



BCM4343W IoT Starter Kit Getting Started Guide

Version 2.1

Prior Version History

Version	Date	Comment
1.0	10/23/2015	Initial Release
1.1	20/30/2015	Procedure updated to include remote control of LED plus email and SMS text messages to User's cellphone
2.0	11/15/2016	Updated to show latest AWS user interfaces and color scheme. All references to WICED SDK and BCM4343W device changed from Broadcom to Cypress
2.1	11/29/2016	Updated board photos

Contents

1	Introduction.....	4
2	What's Inside the Box?	4
2.1	BCM4343W IoT Starter Kit Kit contents	4
3	What's on the Web?	5
3.1	Tutorials and Reference Designs:.....	5
3.2	Trainings and Videos:.....	5
3.3	BCM4343W IoT Starter Kit Key Features.....	6
3.4	Overview: AWS IoT Service	8
3.5	Overview: Out-of-box Shadow Example.....	8
3.6	Overview: More Advanced Exercises.....	9
4	BCM4343W IoT Starter Kit Setup and Operation	9
4.1	AWS IoT Console Procedure.....	9
4.2	Simple Exercise: Run the Out-of-Box "Thing Shadow" Application	18
4.3	Advanced Exercise: Sending Email and SMS Messages	21
4.3.1	AWS SNS: Setup EMAIL Messaging Topic	21
4.3.2	AWS SNS: Setup EMAIL Subscription to EMAIL Topic	23
4.3.3	AWS SNS: Setup SMS Messaging Topic	24
4.3.4	AWS SNS: Setup SMS Subscription to SMS Topic	25
4.3.5	AWS IoT: Setup Rule1: Send EMAIL on User Button Press	27
4.3.6	AWS IoT: Setup Rule2: Send SMS on Remote Change of User1 LED	29
4.3.7	AWS IoT: Testing Email Messaging (Rule1)	30
4.3.8	AWS IoT: Testing SMS Messaging (Rule2)	31
5	Getting Help and Support	33
5.1	Avnet Support	33
5.2	Cypress WICED Wi-Fi Forum Support.....	33
5.3	AWS Forum Support	33

Figures

Figure 1 – BCM4343W SoC Module.....	4
Figure 2 – BCM4343W IOT STARTER KIT – Feature Identification	6
Figure 3 – BCM4343W IOT STARTER KIT – High-Level Block Diagram	7
Figure 4 – AWS IoT	7
Figure 5 – Simplified view of AWS IoT	8

1 Introduction

This document details the procedure to setup and use the **BCM4343W IoT Starter Kit** for easy prototyping of cloud-connected IoT designs.

The Starter Kit is unique in that it also facilitates rapid transition from development to production using the pre-certified wireless SoC module. It features an Arduino™ form-factor carrier fitted with a pre-certified **Avnet BCM4343W SoC module** that combines advanced Cypress® 802.11 b/g/n WiFi and Bluetooth® 4.1 combo SoC, together with 8Mb SPI Serial Flash and STM32F411 ARM® Cortex™ M4 Microcontroller (with 512KB Flash, 128KB RAM)



Figure 1 – BCM4343W SoC Module

MCU interface peripherals can be connected to a wide range of expansion boards via:

1. Arduino-compatible header connectors, as well as
2. The Pmod™-compatible 2x6 peripheral connector

Various options are provided for cloud-connected application development: **Amazon Web Services (AWS IoT)** examples are provided within Cypress WICED™ Studio **IBM Bluemix (Watson IoT)** examples based on ZentriOS SDK are provided on the CloudConnectKits website (see detail on next page)

2 What's Inside the Box?

2.1 BCM4343W IoT Starter Kit Kit contents

- Avnet BCM4343W IoT Starter Kit board (part#: **AES-EVB-BCM4343W-G**)
- MicroUSB cable
- Quick Start Card (with instructions on how to run the **AWS IoT Shadow** demo)

3 What's on the Web?

Documents are located at: www.cloudconnectkits.org

- Getting Started Guide
- Hardware User Guide
- Schematics
- Bill Of Materials
- PCB Layout (Gerber files)
- Mechanical Drawing
- Avnet BCM4343W IoT Starter Kit Brochure
- Avnet BCM4343W SoC Module Brochure



3.1 Tutorials and Reference Designs:

These are located under the “**Startup Files & Reference Designs**” tab at:

<http://cloudconnectkits.org/product/avnet-bcm4343w-iot-starter-kit>

- AV01: Connect to IBM Watson IoT using Avnet BCM4343W IoT Starter Kit
- AV02: Sending Alerts from BCM4343W IoT Starter Kit using IBM Bluemix
- AV03: WICED Sense BLE data publish to IBM Bluemix via BCM94343W_AVN
- AV04: Alexa Voice-Controlled Smart Home demo using BCM4343W IoT Starter Kit and IBM Bluemix
- AV05 - Sending Alerts from BCM4343W IoT Starter Kit Alexa Voice-Controlled Smart Home Demo (v1.1)
- AV06: WICED Sense2 BLE Tag to IBM Bluemix demo
- AV07 - Creating Dashboards Using Bluemix Services (v1.1)
- AV08: Sensor-to-Cloud Using TI SensorTag and Watson IoT Quickstart
- AV09: Zentri to IBM Bluemix Notifications and Visualizations
- AV10: NXP 3D Shield LED Matrix demo
- AV11 - Webpage Controlled LED Matrix Display (ZentriOS, Wi-Fi v1.0)

3.2 Trainings and Videos:

Accessible under: <http://cloudconnectkits.org/product/avnet-bcm4343w-iot-starter-kit>

3.3 BCM4343W IoT Starter Kit Key Features

- Arduino™ form-factor baseboard
- Pre-certified Avnet BCM4343W SoC Module
 - WiFi + BLE + MCU module
 - STM32F411 ARM® Cortex™ M4 MCU (512KB Flash, 128KB SRAM)
 - 8Mb SPI Serial Flash
 - Supports 802.11 b/g/n and Bluetooth 4.1 (with upgrade path to Bluetooth 4.2)
 - Dual fractal PCB antennas supporting antenna diversity
 - Onboard connectors for dual external antennas
- Arduino compatible shield expansion connectors
 - GPIO (4), Analog inputs (3)
 - 2x I2C ports (1 shared), 1x SPI port, 2x UARTs (1 shared)
- Peripheral expansion connector (Pmod-compatible, 2x6 format)
 - I2C port (shared)
- USB-based JTAG debugger/ programmer and serial UART port
- 1x Reset push button switch, 1x User push button switch
- 2x User LEDs
- 4x Status LEDs (UART, JTAG and WLAN activity LEDs, plus 3V3 Power LED)
- 1x Ambient light sensor
- USB Powered (5V), onboard high capacity 5V to 3.3V regulated supply

