



**TEKTELIC**  
communications  
— IoT for life —

## TEKTELIC CORE NS to AWS IoT Integration



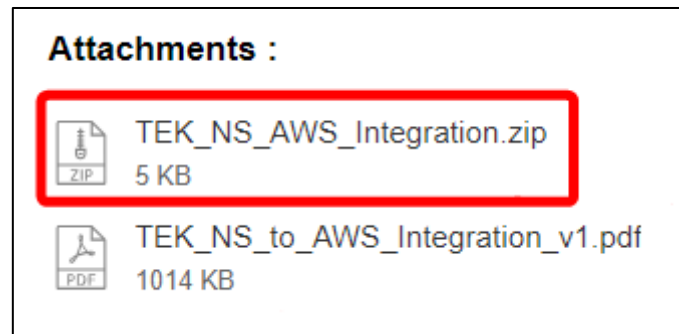
---

# Introduction

- Reference guide to connect TEKTELIC CORE NS to AWS IoT Core through Integration.
- Requirements:
  1. AWS IoT Core account
  2. TEKTELIC CORE NS account
  3. Scripts and miscellaneous files
  4. Latest version of Python 3
  5. Code editor to install required Python packages
- Table of Contents
  1. [Prerequisites](#)
  2. [AWS IoT Core Setup](#)
  3. [TEKTELIC CORE NS Setup](#)

# Prerequisites

***You are required to download files found in the [article of this guide](#).***



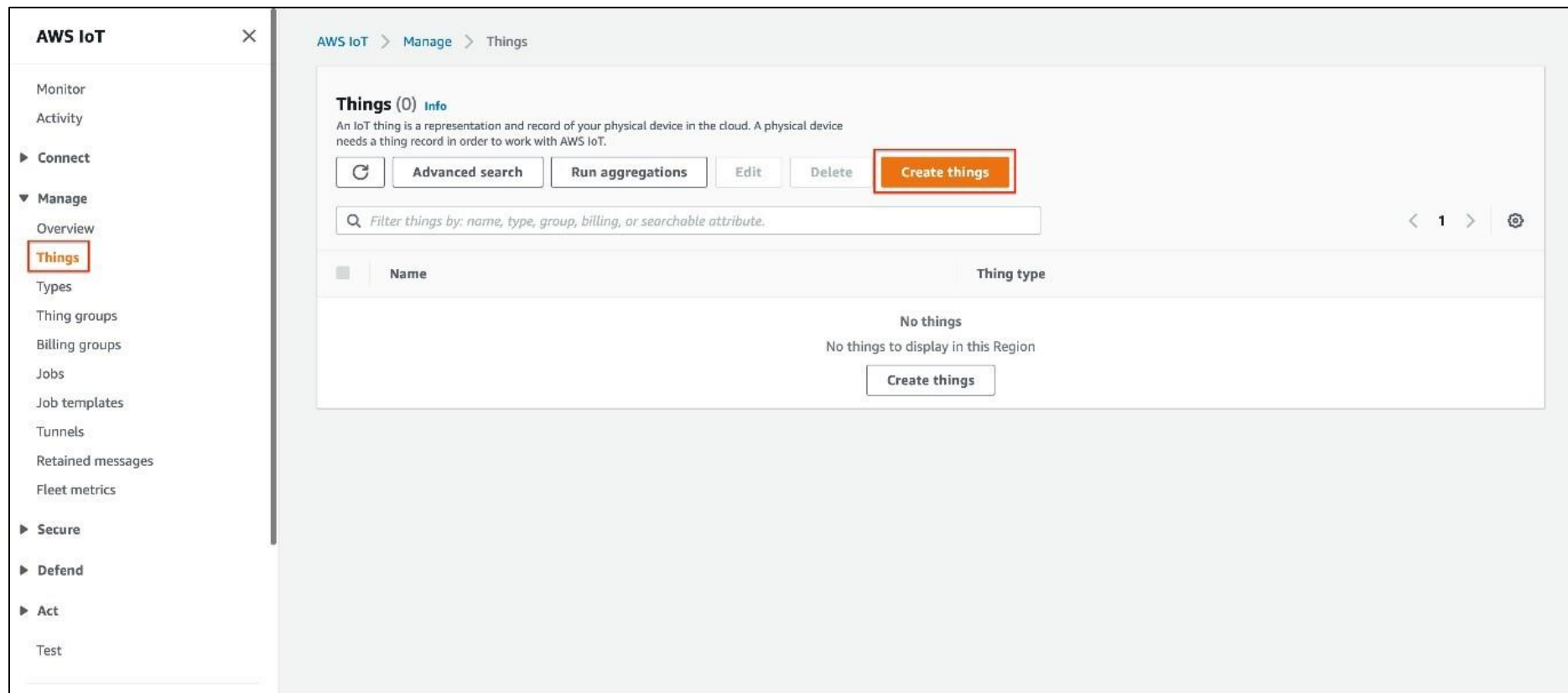
Please ensure that you have the following accounts prepared:

