



AmebaPro2 Amazon FreeRTOS-LTS - Getting Started Guide



Realtek Semiconductor Corp.

No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan

Tel.: +886-3-578-0211. Fax: +886-3-577-6047

www.realtek.com

COPYRIGHT

©2019 Realtek Semiconductor Corp. All rights reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Realtek Semiconductor Corp.

DISCLAIMER

Please Read Carefully:

Realtek Semiconductor Corp., (Realtek) reserves the right to make corrections, enhancements, improvements and other changes to its products and services. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

Reproduction of significant portions in Realtek data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Realtek is not responsible or liable for such reproduced documentation. Information of third parties may be subject to additional restrictions.

Buyers and others who are developing systems that incorporate Realtek products (collectively, “Customers”) understand and agree that Customers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Customers have full and exclusive responsibility to assure the safety of Customers' applications and compliance of their applications (and of all Realtek products used in or for Customers' applications) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to their applications, Customer has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Customer agrees that prior to using or distributing any applications that include Realtek products, Customer will thoroughly test such applications and the functionality of such Realtek products as used in such applications.

Realtek's provision of technical, application or other design advice, quality characterization, reliability data or other services or information, including, but not limited to, reference designs and materials relating to evaluation kits, (collectively, “Resources”) are intended to assist designers who are developing applications that incorporate Realtek products; by downloading, accessing or using Realtek's Resources in any way, Customer (individually or, if Customer is acting on behalf of a company, Customer's company) agrees to use any particular Realtek Resources solely for this purpose and subject to the terms of this Notice.

Realtek's provision of Realtek Resources does not expand or otherwise alter Realtek's applicable published warranties or warranty disclaimers for Realtek's products, and no additional obligations or liabilities arise from Realtek providing such Realtek Resources. Realtek reserves the right to make corrections, enhancements, improvements and other changes to its Realtek Resources. Realtek has not conducted any testing other than that specifically described in the published documentation for a particular Realtek Resource.

Customer is authorized to use, copy and modify any individual Realtek Resource only in connection with the development of applications that include the Realtek product(s) identified in such Realtek Resource. No other license, express or implied, by estoppel or otherwise to any other Realtek intellectual property right, and no license to any technology or intellectual property right of Realtek or any third party is granted herein, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Realtek products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of Realtek Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from Realtek under the patents or other Realtek's intellectual property.

Realtek's Resources are provided “as is” and with all faults. Realtek disclaims all other warranties or representations, express or implied, regarding resources or use thereof, including but not limited to accuracy or completeness, title, any epidemic failure warranty and any implied warranties of merchantability, fitness for a particular purpose, and non-infringement of any third party intellectual property rights.

Realtek shall not be liable for and shall not defend or indemnify Customer against any claim, including but not limited to any infringement claim that related to or is based on any combination of products even if described in Realtek Resources or otherwise. In no event shall Realtek be liable for any actual, direct, special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of Realtek's Resources or use thereof, and regardless of whether Realtek has been advised of the possibility of such damages. Realtek is not responsible for any failure to meet such industry standard requirements.

Where Realtek specifically promotes products as facilitating functional safety or as compliant with industry functional safety standards, such products are intended to help enable customers to design and create their own applications that meet applicable functional safety standards and requirements. Using products in an application does not by itself establish any safety features in the application. Customers must ensure compliance with safety-related requirements and standards applicable to their applications. Designer may not use any Realtek products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death. Such equipment includes, without limitation, all medical devices identified by the U.S.FDA as Class III devices and equivalent classifications outside the U.S.

Customers agree that it has the necessary expertise to select the product with the appropriate qualification designation for their applications and that proper product selection is at Customers' own risk. Customers are solely responsible for compliance with all legal and regulatory requirements in connection with such selection.

Customer will fully indemnify Realtek and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's non-compliance with the terms and provisions of this Notice.

TRADEMARKS

Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

USING THIS DOCUMENT

Though every effort has been made to ensure that this document is current and accurate, more information may have become available subsequent to the production of this guide.

1 Configure AWS IoT Core

1.1 Set up your AWS account and Permissions

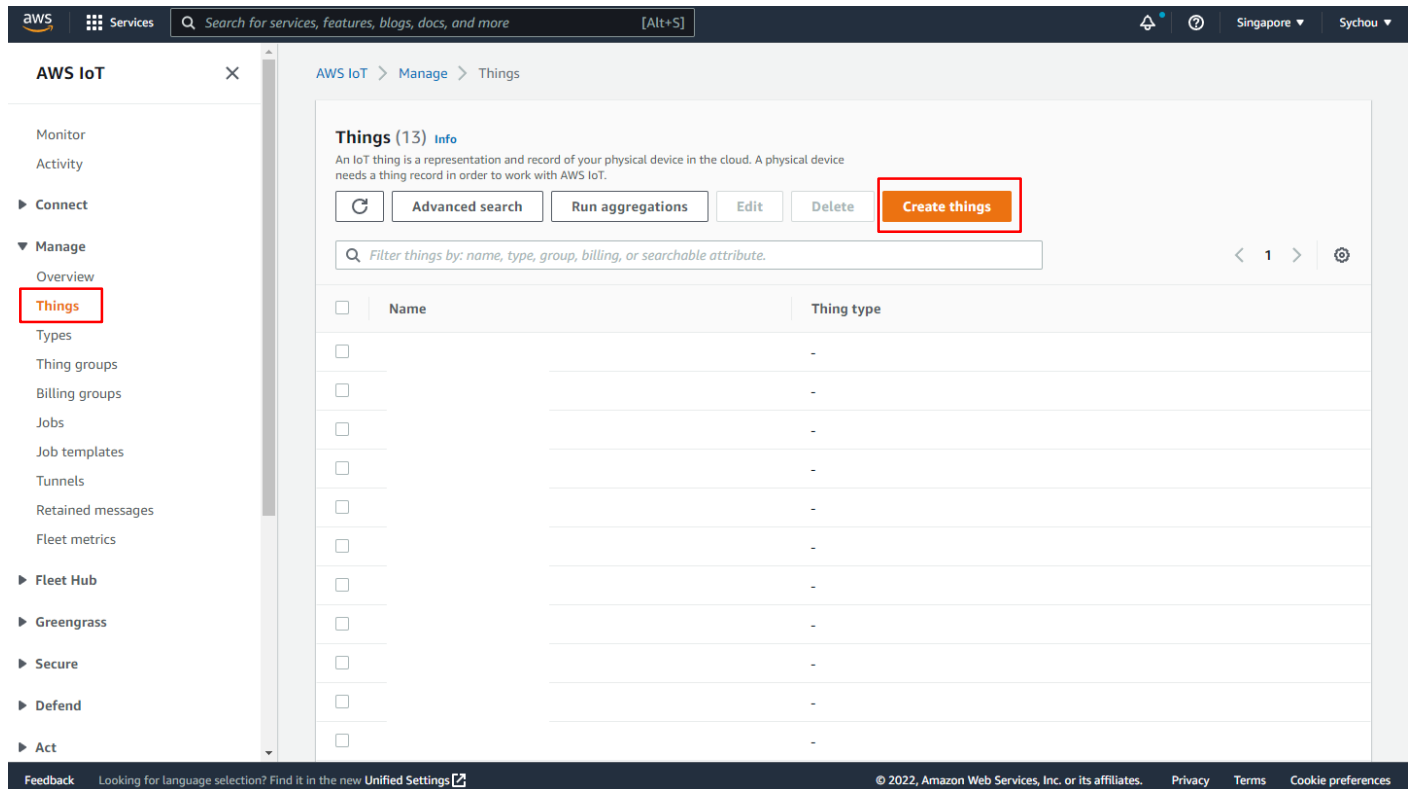
Refer to the instructions at Set up your AWS Account <https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html>. Follow the steps outlined in these sections to create your account and a user and get started:

- Sign up for an AWS account
- Create a user and grant permissions
- Open the AWS IoT console

Please pay special attention to the Notes in AWS webpage.

