

How to Build an IoT Application with Moxa's ThingsPro and AWS IoT Service

Moxa Technical Support Team
support@moxa.com

Contents

1	Introduction.....	2
2	Application Scenario	2
3	Prerequisites.....	3
4	Solution	3
5	How to Get the AWS IoT Parameters for ThingsPro	10
6	Additional Reading	19

Copyright © 2018 Moxa Inc.

Released on February 02, 2018

About Moxa

Moxa is a leading provider of edge connectivity, industrial computing, and network infrastructure solutions for enabling connectivity for the Industrial Internet of Things. With over 30 years of industry experience, Moxa has connected more than 50 million devices worldwide and has a distribution and service network that reaches customers in more than 70 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for industrial communications infrastructures. Information about Moxa’s solutions is available at www.moxa.com.

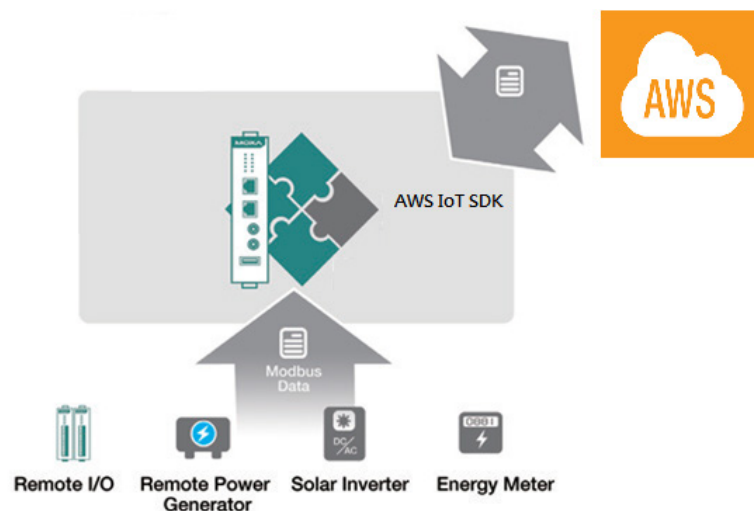
How to Contact Moxa

Tel: 1-714-528-6777
Fax: 1-714-528-6778



1 Introduction

Moxa's ThingsPro™ 2.0 Suite simplifies the development of IoT applications and facilitates data acquisition as well as remote device management. ThingsPro provides Modbus communications, data acquisition, wireless networking, and device management, in a few simple steps, allowing users to focus on developing applications instead of complex system integration.



For users who develop data acquisition and asset management software programs on their own, ThingsPro provides the ability to transfer field data to the AWS IoT service without requiring any additional programming on the gateway. ThingsPro 2.0 includes generic Modbus protocol and AWS IoT support, which you can use to easily configure protocol data polling tables and AWS IoT connection settings. You can then upload the data collected to the AWS IoT service. In this article, we describe how to configure the MODBUS device template in ThingsPro, retrieve data from ThingsPro, and upload the data to a remote HTTP server.

2 Application Scenario

Set up an environment to send real-time device data updates to the AWS IoT service using ThingsPro 2.0. Here, ThingsPro is used as a Modbus data logger and the AWS function provided by ThingsPro is used to configure the connection to the AWS IoT service.