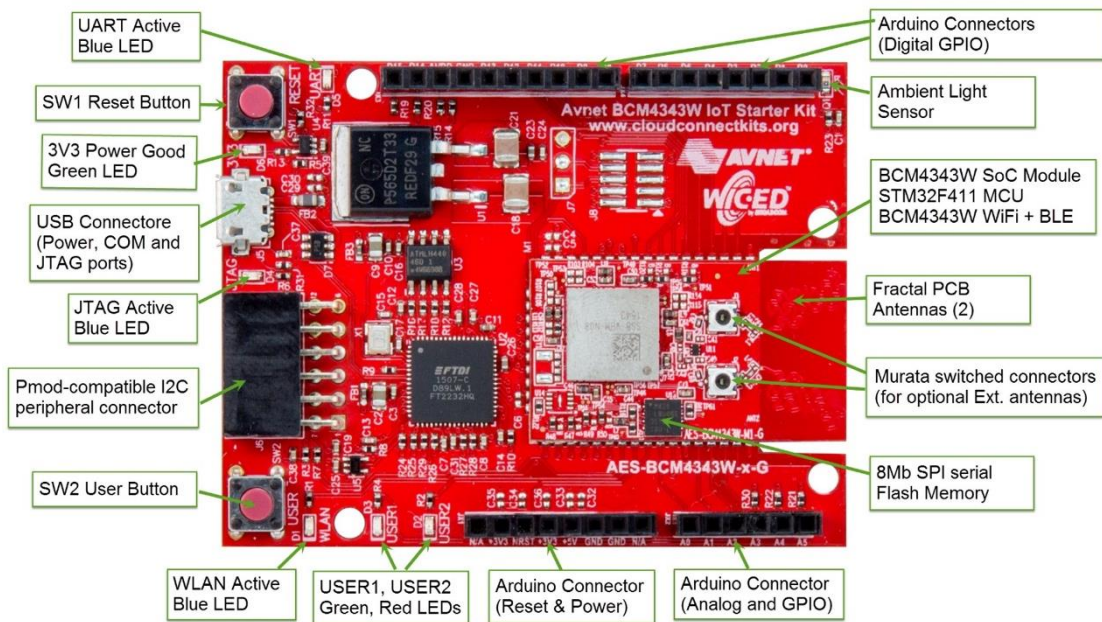


Figure 2 – BCM4343W IOT STARTER KIT – Feature Identification

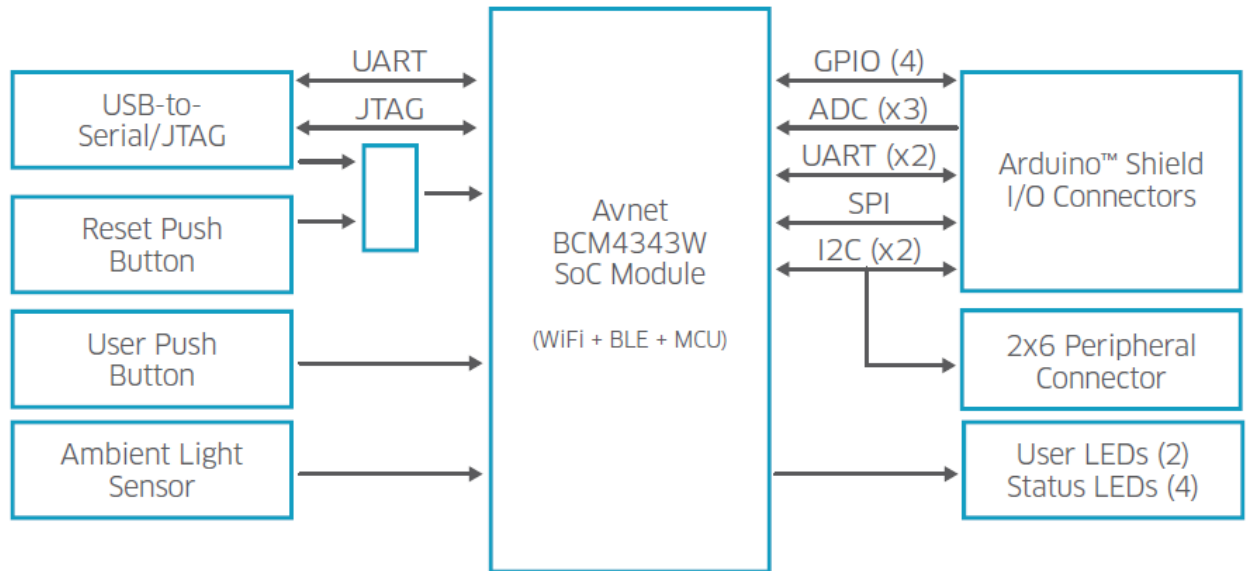


Figure 3 – BCM4343W IOT STARTER KIT – High-Level Block Diagram

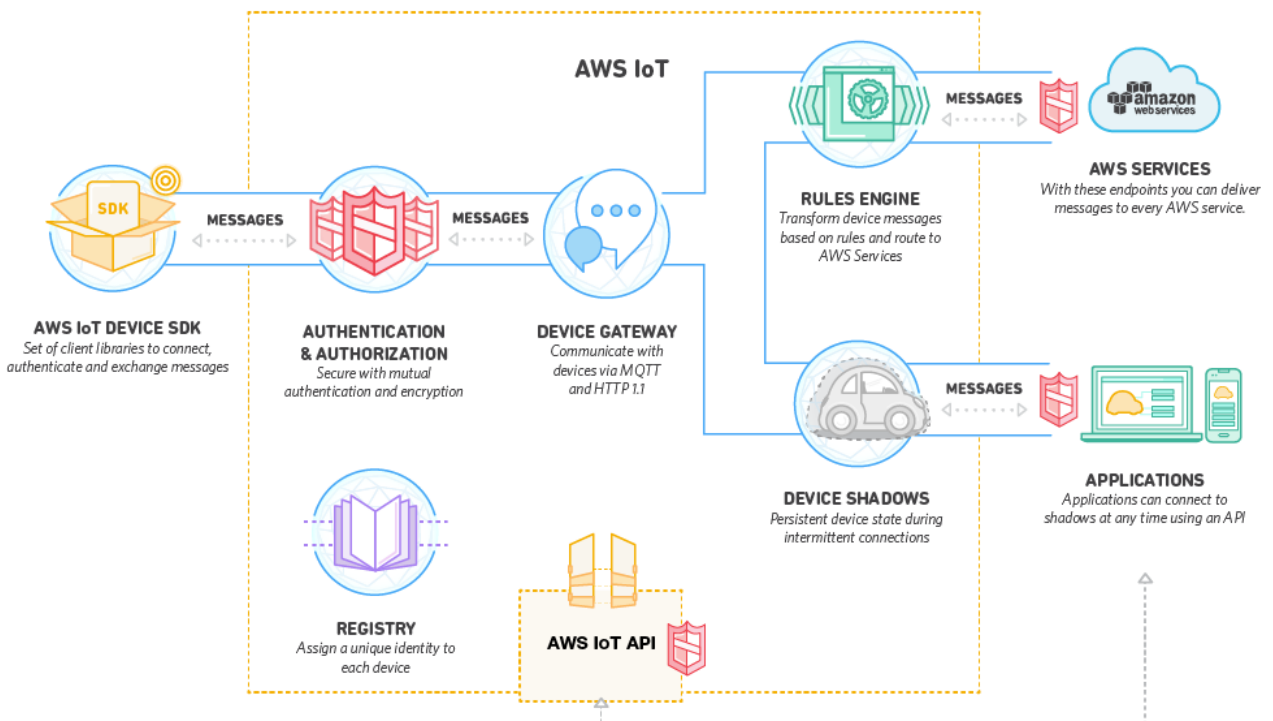


Figure 4 – AWS IoT

3.4 Overview: AWS IoT Service

AWS IoT is a new managed service that enables Internet-connected **THINGS** (sensors, devices, actuators, apps) to easily and securely interact with each other and the cloud.

A **THING / DEVICE** reports it's **STATE** by **PUBLISHING** messages to a **BROKER** through **TOPICS**. A **BROKER** delivers received messages to all clients **SUBSCRIBED** to that **TOPIC**.

THING SHADOW is a **JSON** document located in the cloud, used to store and retrieve current **STATE** info for a **THING**. It provides a persistent record of the physical device, which by design typically has only intermittent connection to the Internet.

RULES ENGINE evaluates inbound messages published to **AWS IoT**, then transforms and delivers these to AWS Services or External Endpoints (via **AWS SNS** and **AWS Lambda**)

3.5 Overview: Out-of-box Shadow Example

This exercises a number of key AWS IoT concepts:

- At launch of the Shadow App, the Starter Kit software retrieves its X.509 CERTIFICATE and PRIVATE KEY that you previously stored in the MCU's DCT flash memory.
- Using these credentials, it then establishes an authenticated, TLS secured connection with the specified MQTT MESSAGE BROKER.
- When the USER Pushbutton is pressed on the Starter Kit,
 - an MQTT protocol message is published to this MQTT Broker and
 - the SHADOW JSON description of USER1 LED status is toggled (ON or OFF)
- From the AWS IoT Console webpage, a User edit to the SHADOW record then demos the ability to remotely control the state of the physical USER1 LED on the Starter Kit

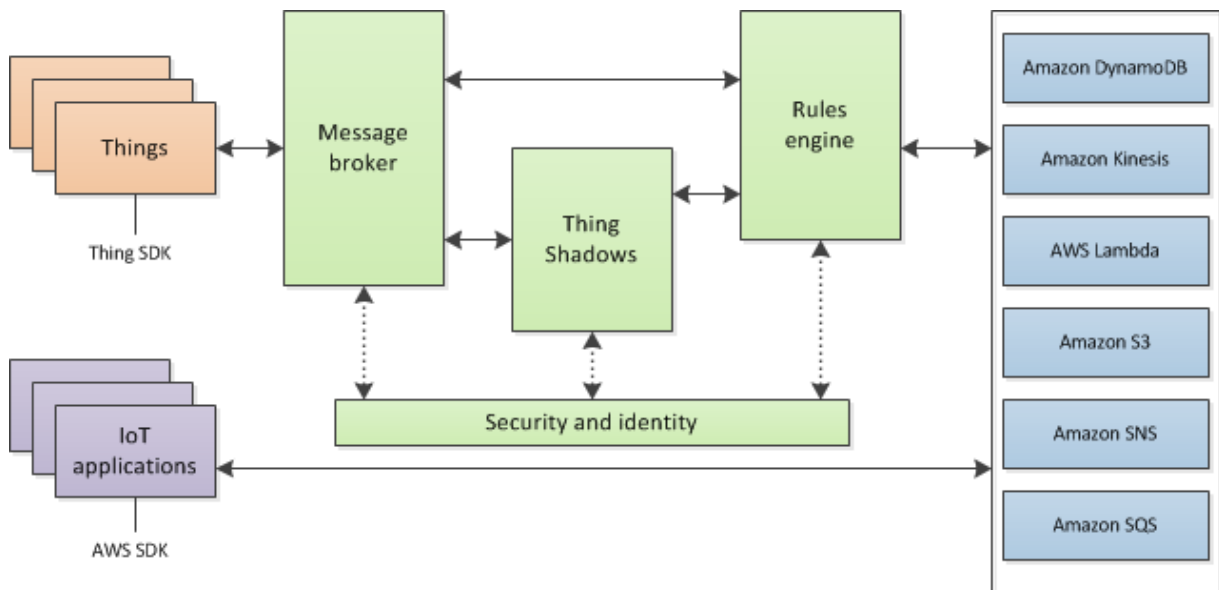


Figure 5 – Simplified view of AWS IoT

3.6 Overview: More Advanced Exercises

This document is currently being updated to exercise more advanced AWS IoT concepts, highlighting their utility in solving real world IoT challenges, as well as making simple code edits within Cypress's WICED SDK software development tool so that sensor data is also sent to the Cloud, eg.

- From AWS IoT Console: Set-up a RULE to action the AWS SNS service to send a text message to User's cellphone, each time the USER button is pressed on the Starter Kit.
- From WICED SDK: Edit the Shadow app so that sensor data (measured ambient light level) is sent to the cloud each time the USER pushbutton is pressed.
- A template for sampling an I2C sensor (via Pmod interface) and publishing this data to AWS IoT will also be provided

4 BCM4343W IoT Starter Kit Setup and Operation

4.1 AWS IoT Console Procedure

1. Go to Amazon **Web Services** website and register a new user account
<https://aws.amazon.com/iot>





Sign In or Create an AWS Account

What is your email (phone for mobile accounts)?

E-mail or mobile number:

☒ I am a new user.

☐ I am a returning user
and my password is:

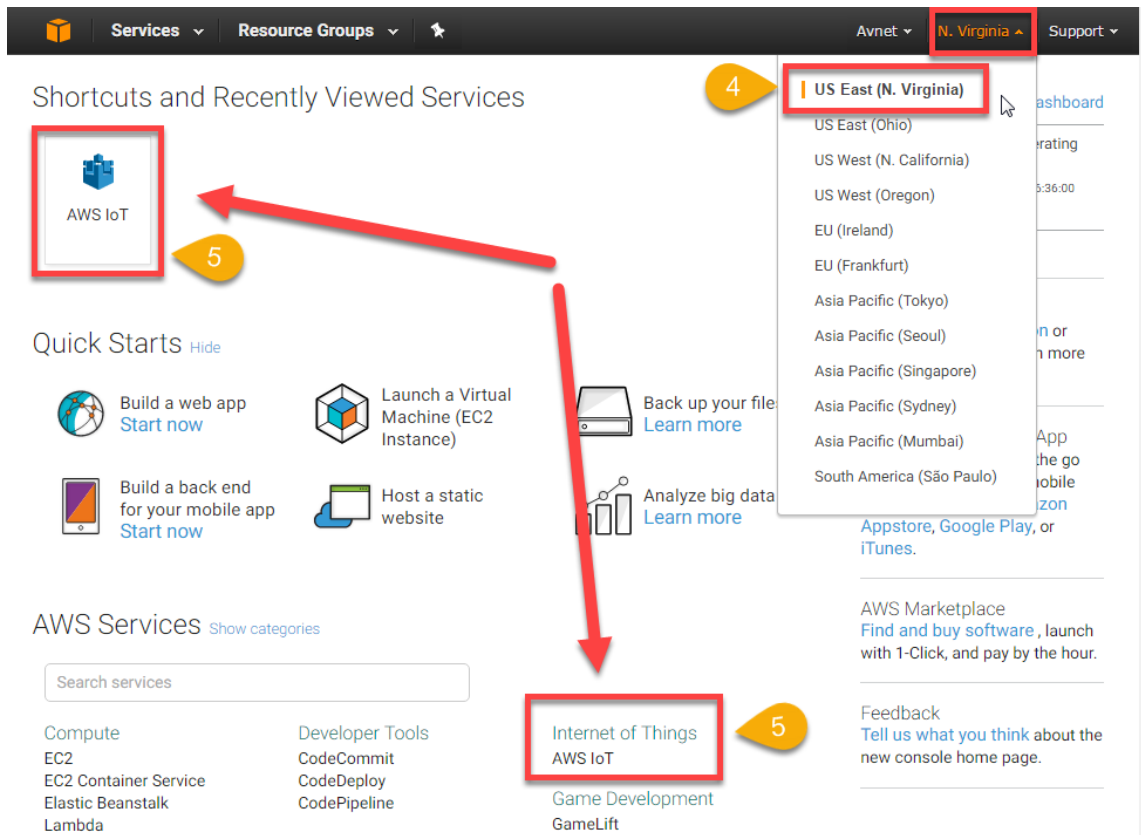
Sign in using our secure server

[Forgot your password?](#)