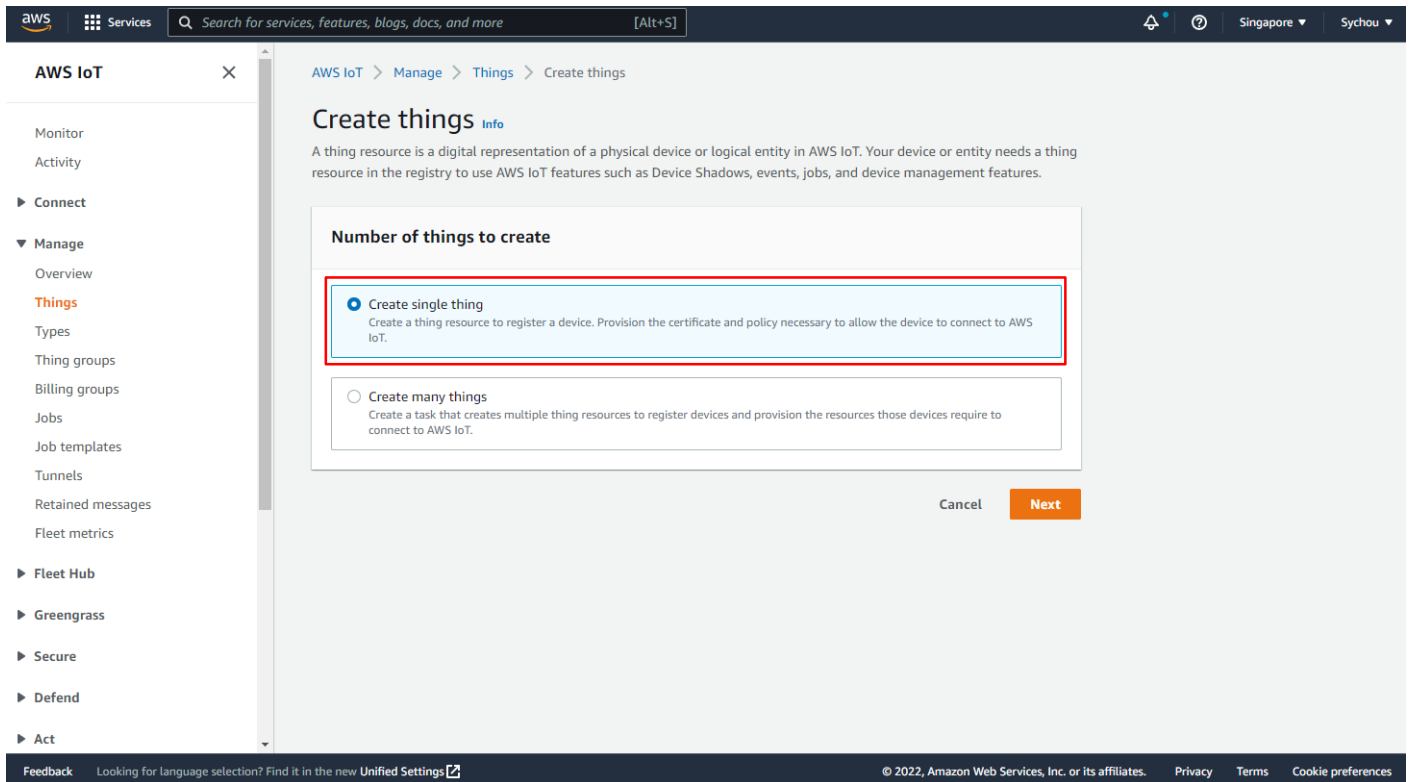
1.2 Create a New Device

To create a new device, navigate to Manage -> Things in the left-hand navigation menu. Then click "Create things".



The screenshot shows the AWS IoT console interface. On the left, the navigation menu is expanded to 'Manage', and 'Things' is highlighted with a red box. In the main content area, the 'Things (13)' page is displayed. The 'Create things' button is highlighted with a red box. Below the button is a search bar and a table with columns 'Name' and 'Thing type'. The table contains 13 rows, all with a '-' in the 'Thing type' column.

	Name	Thing type
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-
<input type="checkbox"/>		-

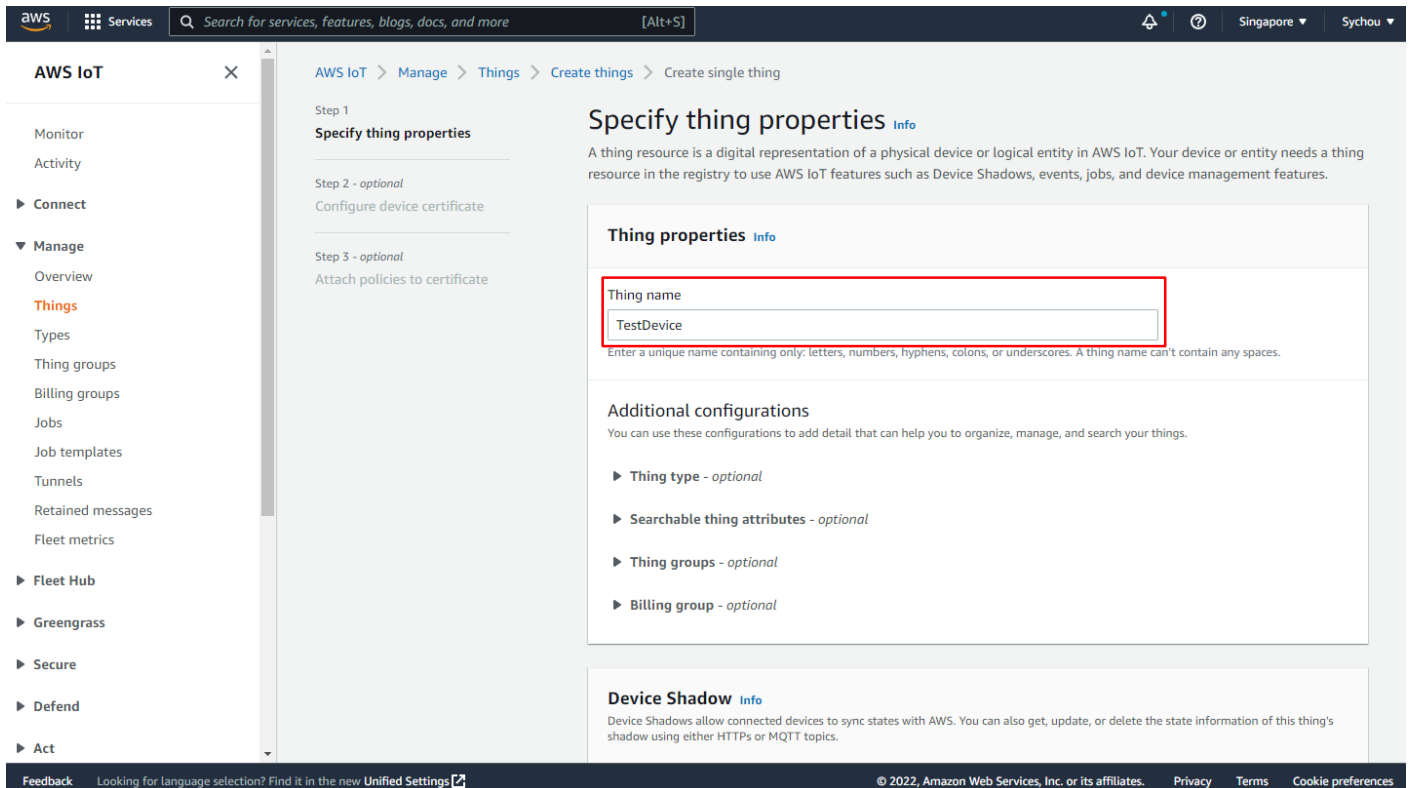


Number of things to create

- ☒ **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**

Then, name the new device. This example uses the name TestDevice.