3 Prerequisites

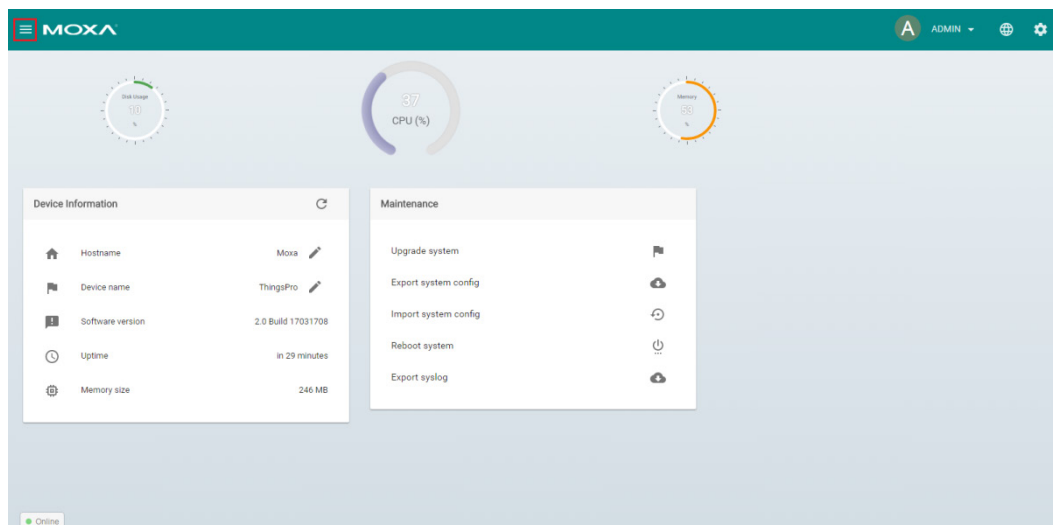
1. Moxa UC-8112 gateway computer with Internet access, and ThingsPro 2.0 Gateway and SD card installed.
For information about setting up a Moxa UC-8112, refer to the *ThingsPro User's Manual*.
2. PC/notebook with Chrome browser and [Microsoft Device Explorer](#) installed
3. Modbus device (e.g., Moxa ioLogik-E2242)
4. [AWS IoT configuration: thing name, CA certificate, device certificate, device private key, mqtt topics, and mqtt endpoint](#)

4 Solution

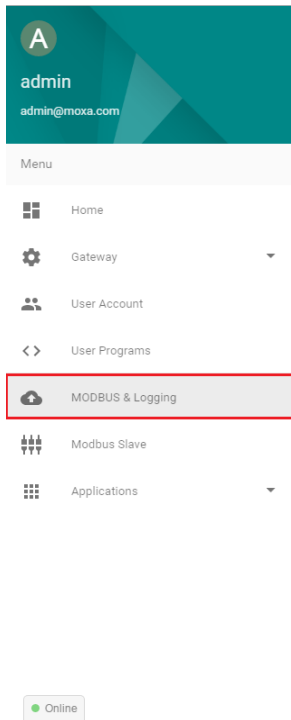
Complete the following steps to create your IoT application using ThingsPro and AWS IoT service:

1. Choose an equipment template.

- a. In the ThingsPro Gateway web page, click  (Menu).

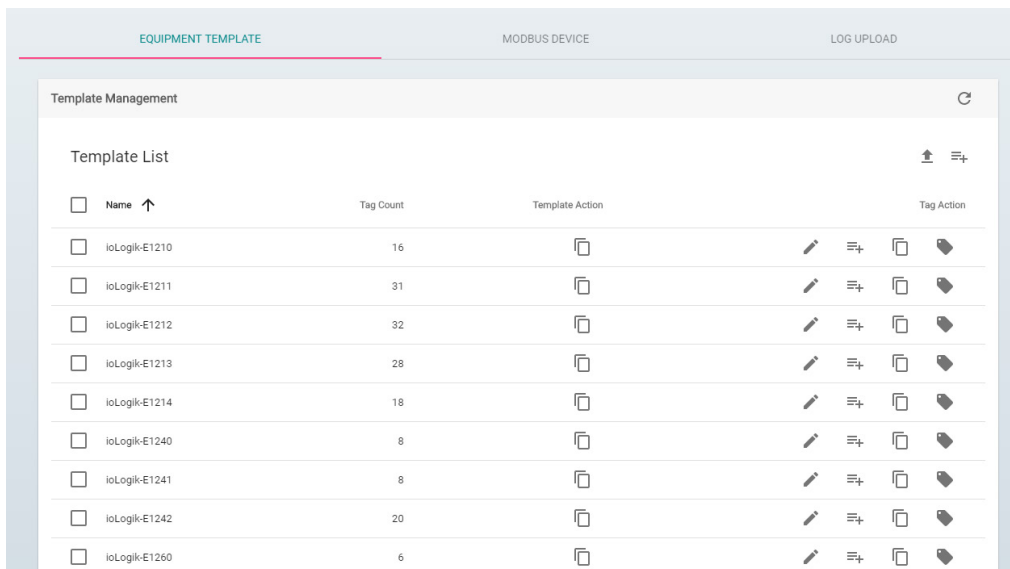


- b. In the ThingsPro Gateway menu, click on the **Modbus & Logging** link.



- c. Open the **EQUIPMENT TEMPLATE** tab
- d. Browse through the **Template List** to locate the template corresponding to the Modbus device or create a new template.