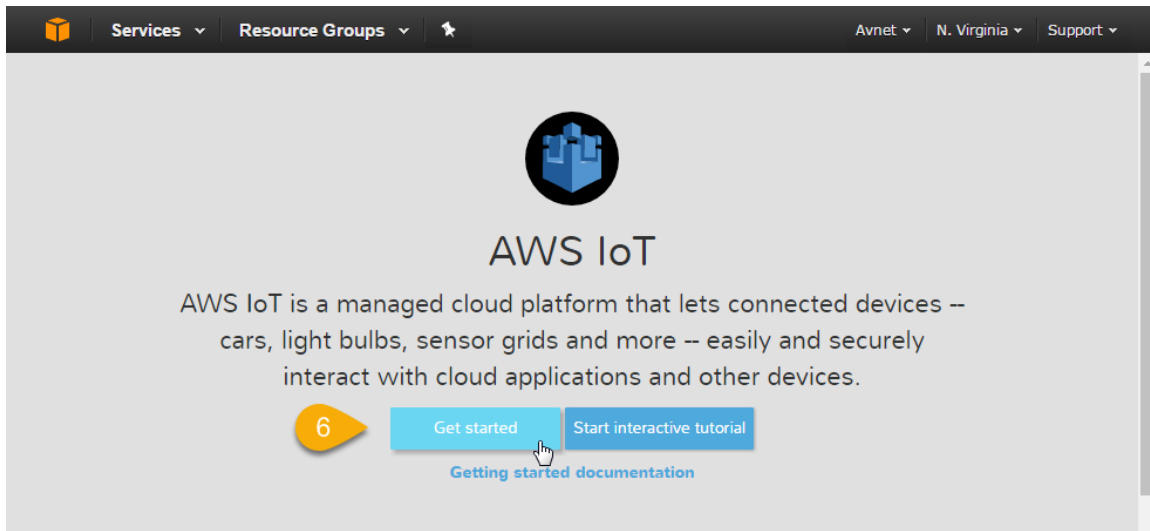
Learn more about [AWS Identity and Access Management](#) and [AWS Multi-Factor Authentication](#), features that provide additional security for your AWS Account. View full [AWS Free Usage Tier](#) offer terms.

2. A credit card number will be requested when registering but you will not incur any cost unless the AWS Free Tier limit is exceeded, ie. 250,000 messages per month
3. Once registered, sign-in to Amazon's general **AWS Management Console**.

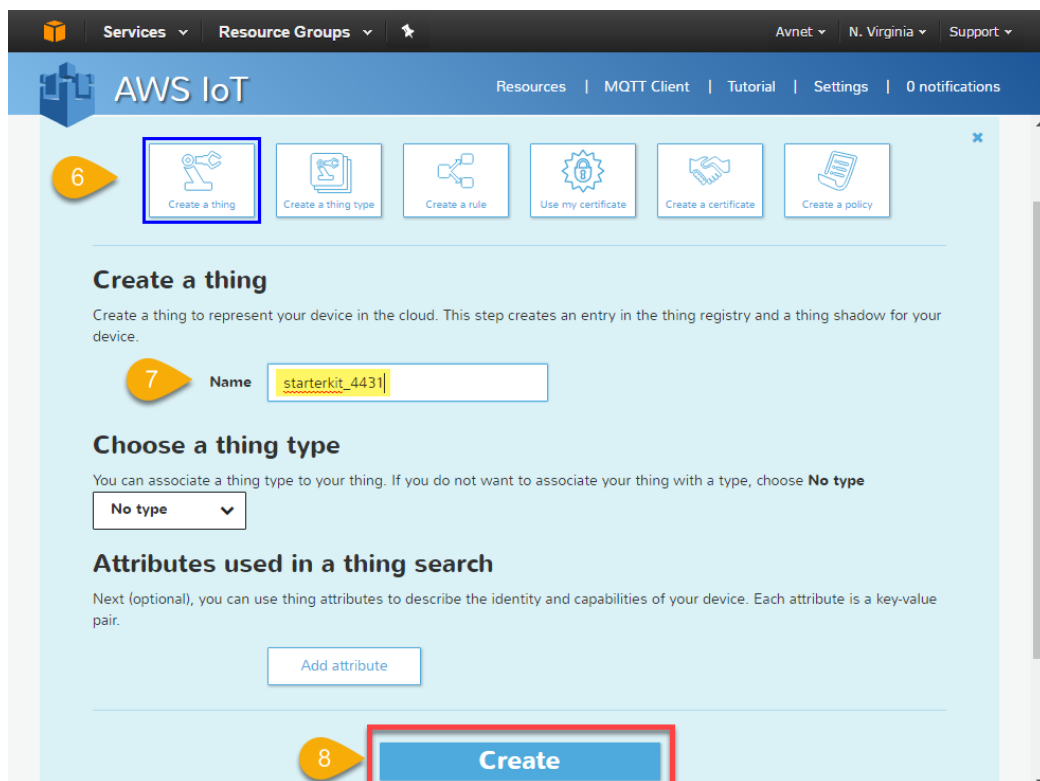


4. **Note:** Before going any further, ensure that the **US East** server region (N. Virginia) is selected!
(Correct the region by selecting **US East (N.Virginia)** from the drop-down list)
5. Click on either the **AWS IoT** icon or “AWS IoT” under the **Internet of Things** category to proceed to **AWS IoT Console**

6. Click on Get started followed by click on Create a Thing



7. Enter a unique name for your board. In the example shown, “starterkit_4431” was used (ie. includes last 4 digits from board’s serial number label)
8. Click on **Create**



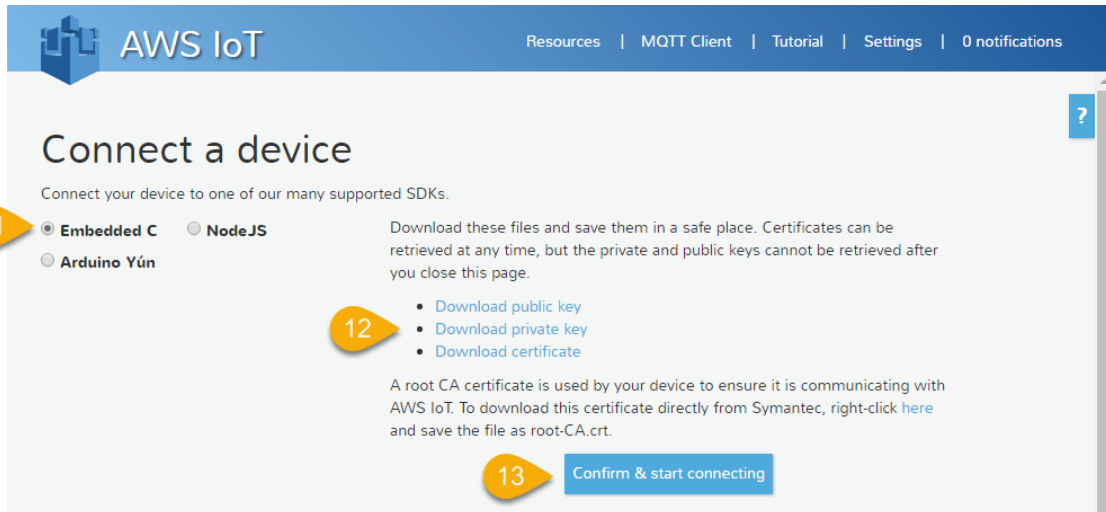
9. Click on View Thing
10. Click on Connect a Device

The screenshot displays the AWS IoT console interface. At the top, there's a navigation bar with 'Services' and 'Resource Groups' dropdowns, and a user profile section showing 'Avnet' and 'N. Virginia' region. Below this is the 'AWS IoT' header with links to 'Resources', 'MQTT Client', 'Tutorial', 'Settings', and '0 notifications'.

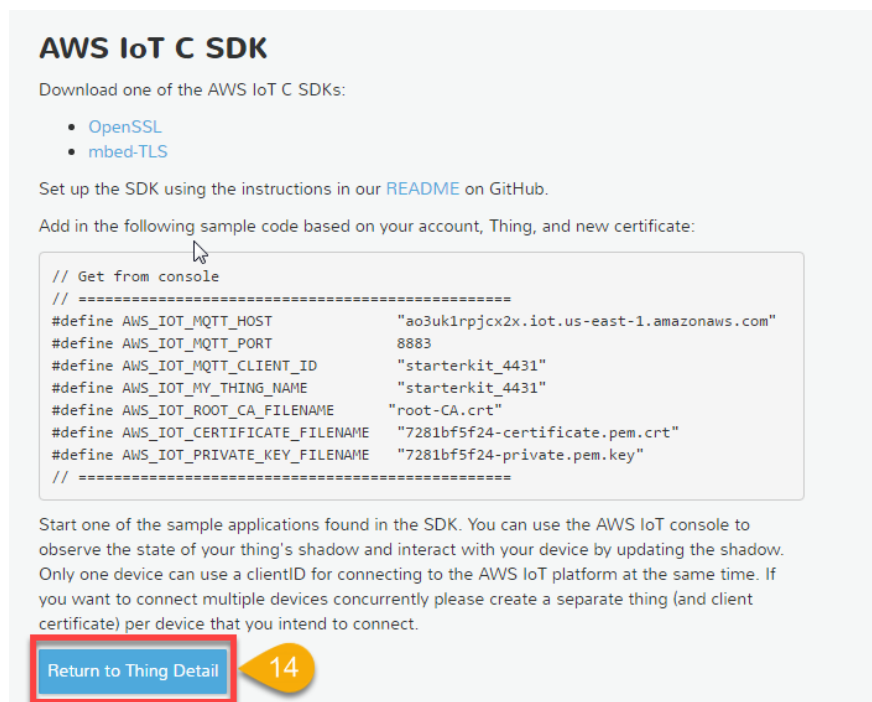
The main content area is divided into two panels. The left panel, titled 'Resources', contains a 'Close create panel' button and a 'Connect AWS IoT Button' link. It features a grid of six icons for creating various resources: 'Create a thing' (highlighted with a blue box), 'Create a thing type', 'Create a rule', 'Use my certificate', 'Create a certificate', and 'Create a policy'. Below the grid, a message states 'Your thing has been created.' and provides instructions on connecting a device or adding a rule. A yellow callout bubble with the number '9' points to a red-bordered 'View thing' button.