Specify thing properties

Step 1
Step 2 - optional
Step 3 - optional

Thing properties

Thing name
TestDevice

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

Additional configurations

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

Device Shadow

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.

Configure device certificate - optional [Info](#)

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.

Device certificate

- ☒ **Auto-generate a new certificate (recommended)**
Generate a certificate, public key, and private key using AWS IoT's certificate authority.
- ☐ **Use my certificate**
Use a certificate signed by your own certificate authority.
- ☐ **Upload CSR**
Register your CA and use your own certificates on one or many devices.
- ☐ **Skip creating a certificate at this time**
You can create a certificate for this thing and attach a policy to the certificate at a later time.

Cancel Previous **Next**

Skip this part and “Create thing”, we will attach the policies to certificate later.

Attach policies to certificate - optional [Info](#)

AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.

Policies (9) [Refresh](#) [Create policy](#)

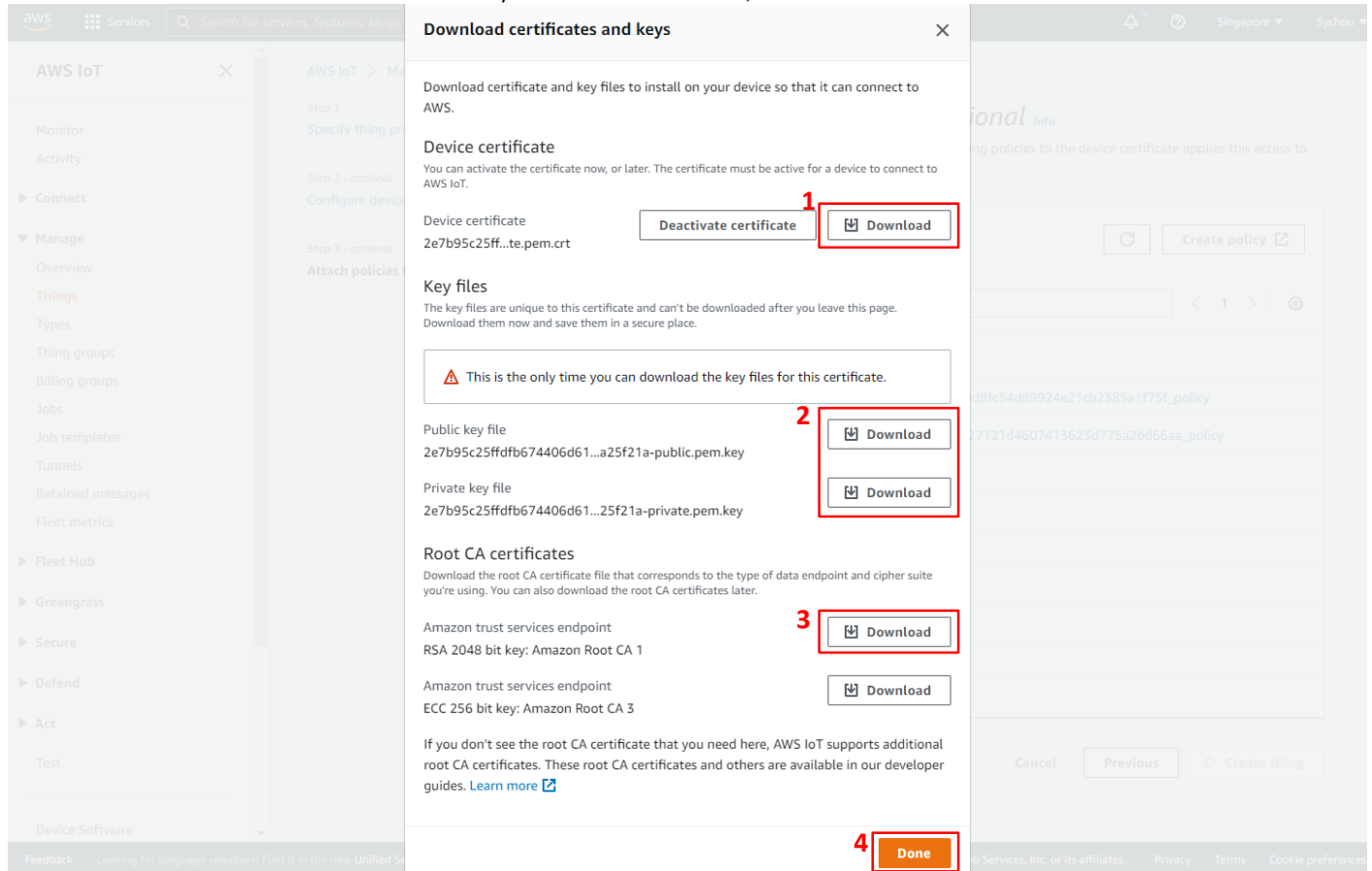
Select up to 10 policies to attach to this certificate.

< 1 > [Settings](#)

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

Cancel Previous **Create thing**

Download the certificate, public key, and private key for the device by clicking Download. Next, download the root CA for AWS IoT by clicking to the Download link. Once all the certificate and keys have been downloaded, click Done



Download certificates and keys

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate

You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate
2e7b95c25ff...te.pem.crt

Deactivate certificate **1** **Download**

Key files

The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

Public key file

2e7b95c25ffdfb674406d61...a25f21a-public.pem.key **2** **Download**

Private key file

2e7b95c25ffdfb674406d61...25f21a-private.pem.key **Download**

Root CA certificates

Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint **3** **Download**

RSA 2048 bit key: Amazon Root CA 1

Amazon trust services endpoint **Download**

ECC 256 bit key: Amazon Root CA 3

If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. [Learn more](#)