- AWS IoT Core
- TEKTELIC CORE Network Server

If you do not have a TEKTELIC CORE Network Server account, [please raise a ticket on the support portal](#).

# AWS IoT Core Setup

1. Navigate to AWS IoT Core / Manage / Things and select **Create Thing**



# AWS IoT Core Setup (cont.)

## 2. Select Create Single Thing

AWS IoT > Manage > Things > Create things

### Create things [Info](#)

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

**Number of things to create**

- Create single thing**  
Create a thing resource to register a device. Provision the certificate and policy necessary to allow the device to connect to AWS IoT.
- Create many things**  
Create a task that creates multiple thing resources to register devices and provision the resources those devices require to connect to AWS IoT.

Cancel **Next**

# AWS IoT Core Setup (cont.)

3. Choose a name for AWS IoT Thing. The purpose is to give permission for TEKTELIC CORE NS to send uplink data. Then click Next.

resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

Step 2 - optional  
Configure device certificate

Step 3 - optional  
Attach policies to certificate

### Thing properties [info](#)

Thing name

LoraWAN\_NetworkServer

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

### Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

- ▶ Thing type - optional
- ▶ Searchable thing attributes - optional
- ▶ Thing groups - optional
- ▶ Billing group - optional

### Device Shadow [Info](#)

Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

- No shadow
- Named shadow  
Create multiple shadows with different names to manage access to properties, and logically group your devices properties.
- Unnamed shadow (classic)  
A thing can have only one unnamed shadow.

Cancel Next

# AWS IoT Core Setup (cont.)

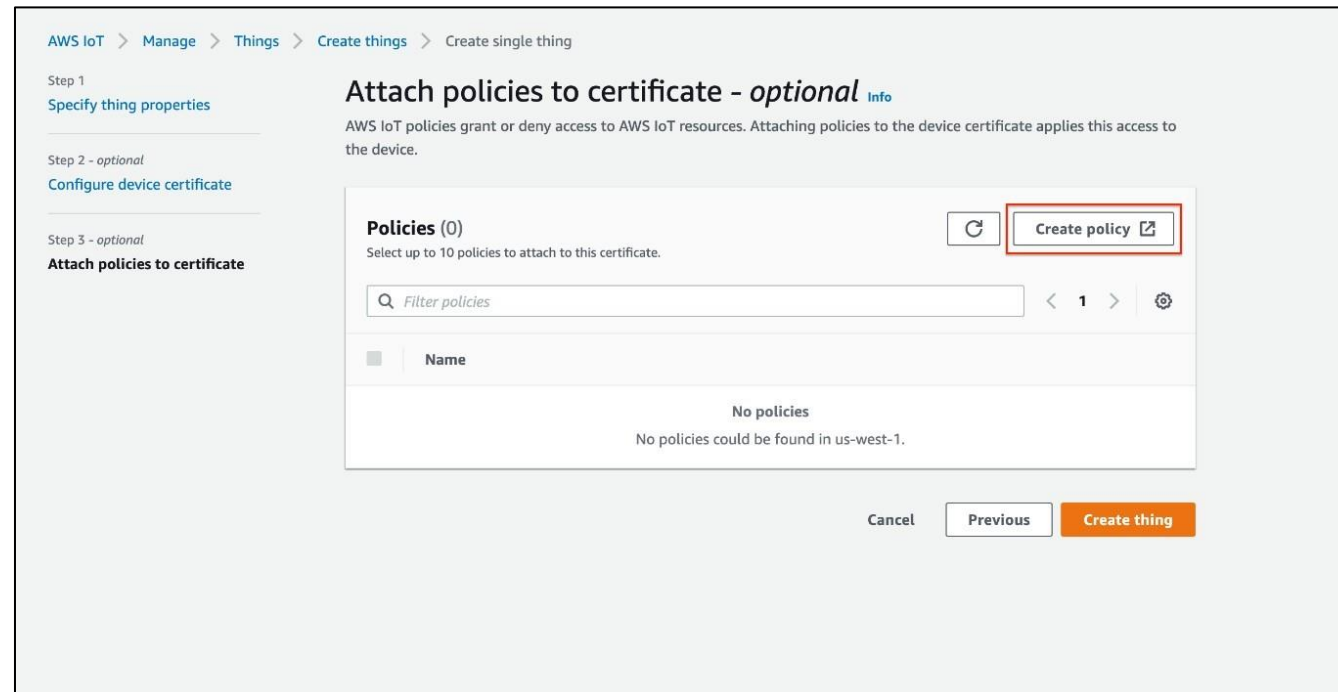
3. Select **Auto-generate a new certificate**. This option is selected by default. Then click **Next**.