The right panel, titled 'Detail', shows the configuration for the 'starterkit_4431' thing. It includes tabs for 'Learn more', 'Detail' (selected), 'Update shadow', and 'Edit'. The details listed are: Name (starterkit_4431), REST API endpoint (https://ao3uk1rpjcx2x.iot.us-east-1.amazonaws.com/things/starterkit_4431/shadow), Thing type (No type), MQTT topic (Saws/things/starterkit_4431/shadow/update), Last update (No state), Attributes used in a thing search (None), and Linked certificates (None). A yellow callout bubble with the number '10' points to a red-bordered 'Connect a device' button at the bottom right.

11. Click on **Embedded C** followed by click on **Generate Certificate and Policy**
12. Click on each of the three download links. Use Windows file-explorer to move the downloaded keys and certificate file to a suitable folder, eg. **IAWS_IoT_Credentials**
13. Click on **Confirm and Start Connecting**

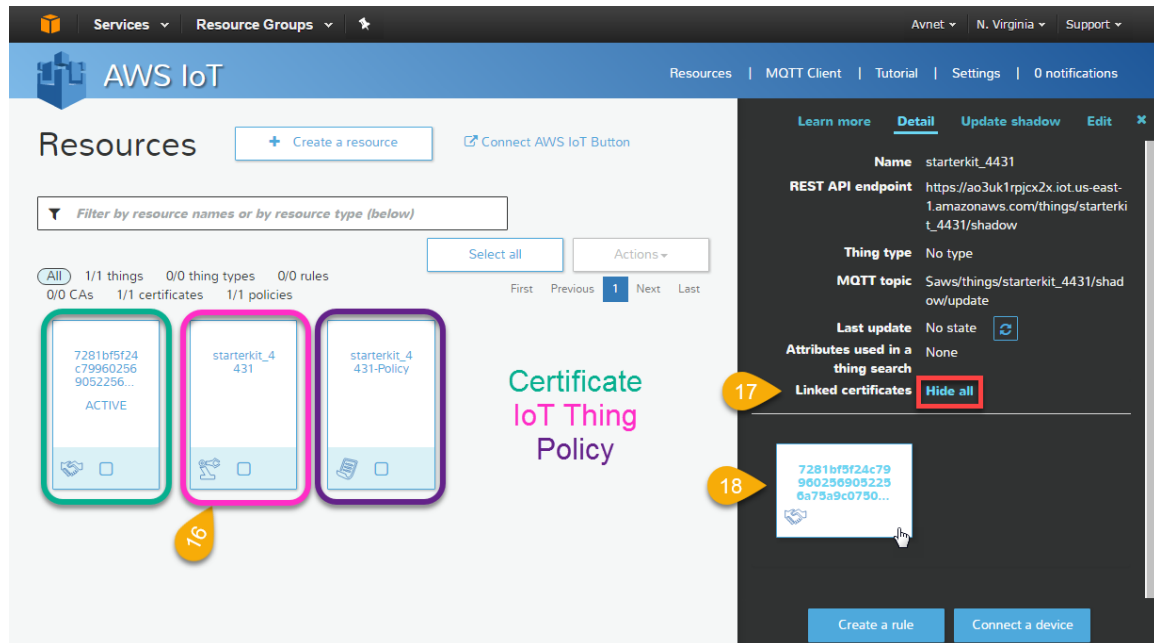


14. Take a screenshot of the reported `#define` values (is a useful reference) then click on **Return to Thing Detail**



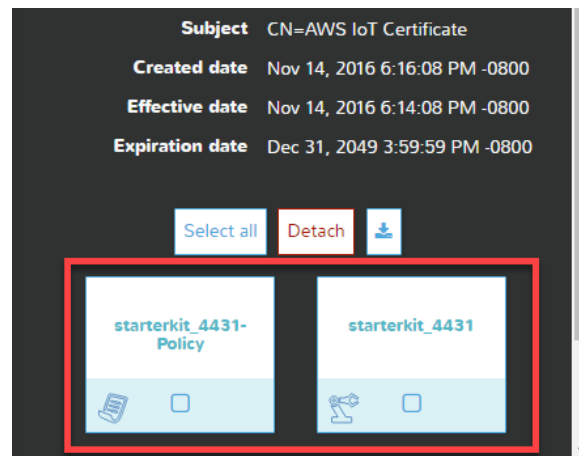
15. At this stage your Starter Kit has been fully provisioned with attachment of an active authentication **certificate** and **policy**. Certificates are used to link together the Policy and the Thing for secure authentication with AWS IoT.
16. This can be checked at any time by clicking on your Thing (eg. `starterkit_4431`)

17. Followed by clicking on the Linked Certificate Show All link (which automatically turns to “Hide All” after clicking)



18. Next, click on the certificate shown. Viewed from this perspective you should see:

- The Policy (eg. starterkit_0040_Policy) , and
- The Thing (eg. starterkit_0040)



19. Your setup in AWS IoT Console is now complete!
20. Next, proceed to the board configuration: the writing of certificate, key and network configuration details into the flash memory of your BCM4343W IoT Starter Kit...

PS: You can return to the AWS IoT Console at any time by going to:

<https://console.aws.amazon.com/iot/home?region=us-east-1>

BCM4343W IoT Starter Kit Board Configuration

21. Connect the provided USB cable from the BCM4343W IoT Starter Kit to your computer, the 3V3 power LED will illuminate.
22. Connect your computer's Wi-Fi to **WICED_AWS** (the SSID of WICED module in SoftAP mode). When prompted, enter a password of **12345678**
23. Enter IP address **192.168.0.1** into your internet browser, this will take you to a webpage served by the board, this is where you define the Thing Name, upload your AWS certificate and Private Key, as well as selecting a suitable Wireless Access Point.

WICED™ AWS IOT Service

Configuration Settings :

24 Thing Name Save Settings

Upload Certificate and Key :

25 Choose File No file chosen Upload Certificate 25

26 Choose File No file chosen Upload key 26

27 Wi-Fi Setup >

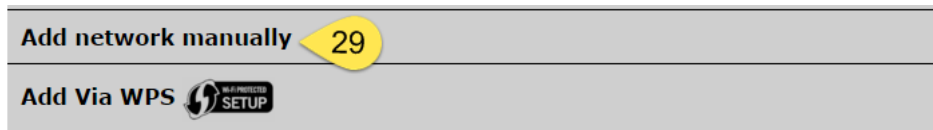
24. Enter the same **Thing Name** that you used in AWS IoT Console (eg. **starterkit_4431**), Follow this by clicking on the adjacent Thing Name **Save Settings** button.
25. Choose the AWS certificate file that was attached in AWS IoT Console to your Thing (eg. **6ae5bf0d7c-certificate.pem.crt** in the example slides previously shown) Follow this by clicking on the adjacent **Upload Certificate** button.
26. Choose the AWS private key file that was downloaded at same time as the certificate for your Thing (eg. **6ae5bf0d7c-private.pem.key** in the example previously shown) Follow this by clicking on the adjacent **Upload Key** button.

(A completed progress bar and “Transfer Complete” message should display for both the certificate and key uploads)

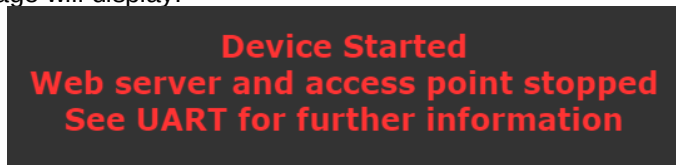
27. With the AWS credentials now written to DCT flash memory, proceed to configuring your network settings. Click on the Wi-Fi Set button, this will take you to a new webpage that displays a listing of SSIDs reported from the module’s local scan



28. Select the **SSID** of the desired Wireless A/P, enter the applicable **Password**, then click on the **Connect** button.
29. If attempting to connect to a Wireless A/P that does not broadcast it’s SSID, use the **Add Network manually** option that is listed...



30. After connection is made with the selected Wireless A/P, the webpage will darken and the following message will display:



31. Your Starter Kit is now configured for communication with the AWS IoT Services and the board exits configuration mode and commences running the Shadow App.
(Board resets will now auto-connect to selected Wireless A/P and launch the App)
32. If needing to re-upload different AWS certificate and key, or connect to a different Wireless A/P, perform a hard reset by holding **SW2** for 5 seconds during Reset
(this clears the module's DCT memory and again launches the SoftAP mode)

4.2 Simple Exercise: Run the Out-of-Box “Thing Shadow” Application

33. To commence testing the Shadow application, your internet browser needs to be displaying the AWS IoT Console page, with your Starter Kit “Thing” selected. The goal is to securely transfer data from your device into the AWS portal, as boxed on the right of the image below.

The screenshot shows the AWS IoT console interface. On the left, the 'Resources' page lists three things: '7281bf5f24c79960256...', 'starterkit_4431', and 'starterkit_4431-Policy'. The 'starterkit_4431' resource is highlighted with a red box. A red arrow points from this box to the 'Detail' view on the right. The 'Detail' view shows the following information:

- Name:** starterkit_4431
- Endpoint:** https://ao3uk1rpjcx2x.iot.us-east-1.amazonaws.com/things/starterkit_4431/shadow
- Thing type:** No type
- MQTT topic:** Saws/things/starterkit_4431/shadow/update
- Last update:** 1 minute ago
- Attributes used in a thing search:** None
- Linked certificates:** Show all
- Shadow status:** In sync
- Shadow version:** 1
- Shadow state:**

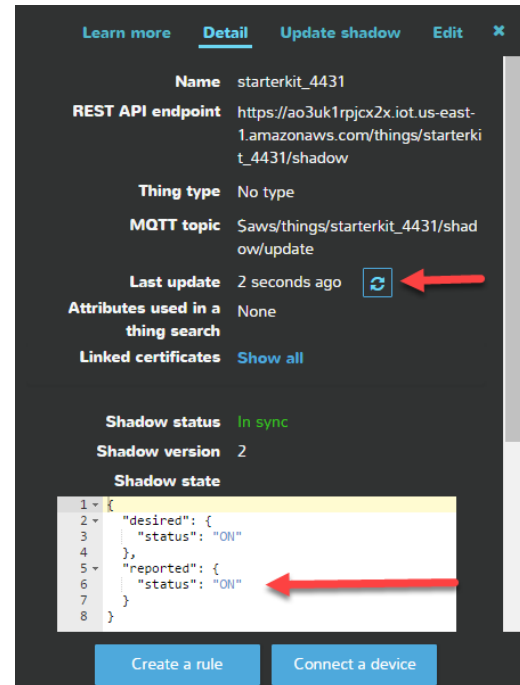
```
1 {
2   "reported": {
3     "status": "OFF"
4   }
5 }
```