4 **Done**

1.3 Create a policy

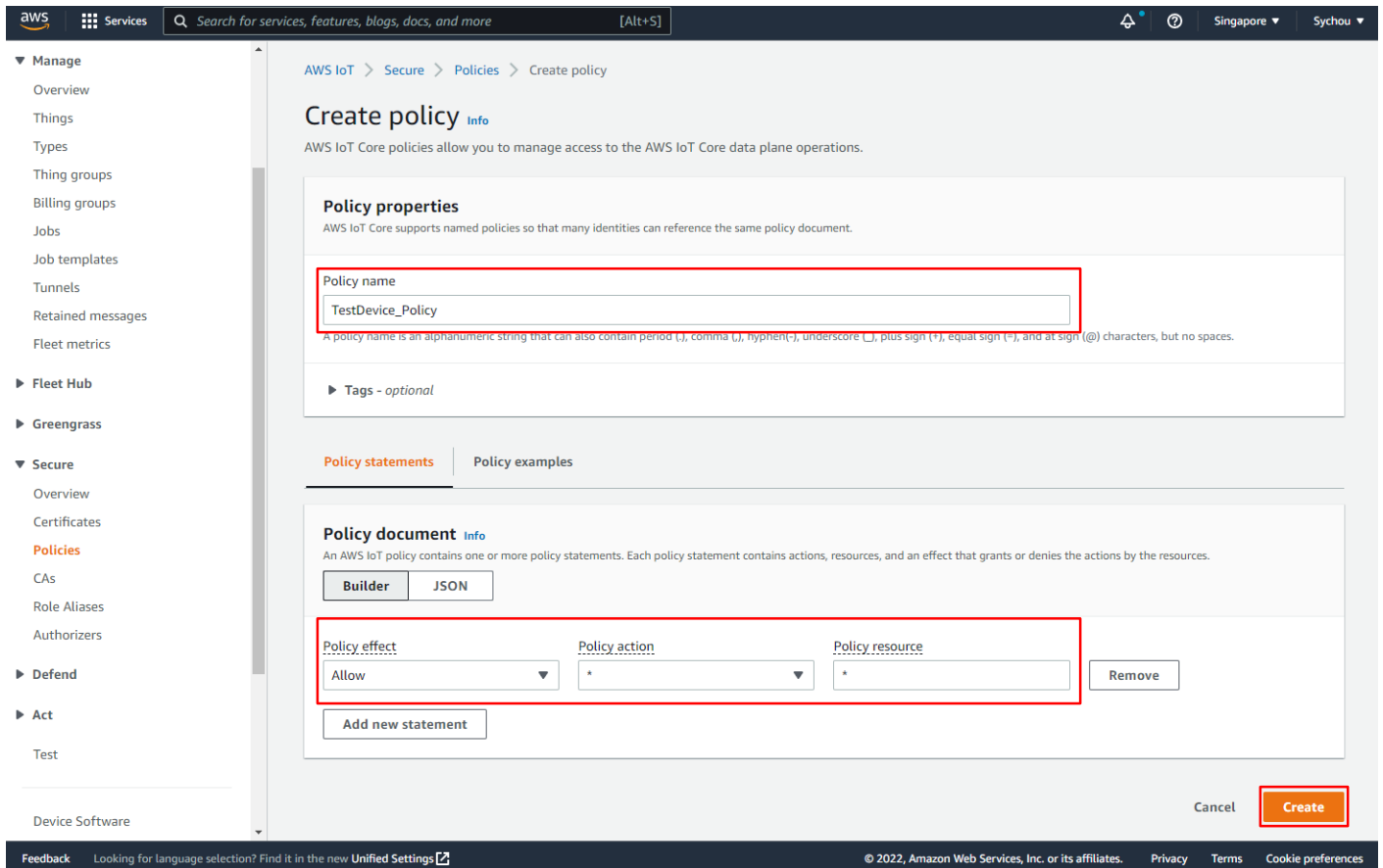
A policy defines a device's access permissions to IoT Core. To create a policy, navigate to Secure -> Policies. Then click "Create policy"

The screenshot shows the AWS IoT console interface. On the left is a navigation menu with categories: Manage, Fleet Hub, Greengrass, Secure, Defend, Act, and Device Software. The 'Secure' category is expanded, and 'Policies' is highlighted with a red box. The main content area is titled 'AWS IoT policies (9)' and includes a description: 'AWS IoT policies allow you to control access to the AWS IoT Core data plane operations. AWS IoT policies are separate and different from IAM policies. AWS IoT policies apply only to AWS IoT data plane operations.' Below the description are buttons for 'Refresh', 'Delete', and 'Create policy', with the 'Create policy' button highlighted by a red box. A search bar labeled 'Find policies' is also present. Below the search bar is a table with a header 'Policy name' and a dropdown arrow. The table currently contains no data rows. The footer of the console shows 'Feedback', a link for language selection, and copyright information for Amazon Web Services, Inc. (© 2022).

NOTE – this policy grants unrestricted access for all iot operations, and is to be used only in a development environment. For non-dev environments, all devices in your fleet must have credentials with privileges that authorize intended actions only, which include (but not limited to) AWS IoT MQTT actions such as publishing messages or subscribing to topics with specific scope and context. The specific permission policies can vary for your use cases. Identify the permission policies that best meet your business and security requirements.

For sample policies, refer to <https://docs.aws.amazon.com/iot/latest/developerguide/example-iot-policies.html>.

Also refer to <https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html>



Create policy [Info](#)

AWS IoT Core policies allow you to manage access to the AWS IoT Core data plane operations.

Policy properties

AWS IoT Core supports named policies so that many identities can reference the same policy document.

Policy name

TestDevice_Policy

A policy name is an alphanumeric string that can also contain period (.), comma (,), hyphen(-), underscore (_), plus sign (+), equal sign (=), and at sign (@) characters, but no spaces.

Tags - optional

Policy statements | **Policy examples**

Policy document [Info](#)

An AWS IoT policy contains one or more policy statements. Each policy statement contains actions, resources, and an effect that grants or denies the actions by the resources.

Builder | **JSON**

Policy effect	Policy action	Policy resource	
Allow	*	*	Remove

Add new statement

Create

1.4 Attach Policy

The last step to configuring the device is attaching a policy. To attach a policy to new device, navigate to Manage -> Things. Then click on the device which was created.

Things (14) Info

An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

Advanced search Run aggregations Edit Delete Create things

Filter things by: name, type, group, billing, or searchable attribute.

Name	Thing type
TestDevice	-
	-
	-
	-
	-
	-
	-
	-
	-
	-
	-
	-
	-
	-

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click Certificate, then choose the certificate create in previous step.

TestDevice Info Edit Delete

Thing details

Name	Type
TestDevice	-
ARN	Billing group
arn:aws:iot:ap-southeast-1:553661462376:thing/TestDevice	-

Attributes Certificates Thing groups Device Shadows Interact Activity Jobs Alarms Defender metrics

Certificates (1) Info Create certificate

The device certificates attached to this thing resource.

Find certificates

Certificate ID	Status
2e7b95c25ffdb674406d61f5d5cad546be1dd06689e2b80dd89dfa4ca25f21a	Active

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS IoT console interface. The left sidebar contains navigation links for Manage, Fleet Hub, Greengrass, Secure, and Defend. The main content area displays the details of a certificate with ID `2e7b95c25ffdfb674406d61f5d5cad546be1dd06689e2b80dd89dfa4ca25f21a`. The 'Actions' menu is open, showing options like Activate, Deactivate, Revoke, and Attach policy. The 'Attach policy' option is highlighted. The certificate details on the right show it is 'Active', created on May 24, 2022, and expires on January 01, 2050.

The screenshot shows the 'Attach policies' dialog box in the AWS IoT console. The dialog box has a search bar at the top and a list of policies below. The policy `TestDevice_Policy` is selected and highlighted with a red box. The dialog box also shows the certificate ID `2e7b95c25ffdfb674406d61f5d5cad546be1dd06689e2b80dd89dfa4ca25f21a` and the 'Attach policies' button is highlighted with a red box.