The screenshot shows the AWS IoT Core console interface for configuring a device certificate. The breadcrumb trail is: AWS IoT > Manage > Things > Create things > Create single thing. The page is divided into three steps: Step 1 (Specify thing properties), Step 2 (optional, Configure device certificate), and Step 3 (optional, Attach policies to certificate). The main heading is 'Configure device certificate - optional' with an 'Info' link. Below the heading is a descriptive paragraph: 'A device requires a certificate to connect to AWS IoT. You can choose how you to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.' The 'Device certificate' section contains three radio button options: 'Auto-generate a new certificate (recommended)' (selected and highlighted with a red box), 'Use my certificate', and 'Upload CSR'. Below these is a fourth option: 'Skip creating a certificate at this time'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next' (highlighted with a red box).

# AWS IoT Core Setup (cont.)

4. Create a Policy using the JSON policy template in the attachments (**policy\_template.txt**):

**Note:** Replace region with the correct region name (e.g. us-west-2) and account number (12-digits).  
e.g.: "arn:aws:iot:**us-west-2**:**123456789102**:topic/\*"





# AWS IoT Core Setup (cont.)

5. Select the newly created Policy and finalize by selecting **Create Thing**

The screenshot shows the AWS IoT console interface for creating a thing. The breadcrumb trail is: AWS IoT > Manage > Things > Create things > Create single thing. The left sidebar shows three steps: Step 1: Specify thing properties, Step 2 (optional): Configure device certificate, and Step 3 (optional): Attach policies to certificate. The main content area is titled 'Attach policies to certificate - optional' with an info icon. Below the title is a description: 'AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.' The 'Policies (1/1)' section includes a search bar for 'Filter policies', a refresh button, and a 'Create policy' button. A table lists the selected policy:

<input checked="" type="checkbox"/>	Name
<input checked="" type="checkbox"/>	lotCorePolicy

