

WEB-AIC

MAY 17 2020

MANISH KOTNI

# THE INTERNET OF THINGS WITH AWS IOT CORE

Understanding its Impact and Applications

LIVING OUT LOUD

# WEBINAR HIGHLIGHTS

## FOCUS AREAS

- Defining the IoT
- MQTT Protocol
- AWS IOT Core
- Creation a Thing In AWS iot core
- Converting AWS IoT Core credential from .pem to .der format
- Arduino Sketch
- Uploading Sketchs to Node MCU
- Subsrciptioin of a Thing in AWS IOT
- Live Demo With AWS IOT Core

## A NEW AGE OF INTERCONNECTIVITY

# DEFINING THE IOT

The Internet of things (IoT) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

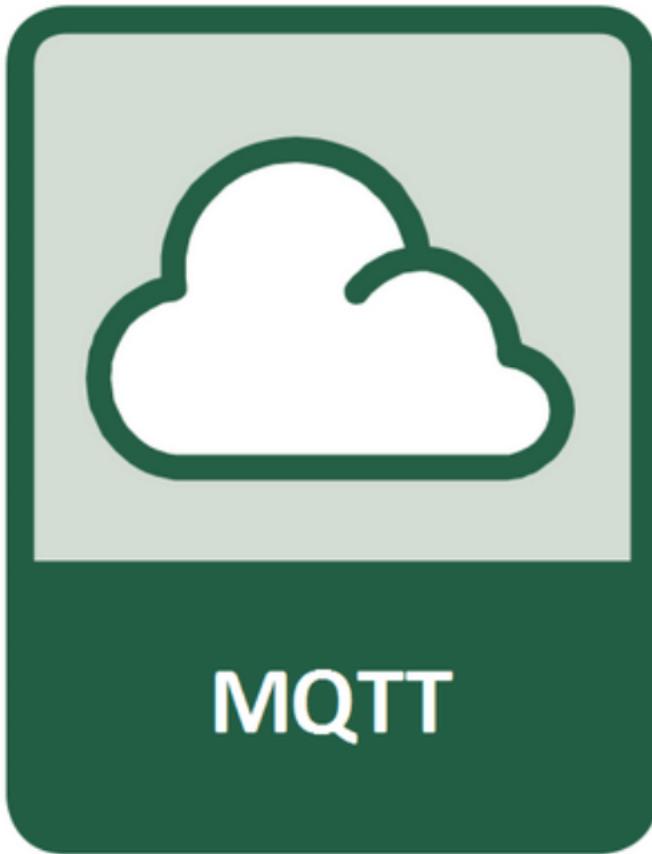


# MQTT PROTOCOL

## MESSAGE QUEUING TELEMETRY TRANSPORT

is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. For example, it has been used in sensors communicating to a broker via satellite link, over occasional dial-up connections with healthcare providers, and in a range of home automation and small device scenarios. It is also ideal for mobile applications because of its small size, low power usage, minimised data packets, and efficient distribution of information to one or many receivers