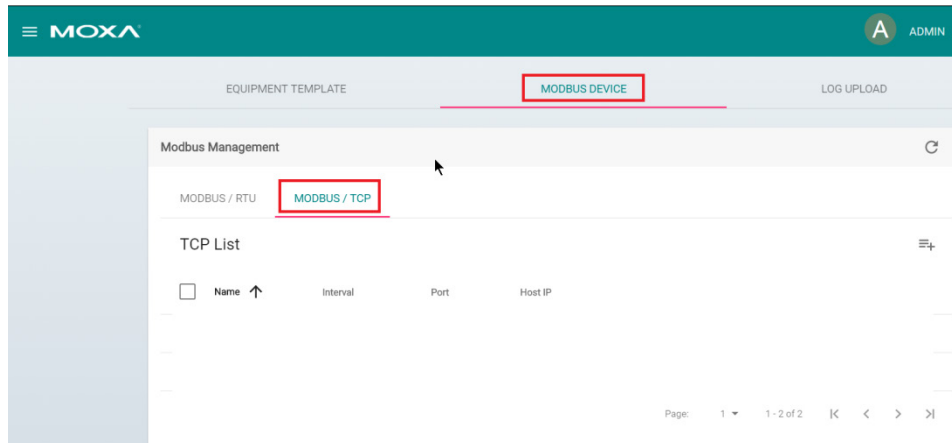
For details on creating a new template, refer to the *ThingsPro User's Manual*.




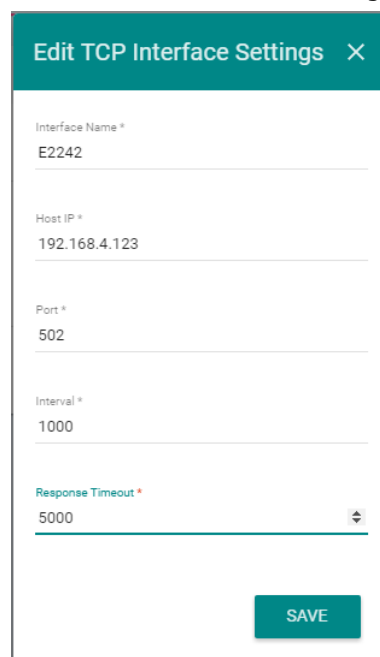
In this article, we use the Modbus device **E2242**, which is already listed in the **Template List**.

2. Create an instance of the Modbus device.

- a. Select the **MODBUS DEVICE** tab.
- b. Configure the Modbus device with a communication interface (serial or Ethernet).
Because the connected device (E2242) uses Modbus TCP protocol, click on the **MODBUS/TCP** tab.



- i. Click  (Add TCP interface).
- ii. Fill in the TCP interface settings and click **SAVE**.



Edit TCP Interface Settings ✕

Interface Name *
E2242

Host IP *
192.168.4.123


Port *
502

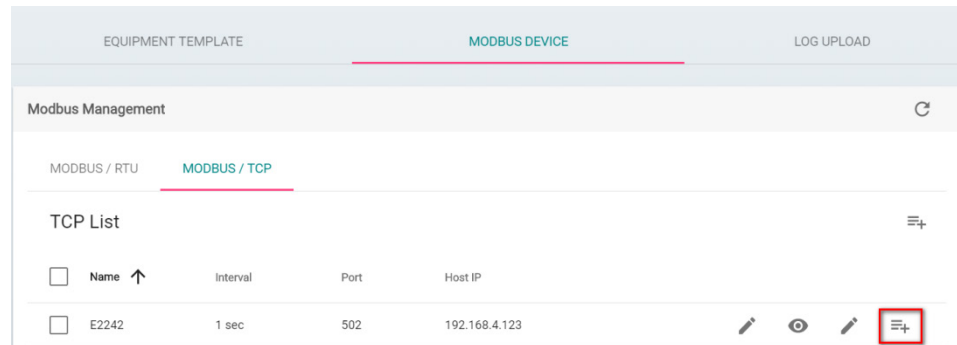
Interval *
1000

Response Timeout *
5000

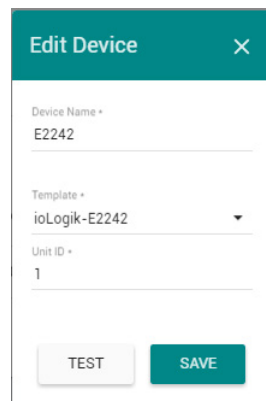
SAVE

The Modbus/TCP interface for E2242 is added to the **TCP List**.