2 Configure AmebaPro2 Amazon FreeRTOS

2.1 Download FreeRTOS-LTS Library Source Code from Github

Open source link: <https://github.com/ambiot/amazon-freertos/tree/amebaPro2-9.x-202107.00-LTS>
branch: **amebaPro2-9.x-202107.00-LTS**

2.1.1 Download Source Code of Required Libraries to SDK

Go to “AmebaPro2_SDK/project/realtek_amebapro2_v0_example/src”:

```
$ cd project/realtek_amebapro2_v0_example/src
$ git clone --recurse-submodules -b amebaPro2-9.x-202107.00-LTS https://github.com/ambiot/amazon-freertos.git aws_iot_freertos_Its
```

2.1.2 Modify FreeRTOSConfig.h

Copy & paste below configurations to the end of FreeRTOSConfig.h in “project\realtek_amebapro2_v0_example\inc”:

```
/* Sets the length of the buffers into which logging messages are written - so
 * also defines the maximum length of each log message. */
#define configLOGGING_MAX_MESSAGE_LENGTH 512

/* Set to 1 to prepend each log message with a message number, the task name,
 * and a time stamp. */
#define configLOGGING_INCLUDE_TIME_AND_TASK_NAME 1

/* Map the FreeRTOS printf() to the logging task printf. */
/* The function that implements FreeRTOS printf style output, and the macro
 * that maps the configPRINTF() macros to that function. */
#define configPRINTF( X ) vLoggingPrintf X

/* Non-format version thread-safe print. */
#define configPRINT( X ) vLoggingPrint( X )

/* Map the logging task's printf to the board specific output function. */
#define configPRINT_STRING( X ) printf( X )

#define iotconfigUSE_PORT_SPECIFIC_HOOKS
```

2.1.3 Configure MbedTLS Setting

In this project, we use mbedtls-2.16.6, same as KVS webrtc. Set mbedtls version to 2.16.6 in “project/realtek_amebapro2_v0_example/GCC-RELEASE/application/CMakeLists.txt”

```
set(mbedtls "mbedtls-2.16.6")
```

You have to modify some mbedtls config before running aws-iot demo, go to “component\ssl\mbedtls-2.16.6\include\mbedtls\config_rsa.h” check the following setting:

```
#define MBEDTLS_THREADING_ALT
// #define MBEDTLS_DEBUG_C
#define MBEDTLS_THREADING_C
```

The default mbedtls version of AmebaPro2 is 3.0.0. However, for the aws iot demo, we use mbedtls version 2.16.6 in default. It might be easier for user to use it with AWS KVS service now.

If user want to use the aws-iot with mbedtls-3.0.0 or mbedtls-2.4.0, user can compare the config file between mbedtls-2.16.6 and mbedtls-3.0.0, mbedtls-2.4.0

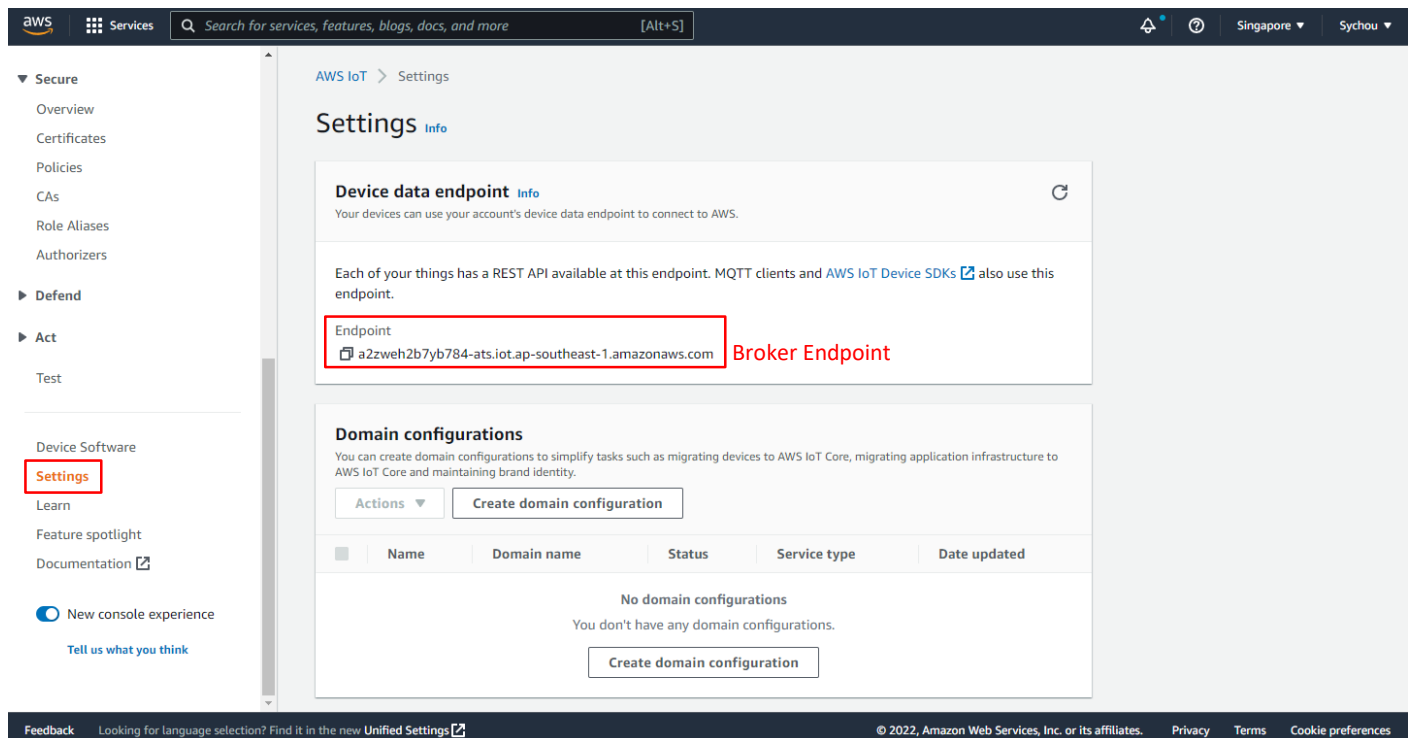
2.1.4 Multiple Definition Issue

There might be multiple definition of “vApplicationGetIdleTaskMemory” and “vApplicationGetTimerTaskMemory”. Since aws demo runner have the same function that have been defined in SDK, so we should comment one of them, go to “component\os\freertos\freertos_cb.c” and comment these two functions

```
//void vApplicationGetIdleTaskMemory(...)
//{
//    ...
//}

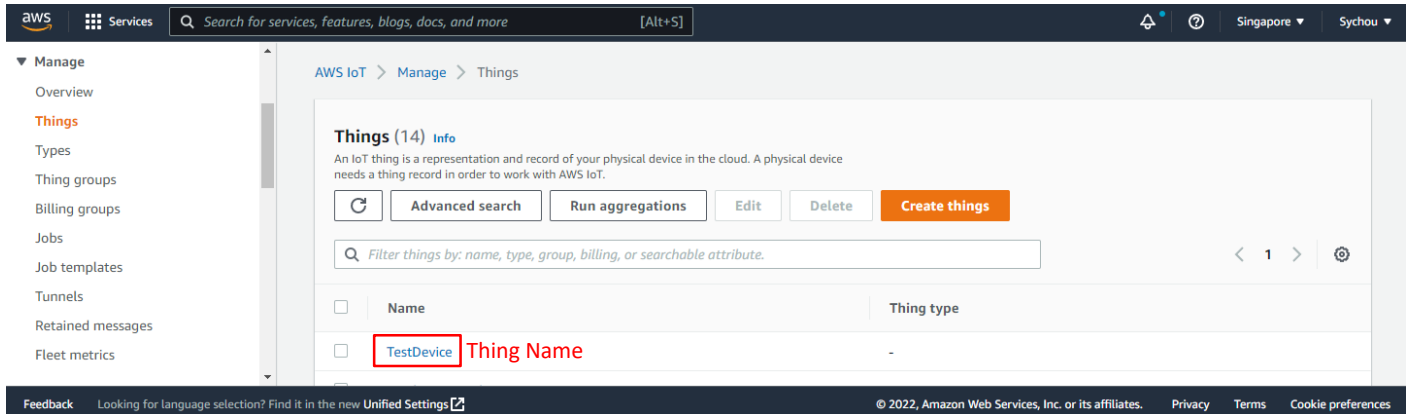
//void vApplicationGetTimerTaskMemory(...)
//{
//    ...
//}
```