WEB-API || ינואר 2020



## ARDUINO IDE

A Basic Command on C Language

## NODE MCU / ESP8266

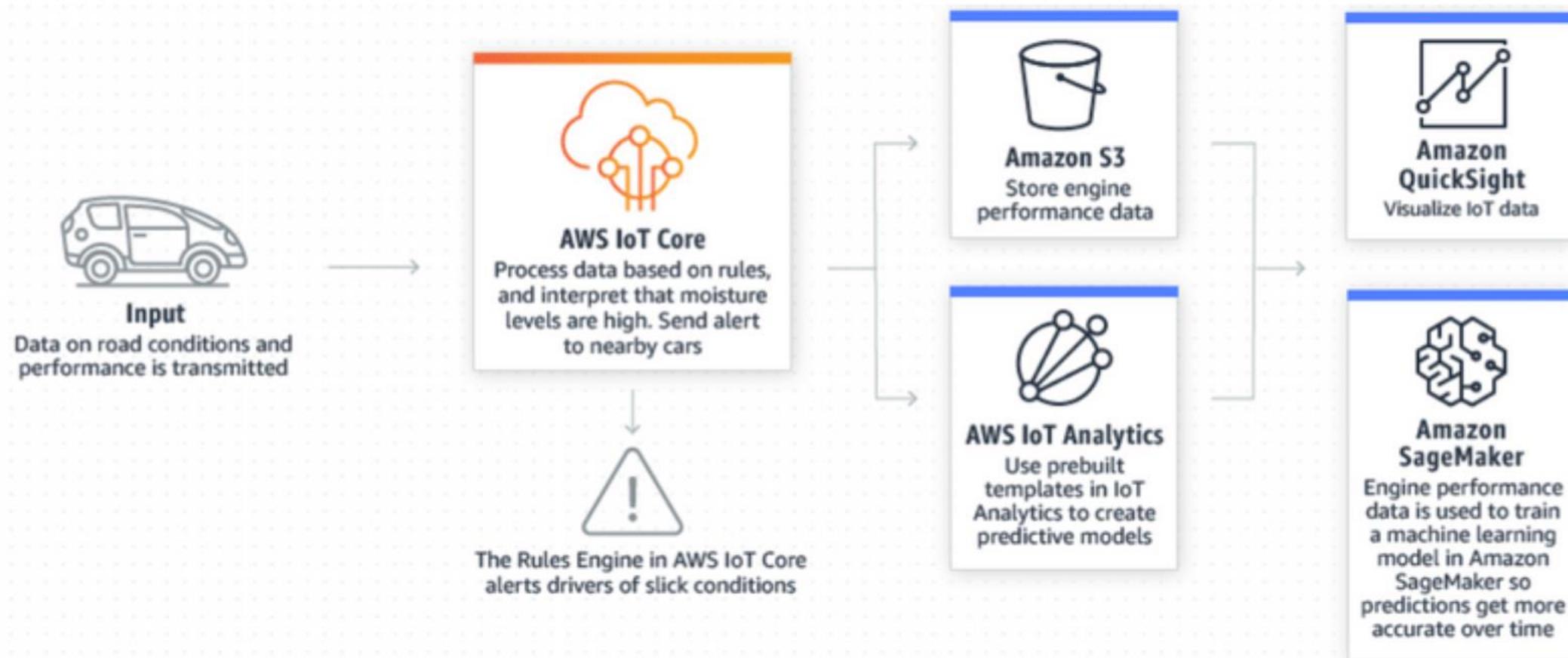
Some Info on the Micro Controllers

## IOT

Some knowledge and command over the IOT

## Prerequisites

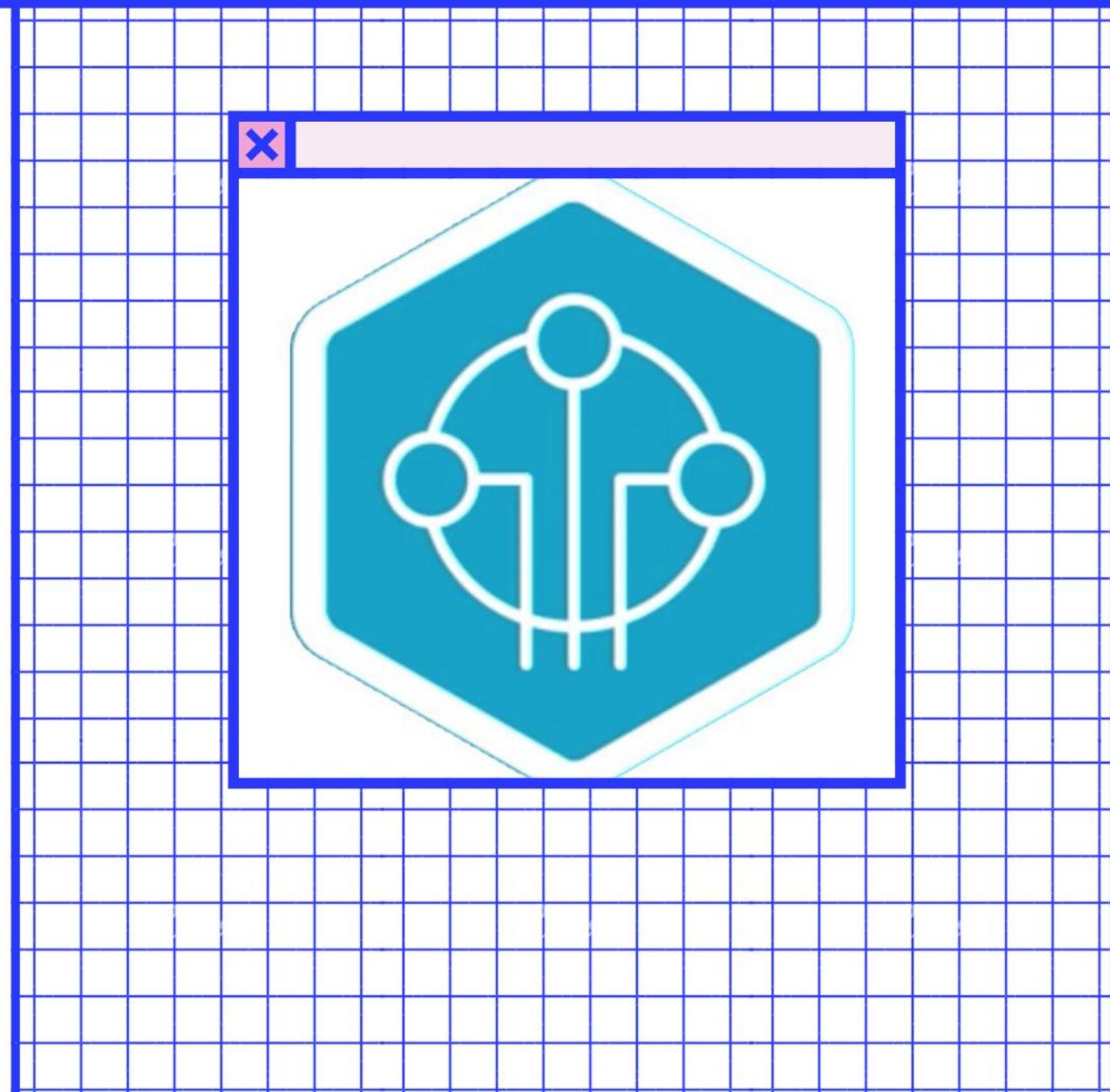
# WHY AWS IOT CORE ?



# OVERVIEW

- 1.Creating a Thing in the AWS IoT Core, generating a certificate and attaching a policy to it.
- 2.Converting AWS IoT Core credential(Certificate, Private Key, Root CA) from .pem to .der format
- 3.DER & PEM Info
- 4.Installing OpenSSL on Windows
- 5.Installing ESP8266 sketch data upload tool in Arduino IDE
- 6.Arduino sketch and modifications according to the thing.
- 7.Uploading AWS certificates & code to the NodeMCU ESP8266
- 8.Testing/Subscription of thing on Amazon Web Services(AWS IoT Core).
- 9.Results& Data Logging.

# Getting Started WITH AWS IOT CORE



This screenshot shows the AWS IoT Monitor dashboard. On the left, a sidebar menu includes options like Monitor, Onboard, Manage, Greengrass, Secure, Defend, Act, Test, Software, Settings, and Learn. The main area is titled 'Monitor' and contains two sections: 'Successful connections' and 'Messages'. Both sections show 'No data' at the moment. At the bottom, there's a feedback link, language selection (English (US)), a search bar, and system status indicators.

## FIRST VIEW OF AWS IOT CORE

TO REGITER A THING

This screenshot shows the AWS IoT Things dashboard. It features a search bar, a 'Fleet Indexing' button, and a 'Create' button. Below the search bar, two things are listed: 'Holo\_NodeMCU' and 'Hologram\_NodeMCU', both categorized under 'NO TYPE'. The dashboard also includes a sidebar with 'Resource Groups' and other navigation links. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows a web-based interface for creating AWS IoT things. At the top, there's a teal header bar with the title "Creating AWS IoT things". Below it, a sub-header explains that an IoT thing is a representation and record of a physical device in the cloud. It includes a link to "Learn more".  
  
