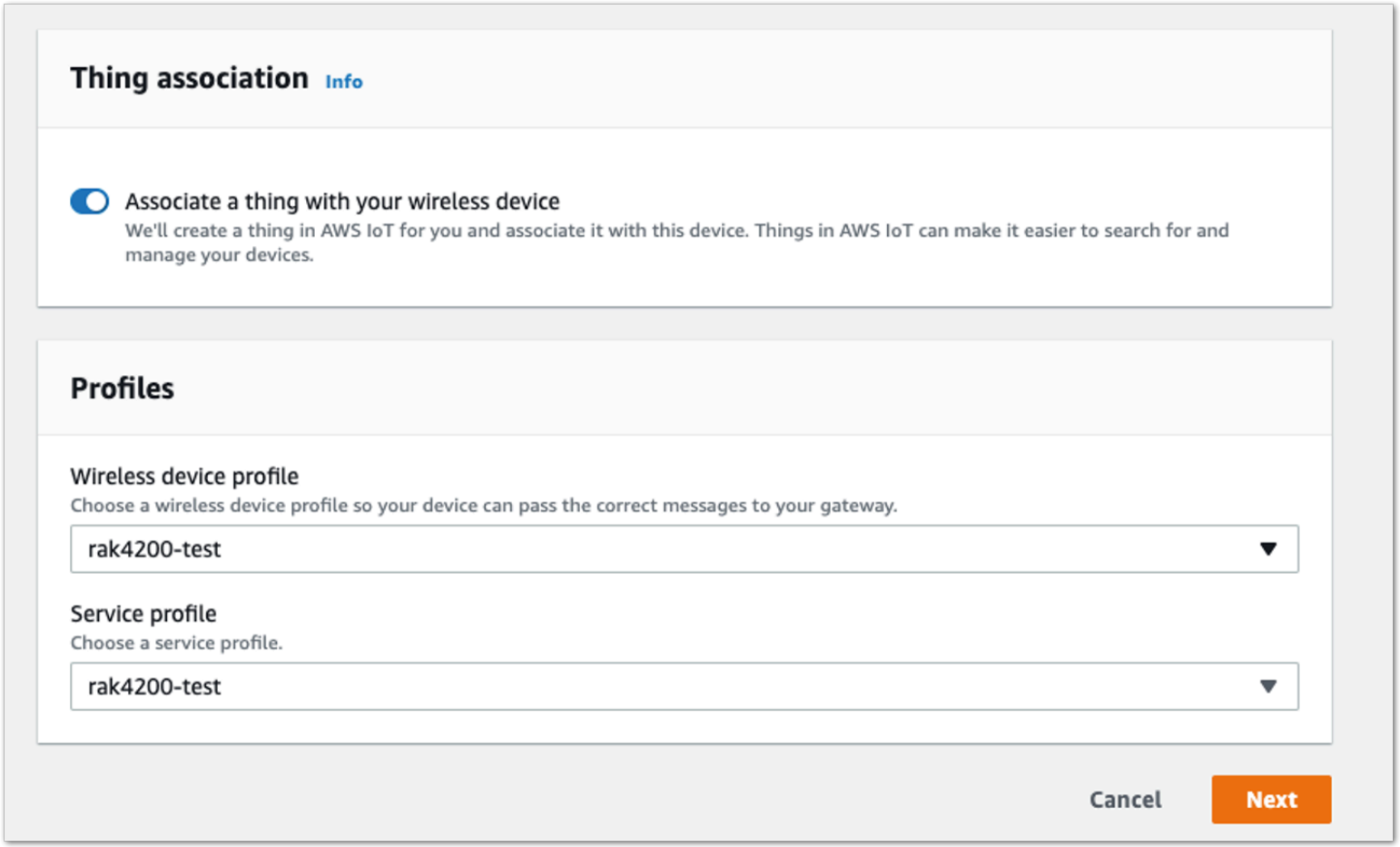


Figure 11: Choosing a Wireless Device Profile

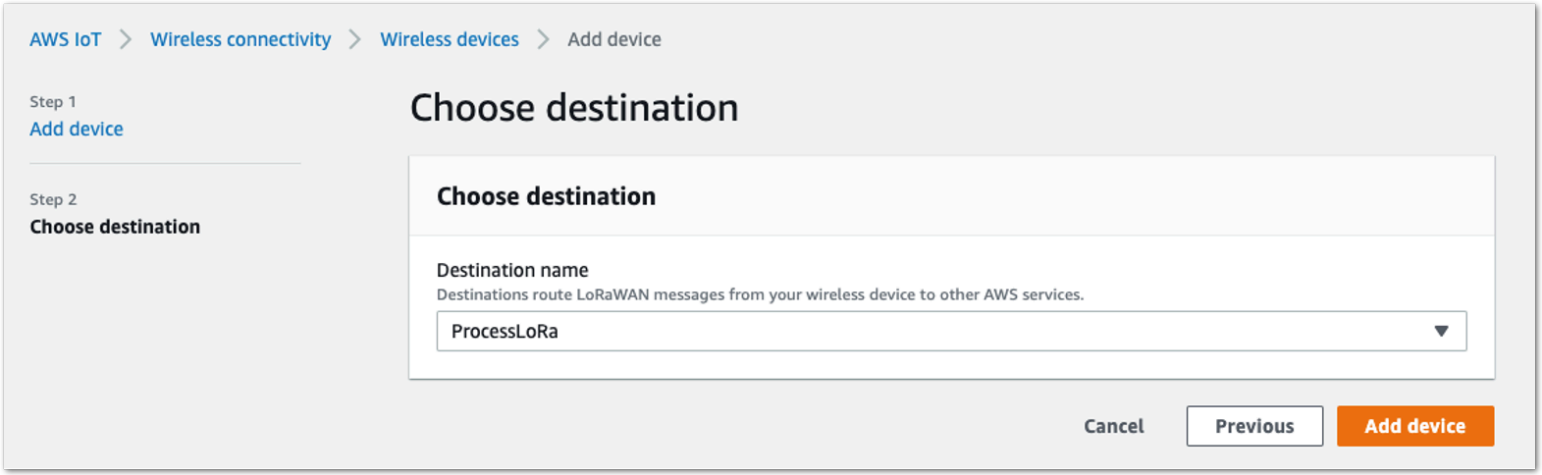


Figure 12: Choosing a Destination

5. Restart the end Device, and it should join the AWS IoT LoRaWAN server.

```
EVENT:0:STARTUP
SYSLOG:4:OTAA Join Request
SYSLOG:4:OTAA Join Success
EVENT:1:JOIN_NETWORK
SYSLOG:4:LoRa Tx :
```