- iii. Click on the  (Add connected device) button corresponding to E2242 to add a connected device.



- iv. Fill in the **Device Name**.
- v. Select a **Template** to combine MODBUS/TCP interface with the Modbus device.
- vi. Specify the **Unit ID** for the device and click **SAVE**.



Edit Device [X]

Device Name *
E2242

Template *
ioLogik-E2242

Unit ID *
1

TEST SAVE

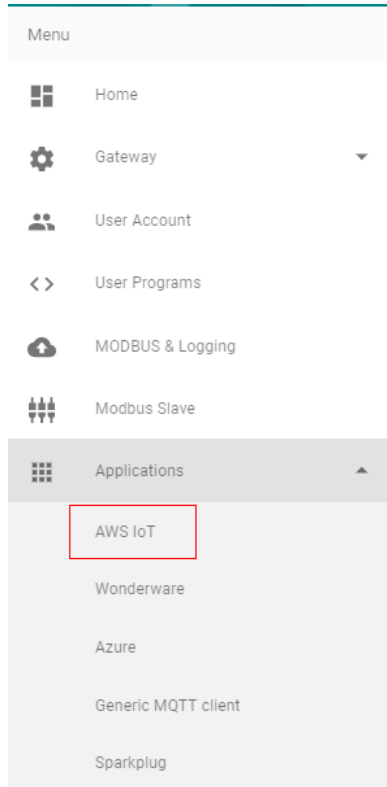
ThingsPro will now start continuously polling data from the device.

To check the status, click  (Show connected devices).

If the **Request Status** is green, ThingsPro is able to send requests to the Modbus TCP device and receive data.

3. Use the AWS IoT service for real-time data update.

- a. In the ThingsPro Gateway menu, click on the **AWS IoT** link in the **Applications** section.



- b. Fill in the AWS IoT service details.
Follow the steps in "How to get AWS IoT parameters for ThingsPro" to get the AWS IoT service parameters.

AWS IoT

Enable

Connection Status

Target Host *
This field is required

Port *
8883

Topic *
This field is required

Client ID *
This field is required

My Thing Name *
This field is required

Root CA File E.g.: ****-G5.pem SELECT

Certificate File E.g.: ****-certificate.pem.crt SELECT

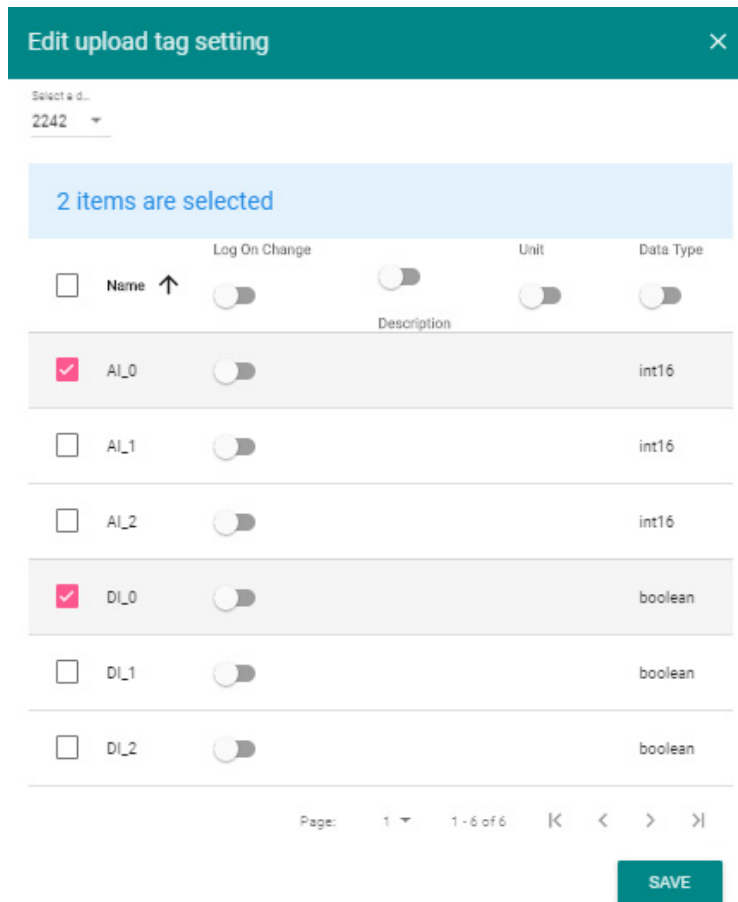
Private Key File E.g.: ****-private.pem.key SELECT