The main content area has two main sections:

- Register a single AWS IoT thing:** This section allows creating a thing in your registry. It features a large teal button labeled "Create a single thing".
- Bulk register many AWS IoT things:** This section allows creating things in your registry for a large number of devices already using AWS IoT, or registering devices so they are ready to connect to AWS IoT. It also features a teal button labeled "Create many things".

At the bottom left, there's a "Cancel" link, and at the bottom right, another teal button labeled "Create a single thing".

# CREATING A THING

eg Node MCU, ESP 8266, R-Pi Etc..

LIVING OUT LOUD

## Add your device to the thing registry

This step creates an entry in the thing registry and a thing shadow for your device.

Name

Demo\_d

1

### Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type

No type selected

Create a type

### Set searchable thing attributes (optional)

Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key

Provide an attribute key, e.g. Manufacturer

Value

Provide an attribute value, e.g. Acme-Corporation

Clear

Add another

Show thing shadow ▾

Cancel

Back

Next

2

CREATE A THING

## Add a certificate for your thing

STEP  
2/3

A certificate is used to authenticate your device's connection to AWS IoT.

### One-click certificate creation (recommended)

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

[Create certificate](#)

### Create with CSR

Upload your own certificate signing request (CSR) based on a private key you own.

[Create with CSR](#)

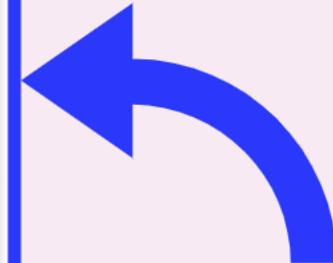
### Use my certificate

Register your CA certificate and use your own certificates for one or many devices.

[Get started](#)

### Skip certificate and create thing

You will need to add a certificate to your thing later before your device can connect to AWS IoT.

[Create thing without certificate](#)

# Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	5210d8f36d.cert.pem	<a href="#">Download</a>
A public key	5210d8f36d.public.key	<a href="#">Download</a>
A private key	5210d8f36d.private.key	<a href="#">Download</a>



You also need to download a root CA for AWS IoT:

A root CA for AWS IoT [Download](#)

[Activate](#)

# CA certificates for server authentication

Depending on which type of data endpoint you are using and which cipher suite you have negotiated, AWS IoT Core server authentication certificates are signed by one of the following root CA certificates:

## VeriSign Endpoints (legacy)

- RSA 2048 bit key: [VeriSign Class 3 Public Primary G5 root CA certificate](#)

## Amazon Trust Services Endpoints (preferred)

- RSA 2048 bit key: [Amazon Root CA 1](#). 
- RSA 4096 bit key: Amazon Root CA 2. Reserved for future use.
- ECC 256 bit key: [Amazon Root CA 3](#).
- ECC 384 bit key: Amazon Root CA 4. Reserved for future use.

These certificates are all cross-signed by the [Starfield Root CA Certificate](#). All new AWS IoT Core regions, beginning with the May 9, 2018 launch of AWS IoT Core in the Asia Pacific (Mumbai) Region, serve only ATS certificates.

Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	5210d8f36d.cert.pem	<a href="#">Download</a>
A public key	5210d8f36d.public.key	<a href="#">Download</a>
A private key	5210d8f36d.private.key	<a href="#">Download</a>

You also need to download a root CA for AWS IoT:

A root CA for AWS IoT [Download](#)

Activate



Deactivate

1

Cancel

2



Done

Attach a policy



# Things

Search things



Fleet Indexing



Holo\_NodeMCU  
NO TYPE



Hologram\_NodeMCU  
NO TYPE



Demo\_0  
NO TYPE





Monitor

Onboard

**Manage**

Things

Types

Thing groups

Billing Groups

Jobs

Tunnels

Greengrass

Secure

Defend

Act

Test

Software

Settings

Learn

Monitor

Onboard

Manage

Greengrass

**Secure**

Certificates

Policies

CAs

Role Aliases

Authorizers

Defend

Act

Test

Software

Settings

**Policies**

Search policies

**UTILIZE****Create****Card**

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name  
Demo\_policy

Add statements

Policy statements define the types of actions that can be performed by a resource.

