

Getting Started Guide for AWS IoT Core: Intel® NUC

Table of Contents

1	<i>Document Information</i>	1
2	<i>Overview</i>	1
3	<i>Hardware Description</i>	2
4	<i>Set up your Development Environment</i>	2
5	<i>Set up your hardware</i>	2
6	<i>Setup your AWS account and Permissions</i>	2
7	<i>Create Resources in AWS IoT</i>	3
8	<i>Provision the Device with credentials</i>	3
9	<i>Build/Run the demo</i>	4
10	<i>Debugging</i>	4
11	<i>Troubleshooting</i>	4

1 Document Information

1.1 Revision History

Version	Date	Description
1.0	August 2022	Publish Document
2.0	November 2022	Publish Document
3.0	April 2023	Publish Document
4.0	July 2023	Publish Document

2 Overview

Intel® NUC is a small form factor PC that allows you to customize your mini PC experience to fit a wide range of use cases. This document describes how to set up AWS IoT Core on an Intel® NUC device running Ubuntu/Windows. At the end of this guide, the Intel® NUC can connect seamlessly with AWS IoT Core and you can easily integrate Intel® NUC into your IoT solution stack.

3 Hardware Description

3.1 Datasheet

Tested NUC models are listed below. Please identify your NUC product and search for the SKUs on the provided link.

Product Name	Datasheet URL
Tiger Canyon	Tiger Canyon Products Datasheet
Elk Bay	Elk Bay Products Datasheet
Atlas Canyon	Atlas Canyon Products Datasheet
Wall Street Canyon	Wall Street Canyon Products Datasheet
Hard Bay	Hard Bay Products Datasheet
Arena Canyon	Arena Canyon Products Datasheet
Serpent Canyon	Serpent Canyon Products Datasheet

3.2 Additional Hardware References

Please refer to the [Intel® NUC device](#) page and select your NUC model for more product details.

4 Set up your Development Environment

4.1 Tools Installation (IDEs, Toolchains, SDKs)

You can use any available tools for Intel® NUC devices based on your chosen operating system.

For demo purposes, AWS IoT Device SDK for Python will be used. Please refer to the [demo](#) section for more details.

5 Set up your hardware

Please refer to [User Guide for Intel® NUC](#) and [Operating System Installation for Intel® NUC](#) to setup Intel® NUC device.

6 Setup your AWS account and Permissions

Refer to the online AWS documentation at [Set up your AWS Account](#). Follow the steps outlined in the sections below to create your account and a user and get started:

- [Sign up for an AWS account](#) and
- [Create a user and grant permissions](#)
- [Open the AWS IoT console](#)

Pay special attention to the Notes.

7 Create Resources in AWS IoT

Refer to the online AWS documentation at [Create AWS IoT Resources](#). Follow the steps outlined in these sections to provision resources for your device:

- [Create an AWS IoT Policy](#)
- [Create a thing object](#)

Pay special attention to the Notes.

8 Provision the Device with credentials

Intel NUC does not come with unique client certificate. In a production environment, devices need to be provisioned by the user using a service called fleet provisioning. Refer to [Provisioning devices that don't have device certificates using fleet provisioning](#) for the instructions.

Note: All the required credentials for device provisioning are created from step 6 & step 7

Follow the steps below to use a sample app as AWS IoT device for development purpose, you will need Git and Python installed on your Intel NUC device before proceeding.

1. Install AWS IoT Device SDK for Python
`python3 -m pip install awsiotsdk`
2. Clone the AWS IoT Device SDK for Python repository
`git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git`
3. Create the `certs` directory. Into the `certs` directory, copy the private key, device certificate, and root CA certificate files you saved when you created and registered the thing object in [Create AWS IoT resources](#). The file names of each file in the destination directory should match those in the table.

File	File path
Private key	<code>~/certs/private.pem.key</code>
Device certificate	<code>~/certs/device.pem.crt</code>
Root CA certificate	<code>~/certs/Amazon-root-CA-1.pem</code>

4. Get your AWS IoT endpoint. Choose setting in the left menu of [AWS IoT Console](#) , your endpoint is displayed in the Device data endpoint section of the setting page.

The endpoint value has a format of: endpoint_id-ats.iot.region.amazonaws.com, for example, a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com.

5. Navigate to the [aws-iot-device-sdk-python-v2/samples/node/pub_sub](#) directory that the SDK created.
6. Replace your-iot-endpoint as indicated and run the sample code.

```
python3 pubsub.py --endpoint your-iot-endpoint --ca_file ~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key
```

You can refer to [Use your Windows or Linux PC or Mac as an AWS IoT device](#) for details instruction and explanation.

9 Build/Run the demo

To test device connectivity to AWS IoT Core, please follow the steps outlined [here](#).

10 Debugging

Select appropriate product link and search for your Intel NUC product SKUs on the page to get debugging info and support.

Product Name	Debugging Documentation
Tiger Canyon	Support for Intel® NUC Kit with 11th Generation Intel® Core™
Elk Bay	Support for Intel® NUC Compute Elements
Atlas Canyon	Support for Intel® NUC Kit with Intel® Pentium® Processors
Wall Street Canyon	Support for Intel® NUC Kit with 12th Generation Intel® Core™
Hard Bay	Support for Intel® NUC Compute Elements
Arena Canyon	Support for Intel® NUC Kit with 13th Generation Intel® Core™
Serpent Canyon	Support for Intel® NUC Kit with 12th Generation Intel® Core™

11 Troubleshooting

Refer to [Intel® NUC support page](#) for common device troubleshooting tips.

Refer to [Troubleshooting AWS IoT Core](#) for AWS IoT Core troubleshooting tips.