2.2 Get Broker Endpoint by AWS IoT Core



The screenshot shows the AWS IoT Settings page. On the left sidebar, the 'Settings' option under 'Device Software' is highlighted with a red box. The main content area shows the 'Settings' page for AWS IoT. The 'Device data endpoint' section is visible, showing the endpoint 'a2zweh2b7yb784-ats.iot.ap-southeast-1.amazonaws.com' which is highlighted with a red box and labeled 'Broker Endpoint'. Below this, the 'Domain configurations' section shows a table with columns: Name, Domain name, Status, Service type, and Date updated. The table is currently empty, and a 'Create domain configuration' button is visible.

2.3 Get Thing Name



The screenshot shows the AWS IoT console interface. On the left, there is a navigation menu with options like 'Manage', 'Overview', 'Things', 'Types', 'Thing groups', 'Billing groups', 'Jobs', 'Job templates', 'Tunnels', 'Retained messages', and 'Fleet metrics'. The main content area is titled 'Things (14) Info' and includes a description: 'An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.' Below this, there are buttons for 'Advanced search', 'Run aggregations', 'Edit', 'Delete', and 'Create things'. A search bar is present with the placeholder text 'Filter things by: name, type, group, billing, or searchable attribute.' Below the search bar, there is a table with columns 'Name' and 'Thing type'. The first row in the table shows 'TestDevice' under the 'Name' column, which is highlighted with a red box. The 'Thing type' column for this row is empty.

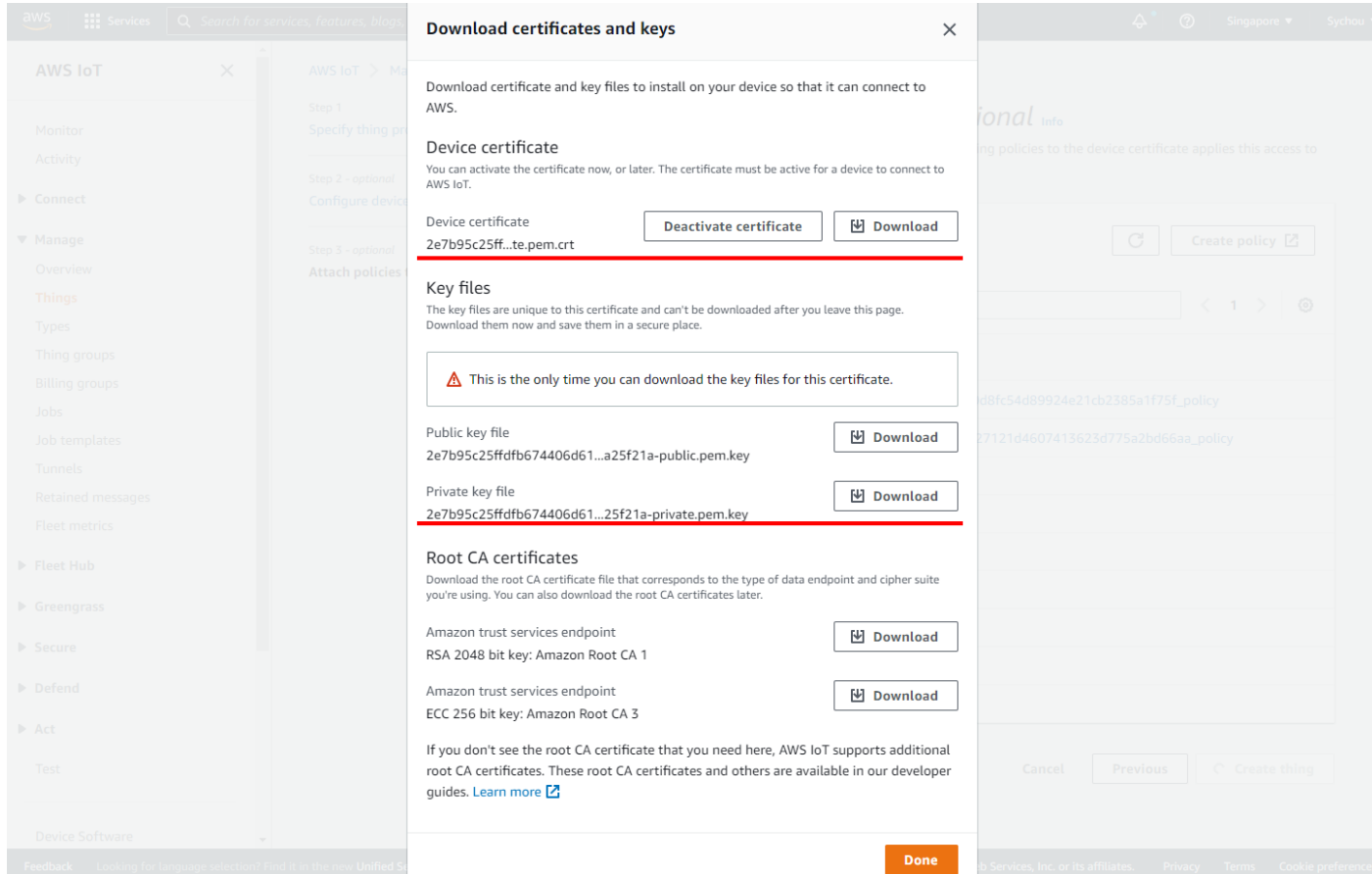
2.4 Setup IoT Core Information with AmebaPro2 Amazon FreeRTOS

Setup BROKER_ENDPOINT, THING_NAME, WIFI_SSID, PASSWORD in
 "project/realtek_amebapro2_v0_example/src/aws_iot_freertos_Its/demos/include/aws_clientcredential.h"