34. Now press the **SW2** User button on the Starter Kit.
This **turns-on USER1** green LED and updates the Shadow state to “On”

To check Shadow state, view this in the right sidebar
“Thing” screen of the **AWS IoT Console**
 (- click refresh button to see the changed state)

35. Press **SW2** again, this **turns-off** USER1 green LED and updates Shadow state to **“Off”**
 (- click refresh button to see the changed state)

36. Repeatedly pressing SW2 increments State Version and toggles on/off the JSON descriptor of the Shadow status.
 Note: Screen updates are delayed!
 For immediate updates, click the refresh button



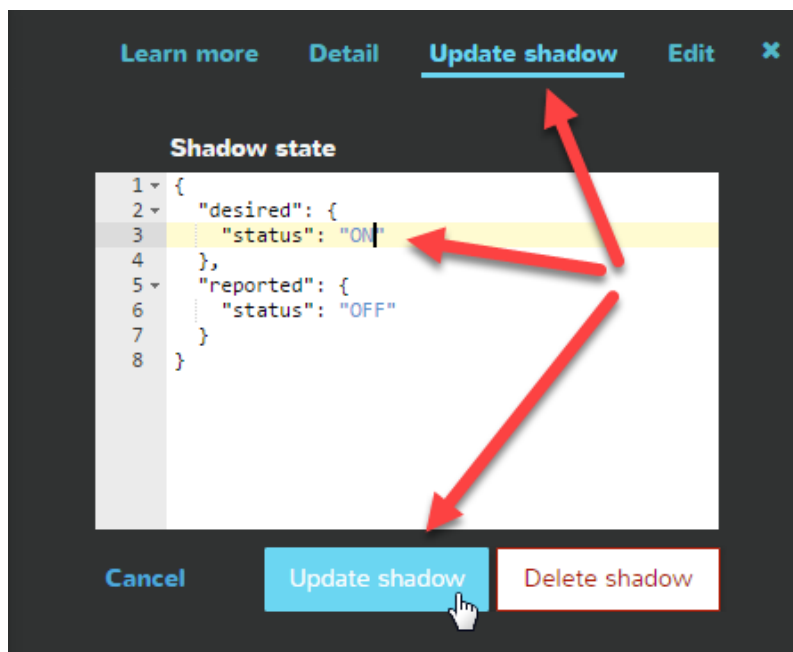
Note: Though not essential to this application, it is helpful to have a Serial Console open in order to view the various status messages from the microcontroller as connection is made with the MQTT broker and each time the USER button is pressed. (The USB JTAG and UART Windows drivers are installed automatically during WICED SDK installation)

```
COM157:115200baud - Tera Term VT
File Edit Setup Control Window Help
[MQTT] Connecting to broker 52.6.53.137 ...
Thing Name: starterkit_0040
Shadow State Topic: $aws/things/starterkit_0040/shadow/update
Shadow Delta Topic: $aws/things/starterkit_0040/shadow/update/delta
Reading the certificate and private key from DCT...
[MQTT] Connecting to MQTT Broker...
Reading the certificate and private key from DCT...
[MQTT] Connecting to MQTT Broker...
[MQTT] Successfully connected MQTT Broker
Publish SUCCEEDED for topic [$aws/things/starterkit_0040/shadow/update]
Subscribe SUCCEEDED for topic [$aws/things/starterkit_0040/shadow/update/delta]
Button is pressed
Publish SUCCEEDED for topic [$aws/things/starterkit_0040/shadow/update]
Button is pressed
Publish SUCCEEDED for topic [$aws/things/starterkit_0040/shadow/update]
```

The Starter Kit **“Publishes”** its status to the **update topic**
 \$aws/things/starter_kit_4431/shadow/update/

The Starter Kit “**Subscribes**” to (and takes instruction from) the **delta topic**
\$aws/things/starter_kit_4431/shadow/update/delta

37. Next we will remotely update the physical USER1 LED on the Starter Kit, by publishing to the delta topic, from the IoT Console sidebar in your browser.
38. If the User1 LED is currently illuminated, press the User button one more time to turn it OFF. Using the “Thing” screen of the AWS IoT Console still displayed in your browser, select the “Update Shadow” tab and edit the value of the desired status to from “OFF” to “ON” then follow this by clicking Update Shadow
39. On clicking this button, the Shadow Thing immediately publishes a JSON update message to the Delta Topic. The Starter Kit which is a Subscriber to the Delta Topic, decodes this message and updates it’s USER1 LED accordingly.



4.3 Advanced Exercise: Sending Email and SMS Messages

We have completed the basic exercise, verifying Publish and Subscribe operation between the Starter Kit “Thing” and it’s persistent cloud-based Shadow JSON record.

Next we will set-up **AWS SNS (Simple Notification Service)** plus two **Rules** in **AWS IoT** to trigger two different **SNS** services:

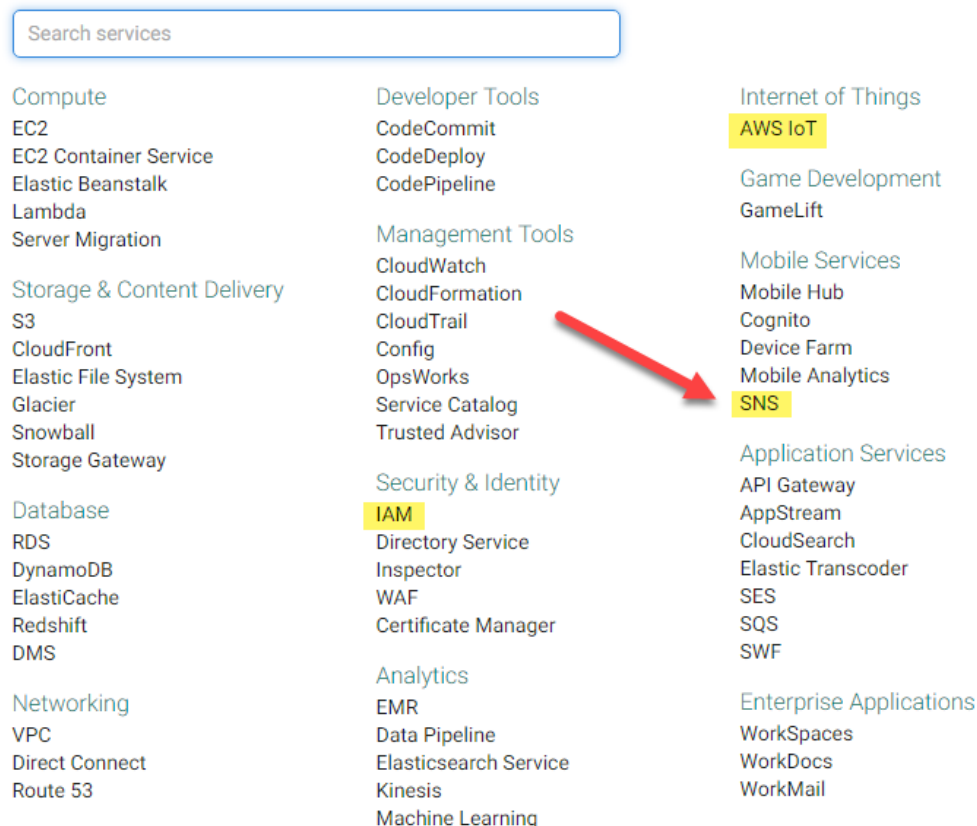
- **EMAIL message** each time the User pushbutton is pressed on Starter Kit
- **SMS text message** whenever User1 LED is remotely updated from AWS IoT Console

4.3.1 AWS SNS: Setup EMAIL Messaging Topic

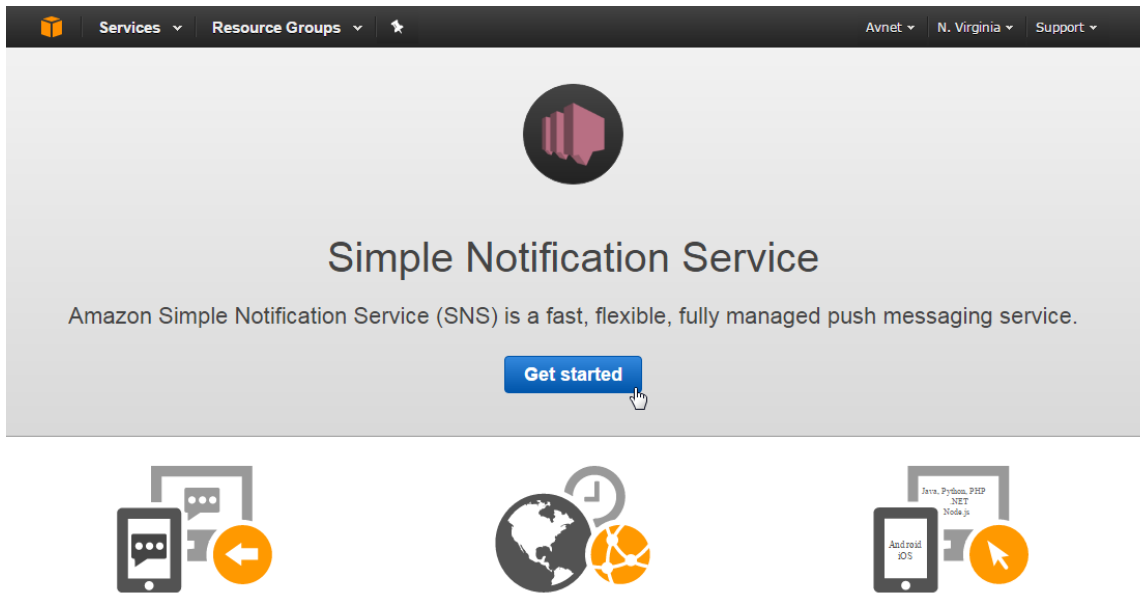
Up to this point, AWS IoT was the only service used from Amazon Web Services. For this next exercise, two additional aspects of AWS will be utilized:

- **SNS** (Notification delivery service: where we define topics and subscriptions)
- **IAM** (Identity and Access Management: to define role and policy)

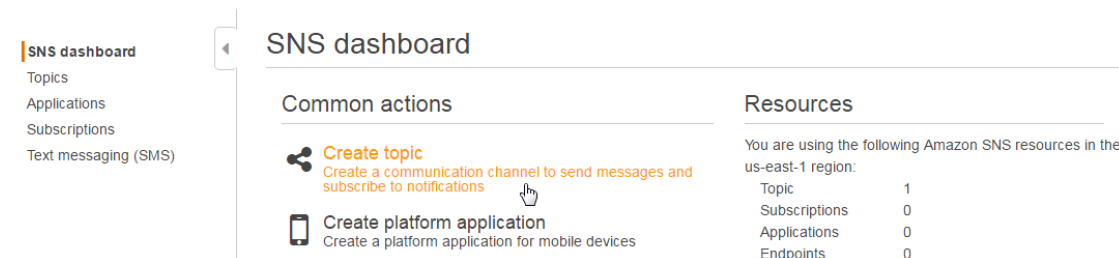
40. Return to the main AWS Console and scroll down to select SNS



41. In SNS, click on **Get started**



42. Click **Create Topic**



43. Fill-in **Topic Name** and **Display Name** as shown, then click **Create Topic**
(Display Name is a short label you provide to identify the source of the messages)

Create new topic

A topic name will be used to create a permanent unique identifier called an Amazon Resource Name (ARN).