Action  
\*

Resource ARN  
\*

Effect  
 Allow  Deny

Remove

Add statement

• • •

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name  
Demo\_policy

Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced mode

Action  
\*

Resource ARN  
\*

Effect  
 Allow  Deny

Remove

Add statement

 Create

add \* on Actions & Resource ARN  
to utilize all platforms of IOT core

The screenshot shows the AWS IoT Secure Certificates page. On the left, a sidebar menu lists various services: Monitor, Onboard, Manage, Greengrass, Secure (which is expanded), Certificates (selected and highlighted with a blue arrow), Policies, CAs, Role Aliases, Authorizers, Defend, Act, and Test. The main content area displays two certificates: one with ID 5210d8f36d7e5f31e8... and another with ID 75955d4c1. A search bar at the top allows for certificate search. To the right of the certificates is a vertical context menu with options: ... (three dots), Activate, Deactivate, Revoke, Accept transfer, Reject transfer, Revoke transfer, Start transfer, Attach policy, Attach thing, Download, and Delete. The 'Certificates' item in the sidebar has a blue arrow pointing to it, and the context menu has two blue arrows pointing to the 'Delete' option.

- Monitor
- Onboard
- Manage
- Greengrass
- Secure**
  - Certificates** ←
  - Policies
  - CAs
  - Role Aliases
  - Authorizers
- Defend
- Act
- Test

Search certificates

5210d8f36d7e5f31e8...  
ACTIVE

75955d4c1  
ACTIVE

...

Activate

Deactivate

Revoke

Accept transfer

Reject transfer

Revoke transfer

Start transfer

Attach policy

Attach thing

Download

Delete

## Attach policies to certificate(s)

Policies will be attached to the following certificate(s):

75955d4c1bbecaefc16b8f6899a6fe75533bd0239d9fcb6ba92c291bbc63bd3f

Choose one or more policies

<input type="checkbox"/> NODEMCU_Polocy	<a href="#">View</a>
<input checked="" type="checkbox"/> Demo_polocy	<a href="#">View</a>



1 policy selected

[Cancel](#)

[Attach](#)



 5210d8f36d-certificate.pem.crt	16-05-2020 13:08	Security Certificate	2 KB
 5210d8f36d-private.pem.key	16-05-2020 13:08	KEY File	2 KB
 AmazonRootCA1.pem	16-05-2020 13:09	PEM File	2 KB

These all are the certificates  
downloaded from AWS IOT Core

Need to convert From .pem to .der

```
> openssl x509 -in xxxxxxxxxxxx-certificate.pem.crt -out cert.der -outform DER  
> openssl rsa -in xxxxxxxxxxxx-private.pem.key -out private.der -outform DER  
> openssl x509 -in AmazonRootCA1.pem -out ca.der -outform DER
```

Replace "xxxxxxxxxx" with your certificate name and AmazonRootCA1 will remain the same because there is no change

Example:

```
> openssl x509 -in 2b495edf21-certificate.pem.crt -out cert.der -outform DER  
> openssl rsa -in 2b495edf21-private.pem.key -out private.der -outform DER  
> openssl x509 -in AmazonRootCA1.pem -out ca.der -outform DER
```

Name	Date modified	Type	Size
ca.der	16-05-2020 14:40	Security Certificate	1 KB
cert.der	16-05-2020 14:38	Security Certificate	1 KB
private.der	16-05-2020 14:39	Security Certificate	2 KB