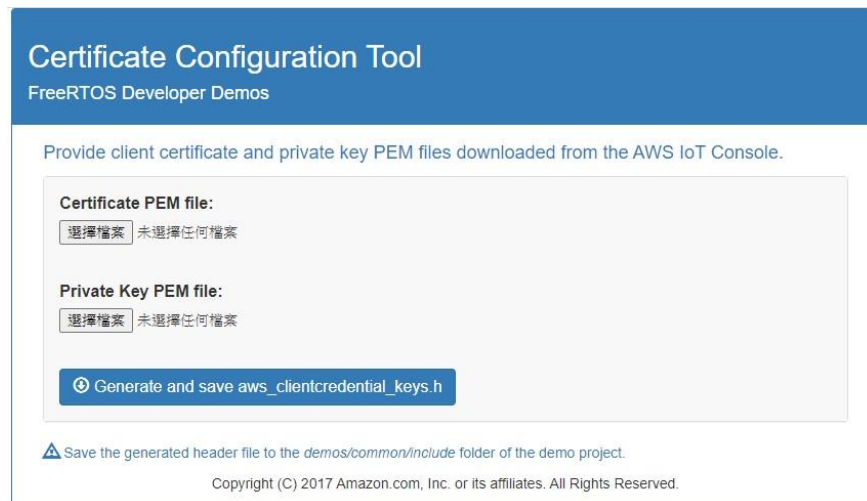
```
#define clientcredentialMQTT_BROKER_ENDPOINT      "xxxxxxxxxxxxx.amazonaws.com"
/*
 * @brief Host name.
 *
 * @todo Set this to the unique name of your IoT Thing.
 */
#define clientcredentialIOT_THING_NAME          "TestDevice"
/*
 * @brief Port number the MQTT broker is using.
 */
#define clientcredentialMQTT_BROKER_PORT        8883
/*
 * @brief Port number the Green Grass Discovery use for JSON retrieval from cloud is using.
 */
#define clientcredentialGREENGRASS_DISCOVERY_PORT  8443
/*
 * @brief Wi-Fi network to join.
 *
 * @todo If you are using Wi-Fi, set this to your network name.
 */
#define clientcredentialWIFI_SSID                "TestAP"
/*
 * @brief Password needed to join Wi-Fi network.
 * @todo If you are using WPA, set this to your network password.
 */
#define clientcredentialWIFI_PASSWORD           "password"
/*
 * @brief Wi-Fi network security type.
 *
 * @see WiFiSecurity_t.
 *
 * @note Possible values are eWiFiSecurityOpen, eWiFiSecurityWEP, eWiFiSecurityWPA,
 * eWiFiSecurityWPA2 (depending on the support of your device Wi-Fi radio).
 */
#define clientcredentialWIFI_SECURITY            eWiFiSecurityWPA2
#endif /* ifndef __AWS_CLIENTCREDENTIAL_H__ */
```

2.4.1 Setup Thing's Private Key and Certificate

Fill `keyCLIENT_CERTIFICATE_PEM` and `keyCLIENT_PRIVATE_KEY_PEM` in “`project/realtek_amebapro2_v0_example/src/aws_iot_freertos_lts/demos/include/aws_clientcredential_keys.h`” by `xxxxxxx-certifiacte.pem` and `xxxxxxx-private.pem.key`.



It can be done by [CertificateConfigurator.html](https://raw.githubusercontent.com/aws/amazon-freertos/202107.00/tools/certificate_configuration/CertificateConfigurator.html) and it can be downloaded from https://raw.githubusercontent.com/aws/amazon-freertos/202107.00/tools/certificate_configuration/CertificateConfigurator.html. Please download it before using.



Final aws_clientcredential_keys.h overview.

```

/*
 * @brief PEM-encoded client certificate.
 *
 * @todo If you are running one of the FreeRTOS demo projects, set this
 * to the certificate that will be used for TLS client authentication.
 *
 * @note Must include the PEM header and footer:
 * "-----BEGIN CERTIFICATE-----\n"
 * "...base64 data...\n"
 * "-----END CERTIFICATE-----\n"
 */
#define keyCLIENT_CERTIFICATE_PEM \
"-----BEGIN CERTIFICATE-----\n"
"MIIDWjCCAkKgAwIBAgIVAIDLSSoG+EARsBbPrT4Imluu8j2vMA0GCSqGSIb3DQEB\n"
"-----\n"
"-----END CERTIFICATE-----\n"

/*
 * @brief PEM-encoded client private key.
 *
 * @todo If you are running one of the FreeRTOS demo projects, set this
 * to the private key that will be used for TLS client authentication.
 *
 * @note Must include the PEM header and footer:
 * "-----BEGIN RSA PRIVATE KEY-----\n"
 * "...base64 data...\n"
 * "-----END RSA PRIVATE KEY-----\n"
 */
#define keyCLIENT_PRIVATE_KEY_PEM \
"-----BEGIN RSA PRIVATE KEY-----\n"
"MIIEpAIBAAKCAQEAwcp96WNucGebARFjD8O+CLsqcBNn/AHyhEcozLZC8qcECUOn\n"
"-----\n"
"-----END RSA PRIVATE KEY-----\n"

```

2.4.2 Enable FreeRTOS demo on AmebaPro2

For example, if you would like to run MQTT mutual authentication demo, please find aws_demo_config.h in “project/realtek_amebapro2_v0_example/src/aws_iot_freertos_lts/vendors/realtek/boards/amebaPro2/aws_demos/config_files” and enable **CONFIG_CORE_MQTT_MUTUAL_AUTH_DEMO_ENABLED**

```

// #define CONFIG_CORE_HTTP_MUTUAL_AUTH_DEMO_ENABLED
#define CONFIG_CORE_MQTT_MUTUAL_AUTH_DEMO_ENABLED
// #define CONFIG_DEVICE_SHADOW_DEMO_ENABLED
// #define CONFIG_JOBS_DEMO_ENABLED

```

Now you can start to compile AmebaPro2 Amazon FreeRTOS project !

3 Compile AmebaPro2 Amazon FreeRTOS

3.1 Compile Program with GCC Toolchain

Run following commands to build the image with option ``-DEXAMPLE=amazon_freertos``

```
$ cd project/realtek_amebapro2_v0_example/GCC-RELEASE
$ mkdir build
$ cd build
$ cmake .. -G"Unix Makefiles" -DCMAKE_TOOLCHAIN_FILE=../toolchain.cmake -DEXAMPLE=amazon_freertos
$ cmake --build . --target flash -j4
```

After successfully build, there should be an image file **flash_ntz.bin** located in "build/" directory.

3.2 Download image to AmebaPro2

Use image tool to download the image to AmebaPro2.

4 MQTT Demo

4.1 Run MQTT Demo

Default setting of SDK are enable MQTT demo. Once the AmebaPro2 EVB has rebooted, the application will automatically start run MQTT demo and communicate to IoT Core.

```
[Driver]: set ssid [RealEZ]
[RF] [RFK] Tx pause!!
[Driver]: start auth to 
[Driver]: auth alg = 2
[Driver]: auth success, start assoc
[Driver]: association success(res=28)
[Driver]: wlan0: DL RSVD page success! DLBcnCount:1, poll:1
0 301 [example_ama] Write certificate...
1 408 [iot_thread] [INFO ][DEMO][408] -----STARTING DEMO-----
2 414 [iot_thread] [INFO ][INIT][414] SDK successfully initialized.
```