SELECT TAGS

Logging data when network is disconnected

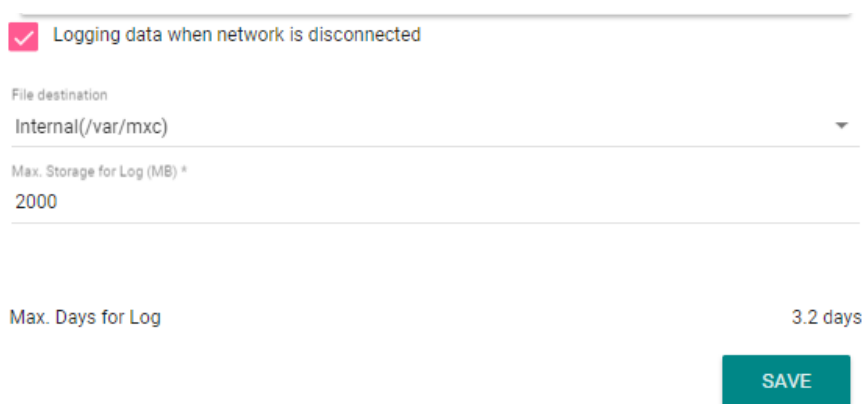
SAVE

- c. Click **SELECT TAGS** to select the tags that you want to upload to AWS IoT.
- d. Click **SAVE**.



- e. If you need to log data when network is down, check the **Logging data when network is disconnected**, select the log **File destination**, and specify the **Max. Storage for Log value**.

Your data will be logged in a data file. Once the network connection comes up, the data in the log file will be transmitted to AWS.



- f. Select the **Enable** option
- g. Click **SAVE**

The AWS IoT interface will use the information provided to create a connection with the AWS IoT service. Once the connection is established, data can be uploaded to the AWS IoT service in real time.

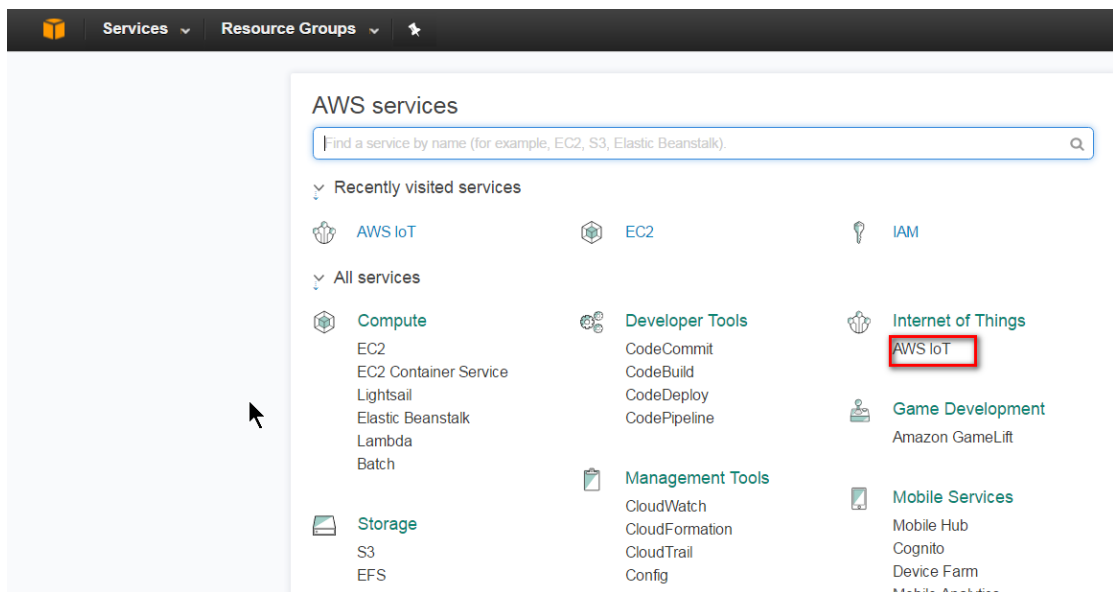
Once the AWS Client App successfully connects to the AWS service, the connection status turns green as shown below:



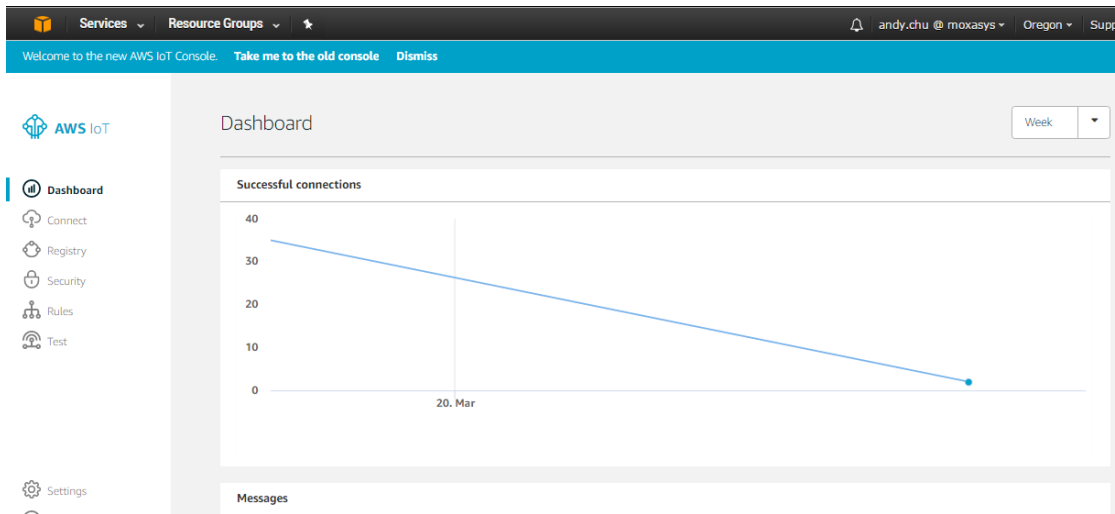
5 How to Get the AWS IoT Parameters for ThingsPro

To get the AWS IoT parameter for ThingsPro, you must log in into the AWS IoT service and do the following:

1. Click on the **AWS IoT** link.



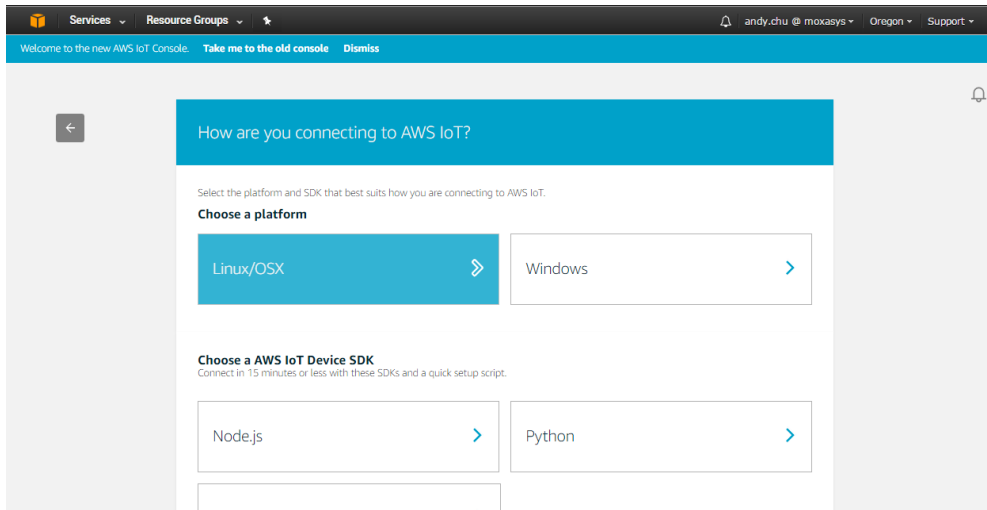
You will see the AWS IoT Dashboard.