Topic name

send_email_topic

Display name

AWS-IOT>

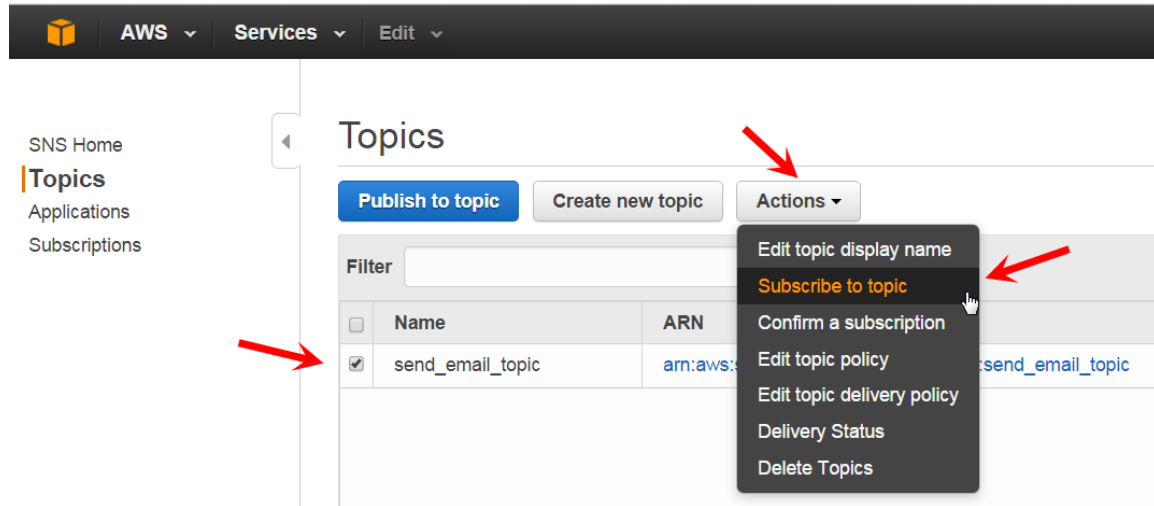
Cancel

Create topic

To this **send_email_topic** we now need to add a subscription that defines the service and endpoint where the email will be sent to

4.3.2 AWS SNS: Setup EMAIL Subscription to EMAIL Topic

44. Back at the SNS Topics page, check the box next to **send_email_topic** that has just been created
45. Click on **Actions** then **Subscribe to topic**



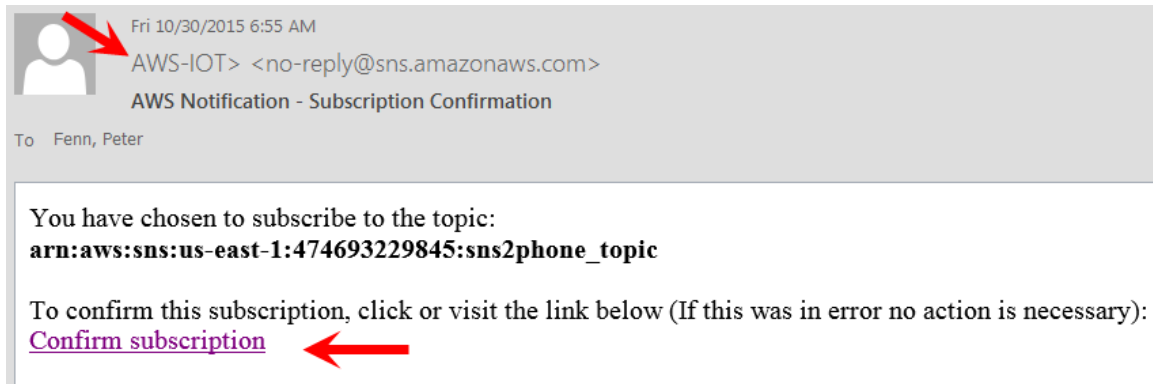
The **Create Subscription** dialog box will now open...

For **Protocol** : select **Email**

For **Endpoint** : enter the **email address** that you'd like messages sent to

The 'Create Subscription' dialog box is shown. It has three input fields: 'Topic ARN' (pre-filled with 'arn:aws:sns:us-east-1:474693229845:send_email_topic'), 'Protocol' (a dropdown menu set to 'Email'), and 'Endpoint' (pre-filled with 'username@gmail.com'). At the bottom right, there are 'Cancel' and 'Create Subscription' buttons. Red arrows point to the 'Protocol' dropdown, the 'Endpoint' field, and the 'Create Subscription' button.

46. Click on **Create Subscription**
47. Click on the Subscriptions on left side of your screen, the new subscription will be listed with a "Pending Confirmation" label.
48. Check your email inbox and click on the **Confirm subscription** link in the subscription confirmation email from **AWS-IOT>** that you should have now received



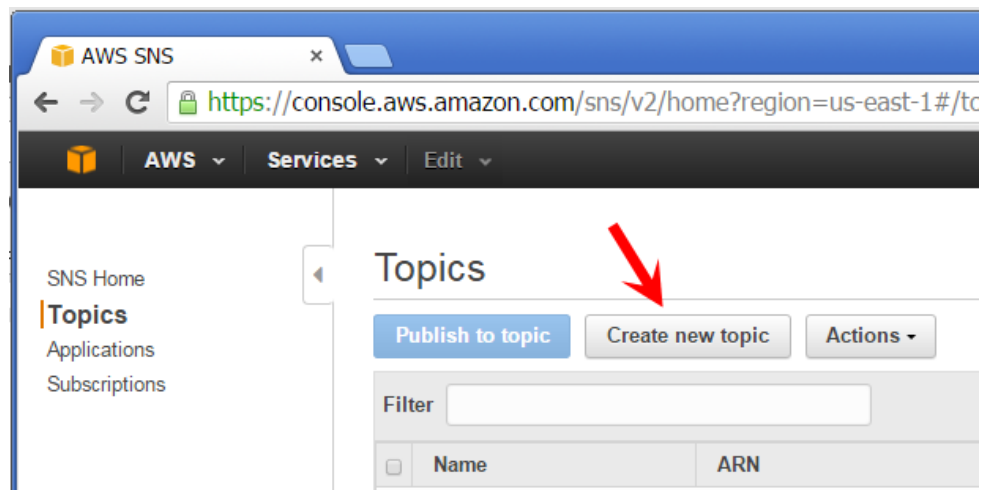
49. Refresh your SNS browser screen. Your subscription should now list as shown below

Subscriptions

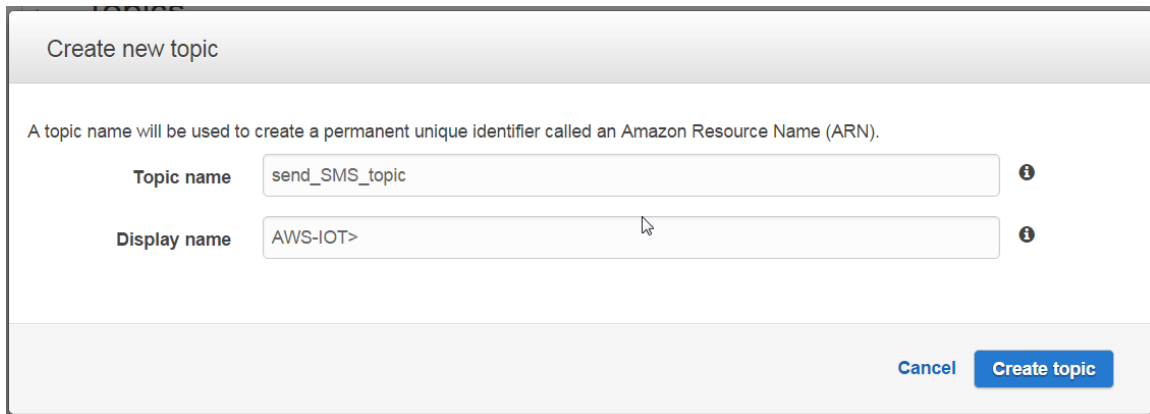
Create Subscription Request confirmations Actions				
Filter				
☐	Subscription ARN	Protocol	Endpoint	Topic ARN
☐	arn:aws:sns:us-east-1:474693229845:sns2phone_topic:a29f6475-47c...	email	peter.fenn@...	arn:aws:sns:us-...

4.3.3 AWS SNS: Setup SMS Messaging Topic

50. We will now set-up a 2nd SNS topic for SMS messaging.
As before, select **Topics**, then **Create New Topic**



51. Fill-in **Topic Name** and **Display Name** as shown, then click **Create Topic** (the same Display Name label as before can be used to ID the source of the messages)



Create new topic

A topic name will be used to create a permanent unique identifier called an Amazon Resource Name (ARN).