...

```
Interface 0 IP address : 192.168. [redacted]
3 53555 [iot_thread] [INFO] [DEMO][53555] Successfully initialized the demo. Network type for the demo: 1

4 53564 [iot_thread] [INFO] Creating a TLS connection to [redacted]-ats.iot.ap-southeast-1.amazonaws.com:8883.
5 54778 [iot_thread] [INFO] Creating an MQTT connection to [redacted]-ats.iot.ap-southeast-1.amazonaws.com.
6 54909 [iot_thread] [INFO] Packet received. ReceivedBytes=2.
7 54913 [iot_thread] [INFO] CONNACK session present bit not set.
8 54919 [iot_thread] [INFO] Connection accepted.
9 54924 [iot_thread] [INFO] Received MQTT CONNACK successfully from broker.
10 54930 [iot_thread] [INFO] MQTT connection established with the broker.
11 54937 [iot_thread] [INFO] An MQTT connection is established with [redacted]-ats.iot.ap-southeast-1.amazonaws.com.
12 54949 [iot_thread] [INFO] Attempt to subscribe to the MQTT topic ameba-ota/example/topic.
13 54956 [iot_thread] [INFO] SUBSCRIBE sent for topic ameba-ota/example/topic to broker.
14 55070 [iot_thread] [INFO] Packet received. ReceivedBytes=3.
15 55074 [iot_thread] [INFO] Subscribed to the topic ameba-ota/example/topic with maximum QoS 1.
16 56082 [iot_thread] [INFO] Publish to the MQTT topic ameba-ota/example/topic.
17 56087 [iot_thread] [INFO] Attempt to receive publish message from broker.
18 56241 [iot_thread] [INFO] Packet received. ReceivedBytes=2.
19 56246 [iot_thread] [INFO] Ack packet deserialized with result: MQTTSuccess.
20 56252 [iot_thread] [INFO] State record updated. New state=MQTTPublishDone.
21 56259 [iot_thread] [INFO] PUBACK received for packet Id 2.
22 56265 [iot_thread] [INFO] Packet received. ReceivedBytes=39.
23 56270 [iot_thread] [INFO] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
24 56280 [iot_thread] [INFO] State record updated. New state=MQTTPubAckSend.
25 56286 [iot_thread] [INFO] Incoming QoS : 1

...

248 122674 [iot_thread] [INFO] Demo run is successful with 3 successful loops out of total 3 loops.
249 123681 [iot_thread] [INFO] [DEMO][123681] Demo completed successfully.

Deinitializing WIFI ...
WIFI deinitialized250 123809 [iot_thread] [INFO] [INIT][123809] SDK cleanup done.

251 123813 [iot_thread] [INFO] [DEMO][123813] -----DEMO FINISHED-----
```

4.2 Monitoring MQTT Messages on the Cloud

To subscribe to the MQTT topic with the AWS IoT MQTT client

1. Sign in to the AWS IoT console.
2. In the navigation pane, choose Test to open the MQTT client.
3. In Subscription topic, enter “+/example/topic”, and then choose Subscribe to topic.

AWS IoT

Monitor

Activity

▶ Onboard

▶ Manage

▶ Greengrass

▶ Secure

▶ Defend

▶ Act

Test

Software

Settings

Learn

Documentation

New console experience

Tell us what you think

AWS IoT > MQTT test client

MQTT test client

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Subscribe to a topic

Publish to a topic

Topic filter

+ /example/topic

Additional configuration

Subscribe

Subscriptions

Topic

You have no topic subscriptions.

Subscribe to a topic to view incoming messages.

AWS IoT

Monitor

Activity

▶ Onboard

▶ Manage

▶ Greengrass

▶ Secure

▶ Defend

▶ Act

Test

Software

Settings

Learn

Documentation

New console experience

Tell us what you think

Subscriptions

+ /example/topic

Pause

Clear

Export

Edit

+ /example/topic

▼ ameba-ota/example/topic

March 08, 2021, 17:14:36 (UTC+0800)

Hello World!

▼ ameba-ota/example/topic

March 08, 2021, 17:14:23 (UTC+0800)

Hello World!

▼ ameba-ota/example/topic

March 08, 2021, 17:14:21 (UTC+0800)

Hello World!

▼ ameba-ota/example/topic

March 08, 2021, 17:14:20 (UTC+0800)

Hello World!

▼ ameba-ota/example/topic

March 08, 2021, 17:14:17 (UTC+0800)

Hello World!

5 Troubleshooting

If these steps don't work, look at the device log in the serial terminal. You should see some text that indicates the source of the problem.

For general troubleshooting information about Getting Started with FreeRTOS, see [Troubleshooting getting started](#).

5.1 ERROR: Invalid Key

Please check **WIFI_SSID** and **WIFI_PASSWORD** in "project/realtek_amebapro2_v0_example/src/aws_iot_freertos_its/demos/include/aws_clientcredential.h"

```
Enter SSID for Soft AP started
3 1098 [example_a] Wi-Fi configuration successful.
4 1108 [iot_threa] [INFO ][DEMO][1108] -----STARTING DEMO-----

5 1115 [iot_threa] [INFO ][INIT][1115] SDK successfully initialized.

LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
WIFI deinitialized
Initializing WIFI ...
WIFI initialized

Joining BSS by SSID ...

ERROR:Invalid Key
ERROR: Can't connect to AP
Joining BSS by SSID ...

ERROR:Invalid Key
ERROR: Can't connect to AP
Joining BSS by SSID ...
```

5.2 Failed to establish new MQTT connection

Please check **clientcredentialMQTT_BROKER_ENDPOINT** in

"project/realtek_amebapro2_v0_example/src/aws_iot_freertos_its/demos/include/aws_clientcredential.h"

```
6 12508 [iot_threa] [INFO ][DEMO][12508] Successfully initialized the demo. Network type for the demo: 1
7 12517 [iot_threa] [INFO ][MQTT][12517] MQTT library successfully initialized.
8 12524 [iot_threa] [INFO ][DEMO][12524] MQTT demo client identifier is ameba-ota (length 9).
9 12624 [iot_threa] [ERROR][NET][12624] Failed to resolve [redacted].amazonaws.com.
10 12934 [iot_threa] [ERROR][MQTT][12934] Failed to establish new MQTT connection, error NETWORK ERROR.
11 12943 [iot_threa] [ERROR][DEMO][12943] MQTT CONNECT returned error NETWORK ERROR.
12 12951 [iot_threa] [INFO ][MQTT][12950] MQTT library cleanup done.
13 12957 [iot_threa] [ERROR][DEMO][12957] Error running demo.
Interface 0 IP address : 192.168.90.185
LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
14 13094 [iot_threa] [INFO ][INIT][13094] SDK cleanup done.
15 13099 [iot_threa] [INFO ][DEMO][13099] -----DEMO FINISHED-----
```

5.3 TLS_Connect fail

Please check **keyCLIENT_CERTIFICATE_PEM** and **keyCLIENT_PRIVATE_KEY_PEM** in

"project/realtek_amebapro2_v0_example/src/aws_iot_freertos_its/demos/include/aws_clientcredential_keys.h"

```
8 13501 [iot_threa] [INFO ][DEMO][13501] Successfully initialized the demo. Network type for the demo: 1
9 13511 [iot_threa] [INFO ][MQTT][13511] MQTT library successfully initialized.
10 13518 [iot_threa] [INFO ][DEMO][13518] MQTT demo client identifier is ameba-ota (length 9).
11 20102 [iot_threa] [ERROR][MQTT][20102] TLS Connect fail (0x7d4, [redacted].amazonaws.com)
13 20115 [iot_threa] [ERROR][NET][20115] Failed to establish new connection. Socket status: -1.
14 20424 [iot_threa] [ERROR][MQTT][20424] Failed to establish new MQTT connection, error NETWORK ERROR.
15 20433 [iot_threa] [ERROR][DEMO][20433] MQTT CONNECT returned error NETWORK ERROR.
16 20441 [iot_threa] [INFO ][MQTT][20441] MQTT library cleanup done.
17 20447 [iot_threa] [ERROR][DEMO][20447] Error running demo.
Interface 0 IP address : 192.168.90.185
LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
18 20586 [iot_threa] [INFO ][INIT][20586] SDK cleanup done.
19 20591 [iot_threa] [INFO ][DEMO][20591] -----DEMO FINISHED-----
```