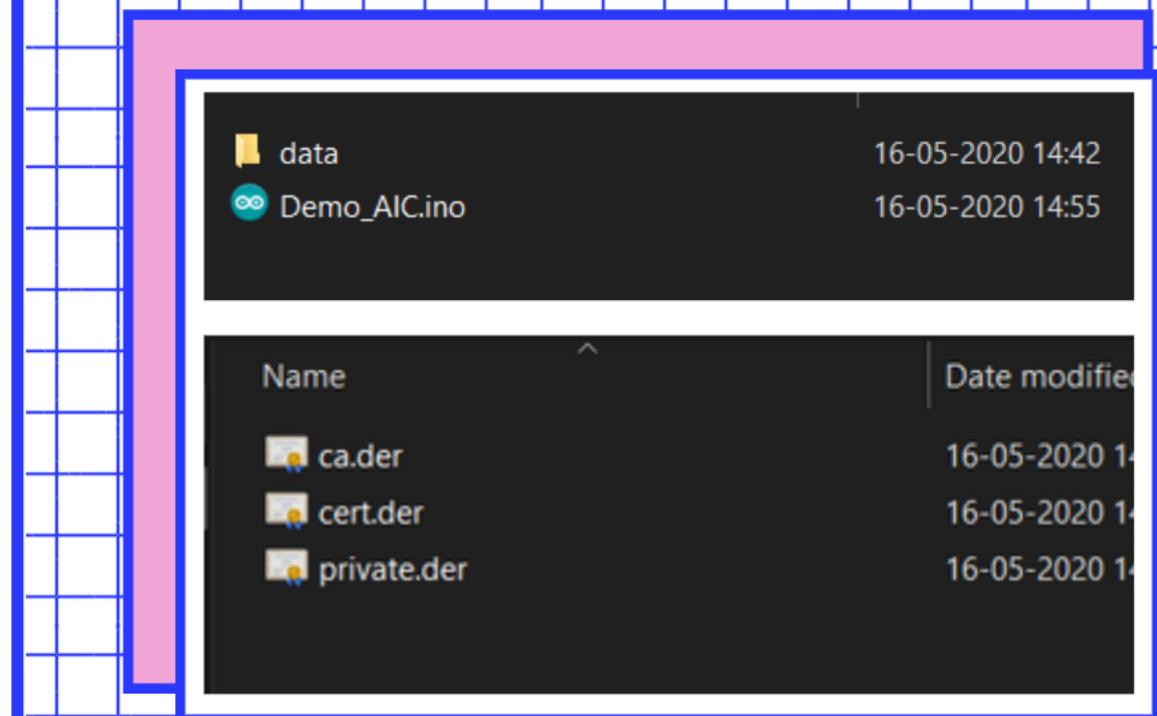
This is how the certificates were stored in a file named data after conversion in to .der format