Topic name

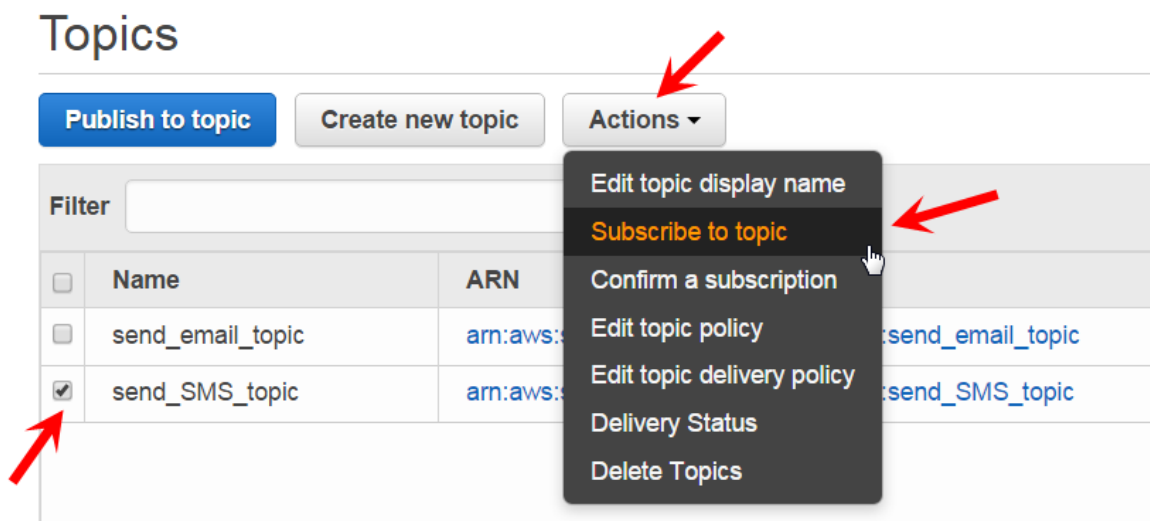
Display name

Cancel Create topic

To this **send_SMS_topic** we will now add a **subscription** that defines the service protocol and the cellphone number where the SMS will be sent to

4.3.4 AWS SNS: Setup SMS Subscription to SMS Topic

52. Back at the SNS Topics page, check the box next to **send_SMS_topic** that has just been created
53. Click on **Actions** then **Subscribe to topic**



Topics

Publish to topic Create new topic Actions

Filter

☐	Name	ARN
☐	send_email_topic	arn:aws:sns:us-east-1:123456789012:send_email_topic
☒	send_SMS_topic	arn:aws:sns:us-east-1:123456789012:send_SMS_topic

- Edit topic display name
- Subscribe to topic
- Confirm a subscription
- Edit topic policy
- Edit topic delivery policy
- Delivery Status
- Delete Topics

54. The **Create Subscription** dialog box will now open...
- For **Protocol**: select **SMS**
- For **Endpoint**: enter cellphone number where you'd like the messages sent

Create Subscription

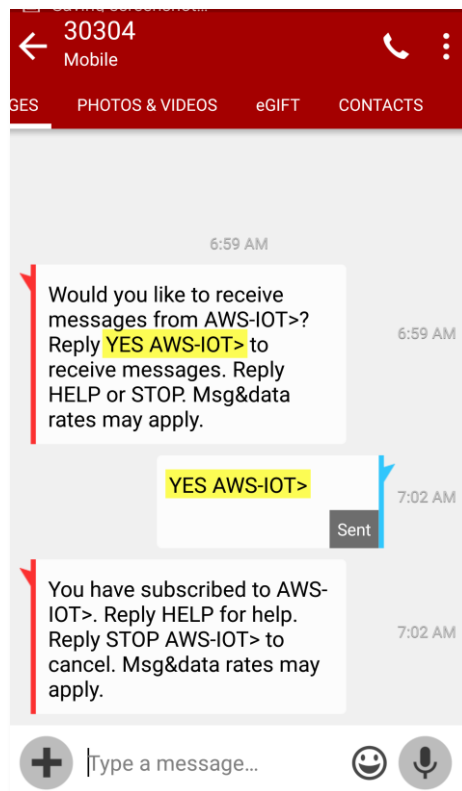
Topic ARN

Protocol **SMS**

Endpoint

[Cancel](#) [Create Subscription](#)

55. Click on **Create Subscription**
56. Select Subscriptions on left side of your screen, the new SMS subscription will be listed with a "Pending Confirmation" label.
57. Check your cellphone for a subscription confirmation text message from **AWS-IOT>** and follow the on-screen directions...



4.3.5 AWS IoT: Setup Rule1: Send EMAIL on User Button Press

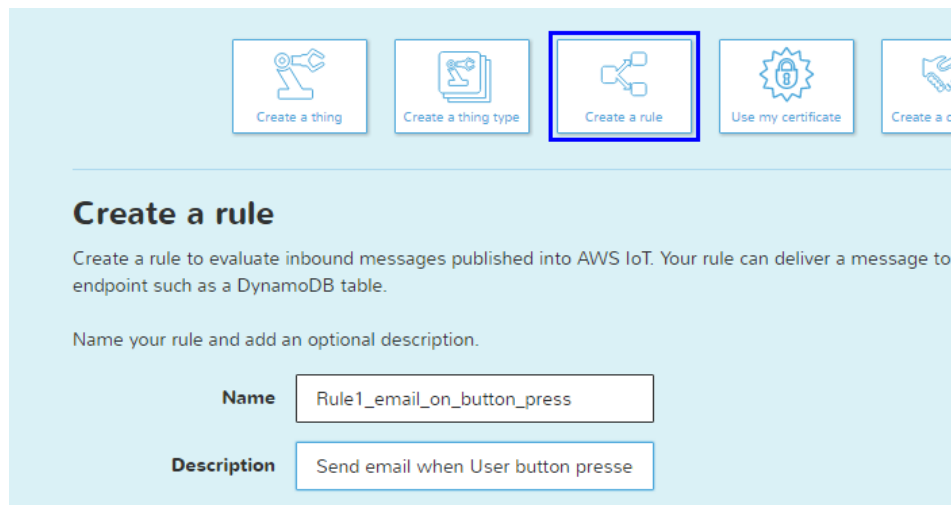
58. Clear your AWS IoT Console by selecting **Rules** and deselecting the other resources

59. Click on **Create a resource**



60. Click on **Create a Rule** (you can also get to this step via the “Create a Rule” button at the bottom of Thing Shadow’s detail screen)

61. Enter/select the values shown below... (Topic filter will be in the following format, with your thing name in lieu of ‘starterkit_XXXX’ - \$aws/things/starterkit_XXXX/shadow/update/+))



Indicate the source of the messages you want to process with this rule.

Rule query statement

SQL version

Attribute

Topic filter

Condition

Select one or more actions to take when an inbound message matches the rule query. Actions can include storing the message in a database, calling a Lambda function, or sending notifications. (* required)

Choose an action

This will push the message to a SNS topic.

***SNS target**

Message format

Choose or create a role to grant AWS IoT access to the selected Amazon SNS resource to perform this action.

***Role name**

[Cancel](#) [Add action](#)

62. For the Role Name, click on **Create a new Role** which will open-up the following screen to create a new **IAM Role** to allow AWS IoT to publish to **SNS**

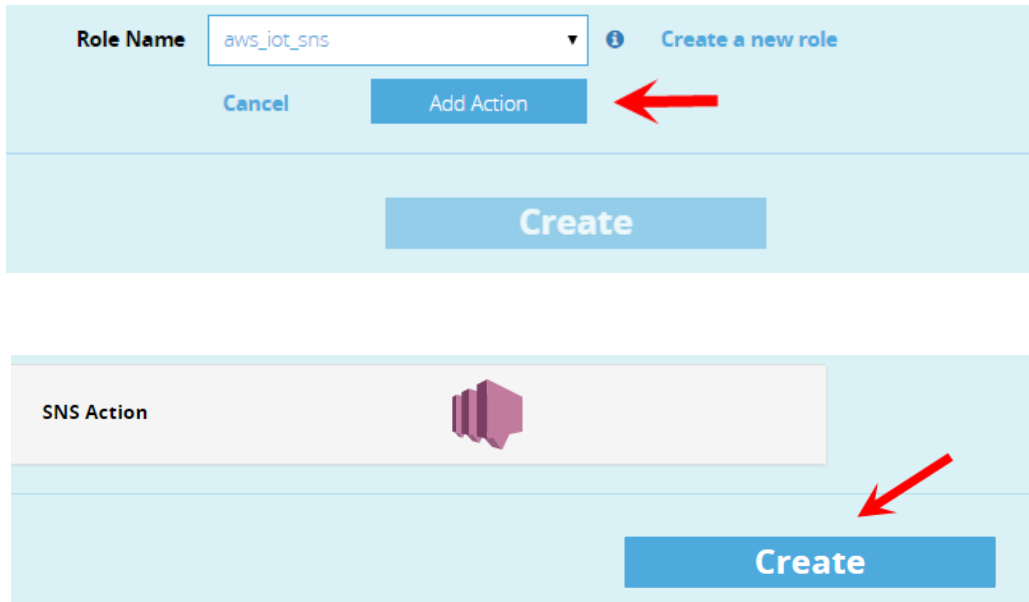
Create a new role

Role name

[Create](#)

63. For simplicity, the same IAM Role will be used for both email and SMS

64. Click on **Add Action**, and then click on **Create**



Role Name: ⓘ Create a new role

Cancel Add Action

Create

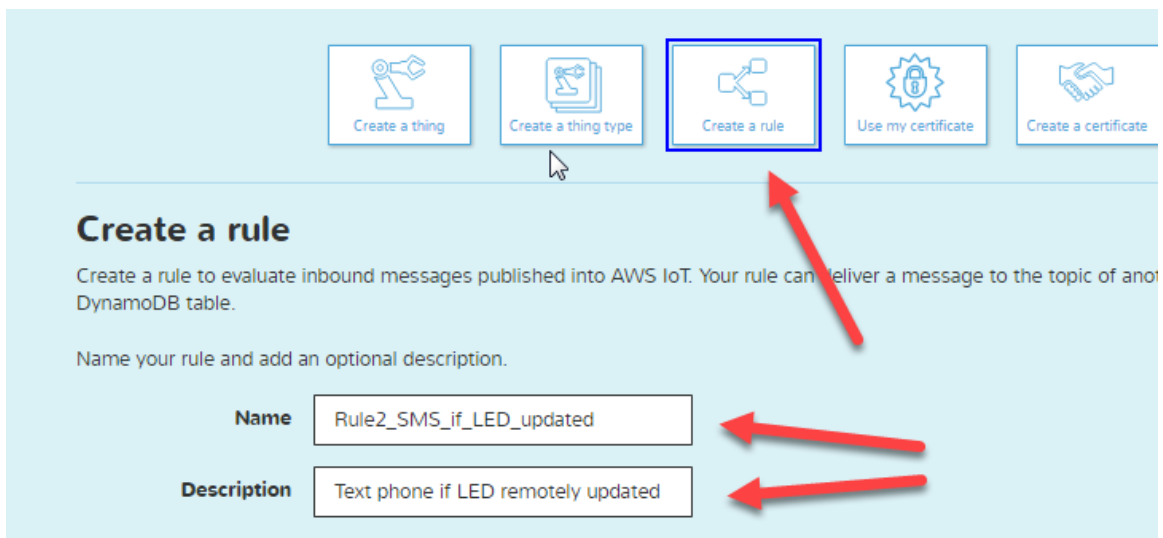
SNS Action 

Create

4.3.6 AWS IoT: Setup Rule2: Send SMS on Remote Change of User1 LED

65. From your AWS IoT Console again click on **Create a resource**, then...

66. Click on **Create a Rule** and enter the values shown below (same topic as the last rule) ...



Create a thing Create a thing type Create a rule Use my certificate Create a certificate

Create a rule

Create a rule to evaluate inbound messages published into AWS IoT. Your rule can deliver a message to the topic of another IoT thing, an Amazon SNS topic, or a DynamoDB table.