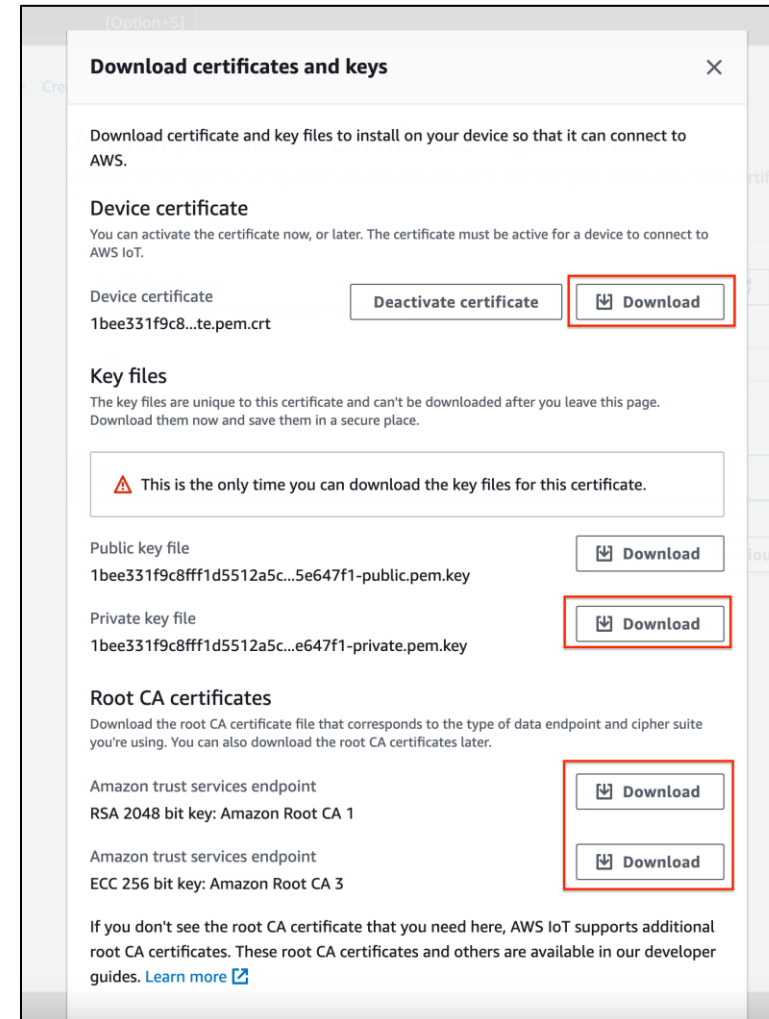
At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create thing' (highlighted with a red border).

# AWS IoT Core Setup (cont.)

6. Download the certificates and keys to allow TEKTELIC CORE NS to connect to AWS IoT.

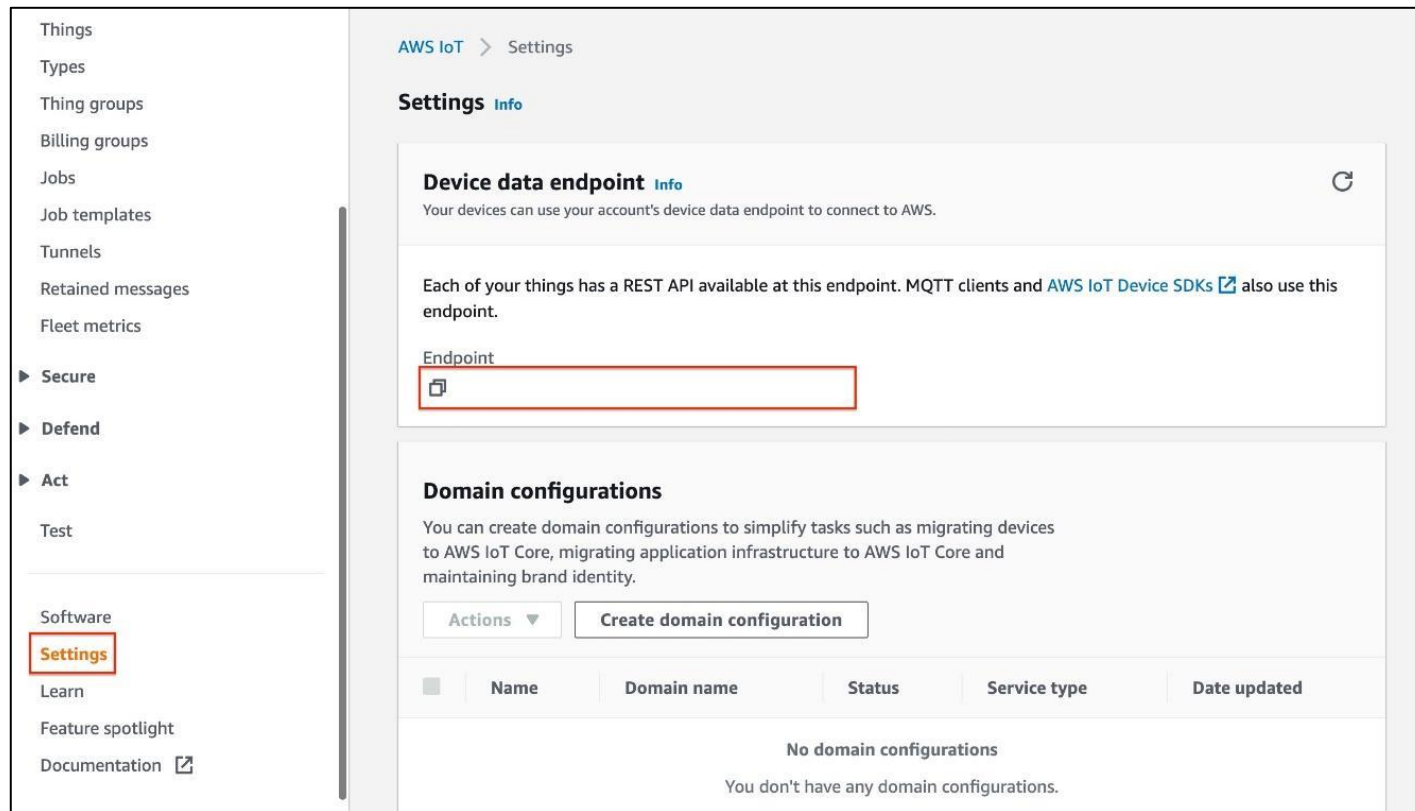
Files required:

- Device Certificate
- Private Key File
- CA, CA3 Root CA Certificates (this will be merged into a single file later).



# AWS IoT Core Setup (cont.)

7. Navigate to the Settings tab in AWS IoT to reveal the **Device Data Endpoint**. This will be used as the hostname in the next section.



The screenshot displays the AWS IoT Core Settings page. On the left, a navigation menu lists various settings categories, with 'Settings' highlighted. The main content area is titled 'Settings' and includes a 'Device data endpoint' section. This section contains a text box for the endpoint, which is currently empty and highlighted with a red border. Below this, there is a 'Domain configurations' section with a 'Create domain configuration' button. At the bottom, a table header is visible, but no data rows are present.

Things  
Types  
Thing groups  
Billing groups  
Jobs  
Job templates  
Tunnels  
Retained messages  
Fleet metrics  
▶ Secure  
▶ Defend  
▶ Act  
Test  
Software  
**Settings**  
Learn  
Feature spotlight  
Documentation

AWS IoT > Settings

**Settings** Info

**Device data endpoint** Info ↻

Your devices can use your account's device data endpoint to connect to AWS.

Each of your things has a REST API available at this endpoint. MQTT clients and [AWS IoT Device SDKs](#) also use this endpoint.

Endpoint

**Domain configurations**

You can create domain configurations to simplify tasks such as migrating devices to AWS IoT Core, migrating application infrastructure to AWS IoT Core and maintaining brand identity.

Actions ▼

	Name	Domain name	Status	Service type	Date updated
No domain configurations You don't have any domain configurations.					

# TEKTELIC CORE NS Setup

**NOTE:** Creating an Integration with x509 certificate is available through API only.

**As a reminder, this section will require the following:**

- Latest version of Python 3
- Code editor to install required Python packages

API References will be provided in the attachment “**api\_reference.txt**”.

The script will prompt the user to enter the ***TEKTELIC CORE NS region, username, password, application name, data-converter name, and the AWS Device data endpoint.*** When successful, an MQTT/HTTP integration will be created on TEKTELIC CORE NS.

After the integration is created on TEKTELIC CORE NS, you can subscribe to the topic using the MQTT Test Client on AWS IoT to view the uplink data from the devices.

# TEKTELIC CORE NS Setup (cont.)

1. Install the required packages in the code editor: **pip install requests**
2. Insert certificates into the script.
  - a. Combine Amazon Root CA1.pem and CA3.pem and copy it into the “**caCertificate**” field.  
-----BEGIN CERTIFICATE-----  
<contents of CA1.pem>  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
<contents of CA3.pem>  
-----END CERTIFICATE-----
  - b. Copy the contents of the “**-certificate.pem**” file key to the “**clientCertificate**” field.
  - c. Copy the contents of the “**-private.pem**” file key to the “**clientPrivateKey**” field.
3. Execute the script.
  - a. Navigate to the folder of the script.
  - b. Execute the script with the following command: **python <mqtt/http>\_aws\_integration.py**

Best-In-Class, Carrier Grade &  
Most Cost Effective  
Portfolio of Gateways, Network Server,  
Sensors & Applications