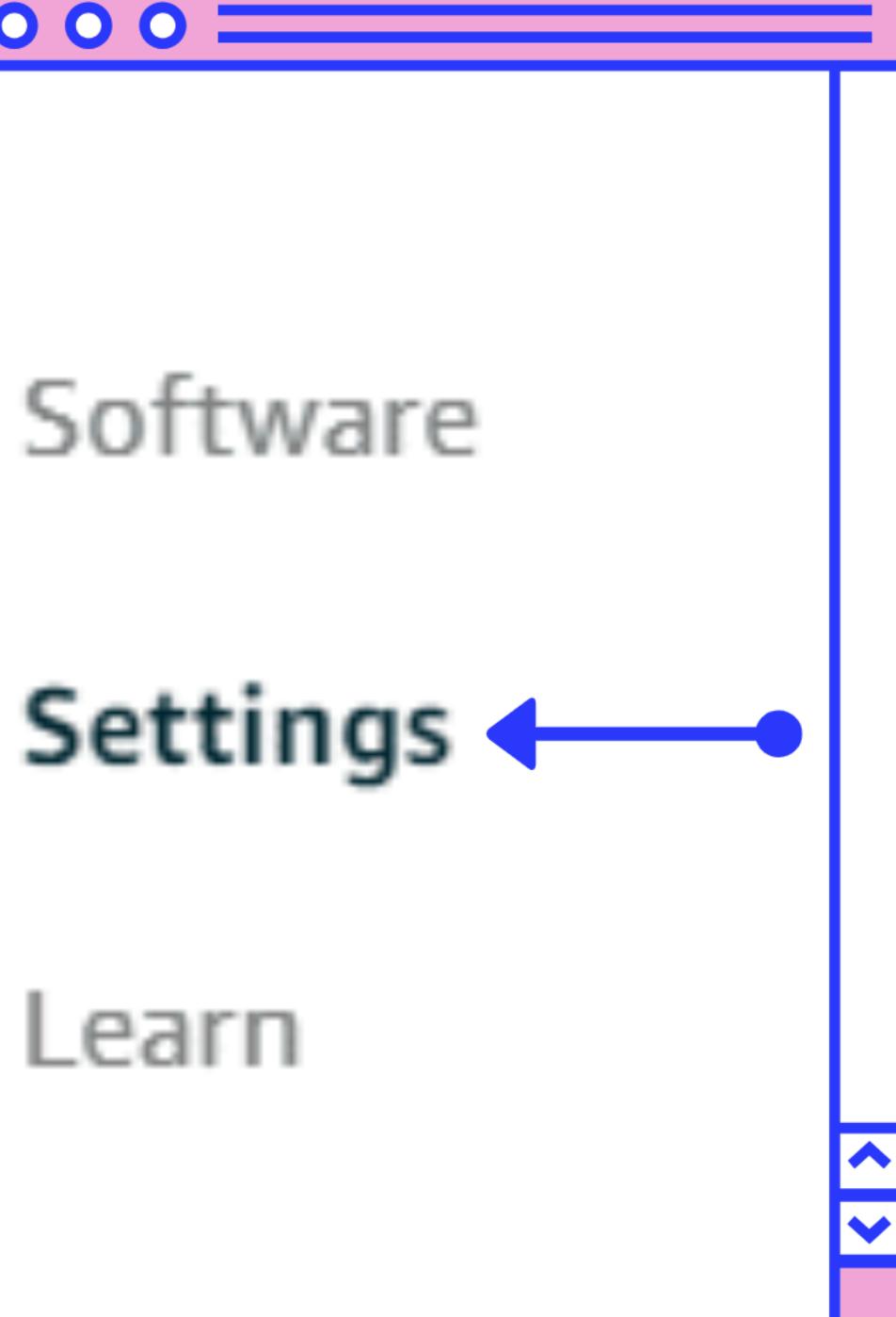
Create a Arduino Sketch and  
write the code for AWS IOT Core

save in a File name it as your  
wish

move previously saved file data  
to the file where the arduino  
sketch is saved

The Above Steps are Crutial





## Settings

### Custom endpoint

ENABLED

This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT [Device SDK](#).

Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.

#### Endpoint

a1uiu4u2y74kw1-ats.iot.eu-west-1.amazonaws.com

copy the end point

```
const char* ssid = "hologram";
const char* password = "09876543";

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

const char* AWS_endpoint = "xxxxxxxxxxxxx-ats.iot.us-west-2.amazonaws.com"; //MQTT broker ip

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
}
```



Monitor

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

Subscriptions

Subscribe to a topic

Publish to a topic

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

Specify a topic to subscribe to, e.g. myTopic/1

Max message capture (?)

100

Subscribe to topic

Quality of Service (?)

0 - This client will not acknowledge to the Device Gateway that messages are received

1 - This client will acknowledge to the Device Gateway that messages are received

MQTT payload display

Auto-format JSON payloads (improves readability)

Display payloads as strings (more accurate)

Display raw payloads (in hexadecimal)



Use "**outTopic**" as a Subscription

## Publish

Specify a topic and a message to publish with a QoS of 0.

[Publish to topic](#)

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

**outTopic**

May 16, 2020 3:27:58 PM +0530

[Export](#) [Hide](#)

```
{
  "message": "DEMO WEB-AIC #157"
}
```

**outTopic**

May 16, 2020 3:27:56 PM +0530

[Export](#) [Hide](#)

```
{
  "message": "DEMO WEB-AIC #156"
}
```



**THANKS FOR  
ATTENDING**

MANISH KOTNI

CERTIFIED IOT ARCHITECT