Name your rule and add an optional description.

Name

Description

Indicate the source of the messages you want to process with this rule.

Rule query statement

SQL version

Attribute ⓘ

Topic filter ⓘ

Condition ⓘ

Select one or more actions to take when an inbound message matches the rule query. Actions can include storing the message in a database, calling a Lambda function, or sending notifications. (* required)

Choose an action ⓘ

This will push the message to a SNS topic.

***SNS target** ⓘ [Create a new resource](#) ⓘ

Message format ⓘ

Choose or create a role to grant AWS IoT access to the selected Amazon SNS resource to perform this action.

***Role name** ⓘ [Create a new role](#)

[Cancel](#) [Add action](#)

67. Click on **Add Action**, and then click on **Create**

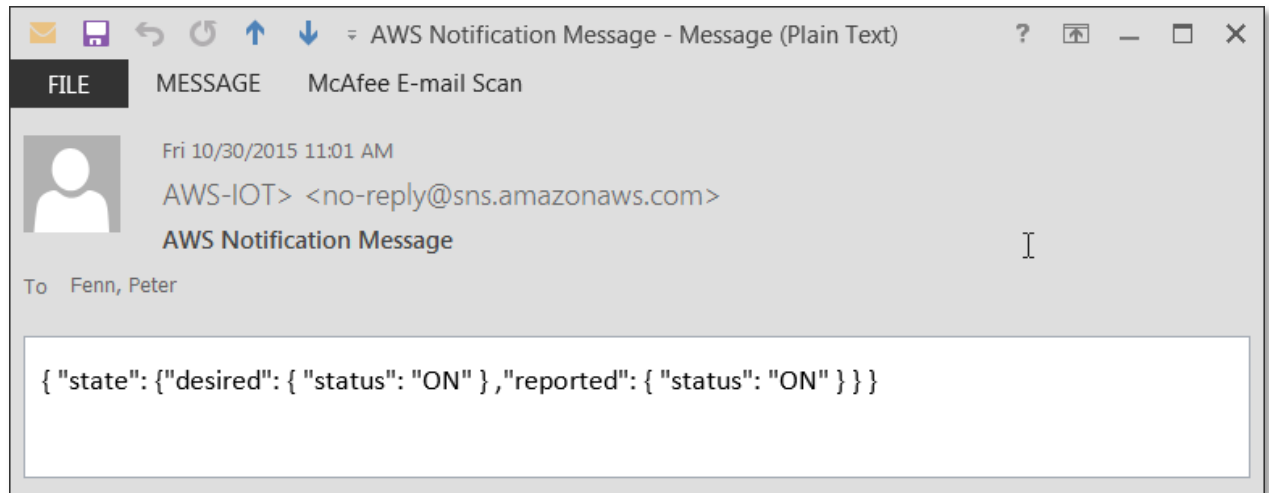
SNS Action ⓘ

[Create](#)

4.3.7 AWS IoT: Testing Email Messaging (Rule1)

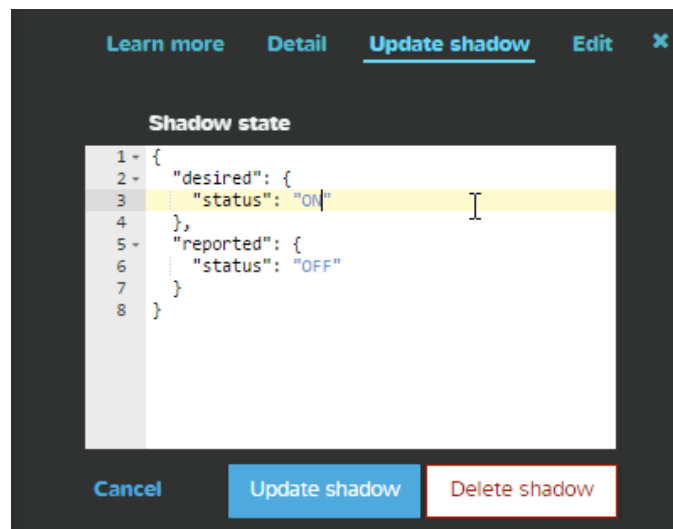
68. Press the User pushbutton on the Starter Kit.

An email such as the following should arrive in the specified email inbox...

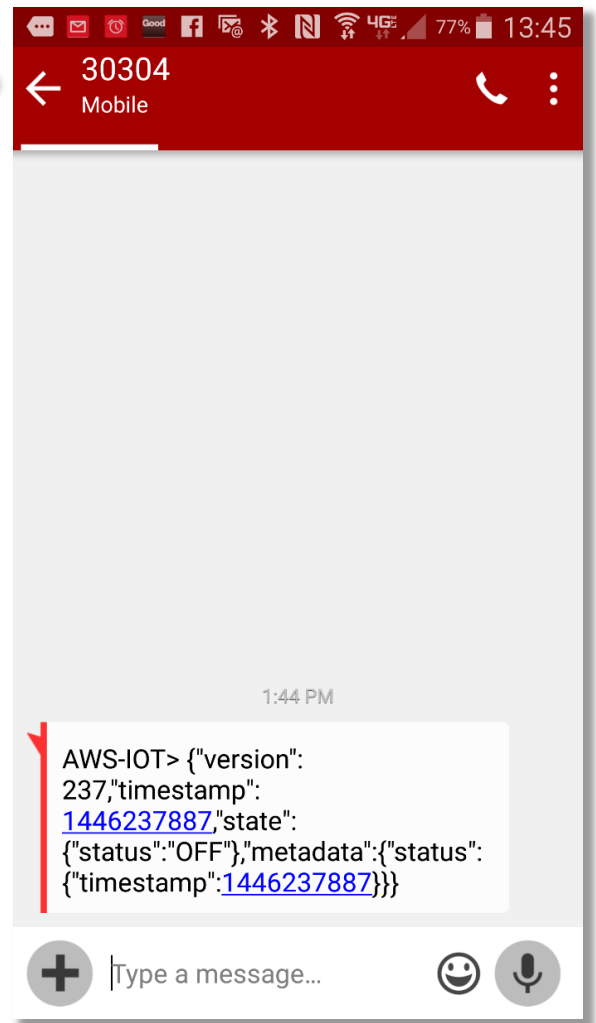
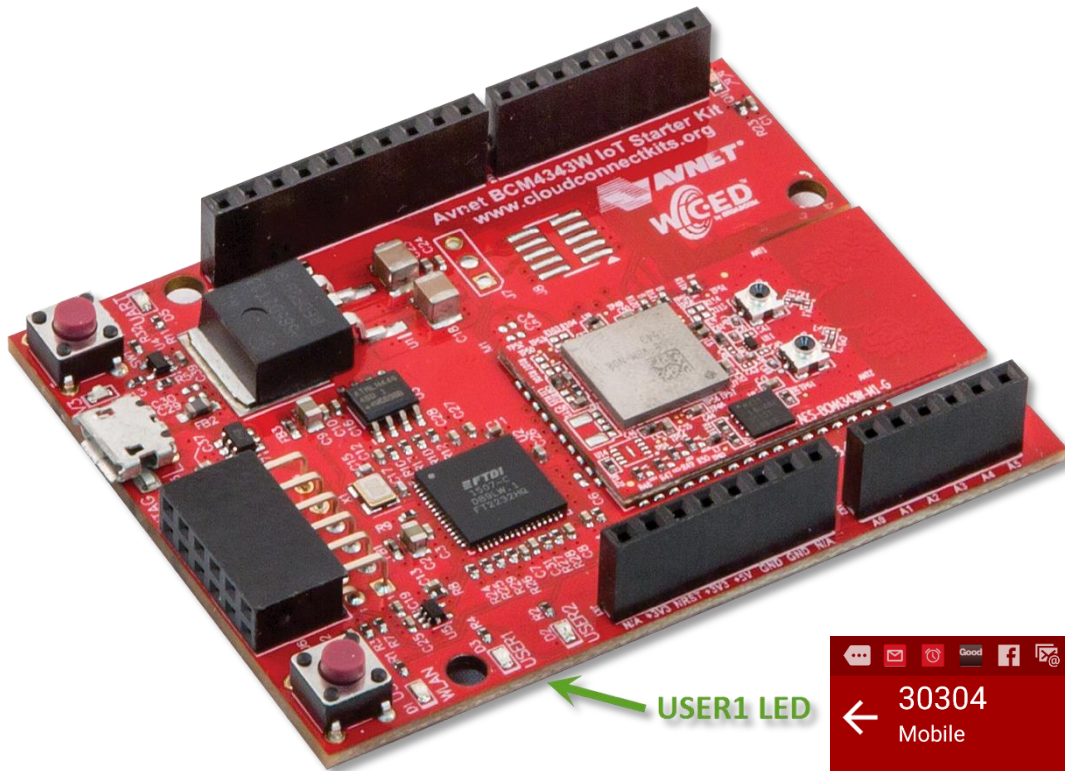


4.3.8 AWS IoT: Testing SMS Messaging (Rule2)

69. As before, from the “Thing” screen of the AWS IoT Console, select “Update Shadow” tab and change the OFF/ON value of desired status then click Update Shadow



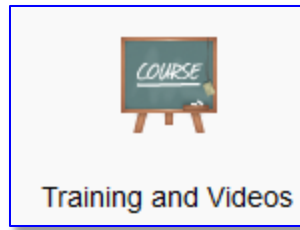
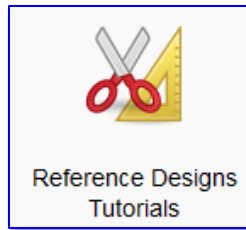
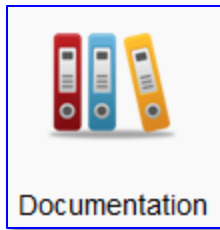
70. The state of USER1 LED on the Starter Kit will change and an SMS text message should immediately arrive on your phone.



5 Getting Help and Support

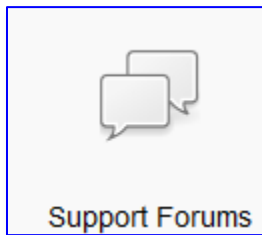
5.1 Avnet Support

For online documentation, training and videos, go to: <http://cloudconnectkits.org>



To access the Avnet technical forum for this Starter Kit, go to:

<http://cloudconnectkits.org/forum>



5.2 Cypress WICED Wi-Fi Forum Support

<https://community.cypress.com/community/wiced-wifi/wiced-wifi-forums>

After registering for this forum you can browse the many postings and solutions as well as submitting your own questions regarding WICED SDK tools as well as the BCM4343W Cypress device on this module

5.3 AWS Forum Support

<https://forums.aws.amazon.com/forum.jspa?forumID=210>

Discussion Forums > Category: Internet of Things > Forum: AWS IoT

This is the AWS Q&A forum specifically focused on the AWS IoT service