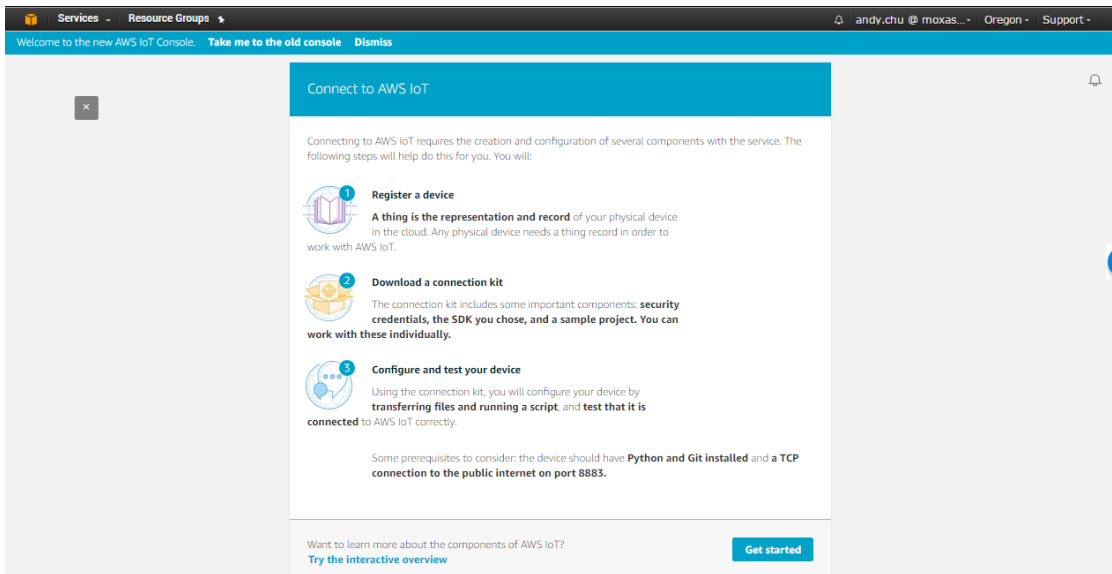
2. Click on the **connect** link and then the **Get started** button.

The screenshot shows the 'Connect to AWS IoT' page in the AWS IoT console. The left navigation menu is updated to include 'Connect' and 'Learn'. The main content area has three cards: 1) 'Configuring a device' with a 'Get started' button and a description: 'Connect a device or your computer to AWS IoT using the connection wizard for AWS IoT Device SDKs.' 2) 'AWS IoT Button' with a 'Configure a button' button and a description: 'The AWS IoT Button is a single-purpose device that sends a message to AWS IoT with a press of a button.' Below this is a link: 'Don't have a button? Buy one'. 3) 'AWS IoT Starter Kit' with a 'Browse starter kits' button and a description: 'Browse AWS IoT Starter Kits that were made for connecting to AWS IoT and getting started with the service.' The top navigation bar is consistent with the previous screenshot.

- 3. Choose the **Linux/OSX** platform and select the **Python** SDK.



- 4. Click **Get started**



5. Fill in the thing name. e.g.:8112.
This name will be used in the **Client ID** and **My Thing name** in the ThingsPro AWS IoT connection UI.
6. Click **Next step**.

CONNECT TO AWS IOT

Register a thing

STEP 1/3

A thing is the representation and record of your physical device in the cloud. Any physical device needs a thing to work with AWS IoT. Creating a thing will also create a thing shadow. [Choose an existing thing instead?](#)

Name

Show optional configuration (this can be done later) ▾

Cancel Back Next step

7. Click on the Download connection kit link
8. After the download has finished, click **Next step**.

CONNECT TO AWS IOT
Download a connection kit
STEP 2/3

The following AWS IoT resources will be created:

A thing in the AWS IoT registry	8112	
A policy to send and receive messages	8112-Policy	Preview policy

The connection kit contains:

A certificate and private key	8112.cert.pem, 8112.private.key
AWS IoT Device SDK	Python SDK
A script to send and receive messages	start.sh

Before your device can connect and publish messages, you will need to download the connection kit.

Download connection kit for

[Linux/OSX](#)

[Cancel](#) [Back](#) [Next step](#)

9. Click **Done**.

Note: You do not need to perform steps 1 to 3 under "Configure and test your device".

CONNECT TO AWS IOT
Configure and test your device
STEP 3/3

To configure and test the device, perform the following steps.

Step 1: Unzip the connection kit on the device

```
unzip connect_device_package.zip
```

Step 2: Add execution permissions

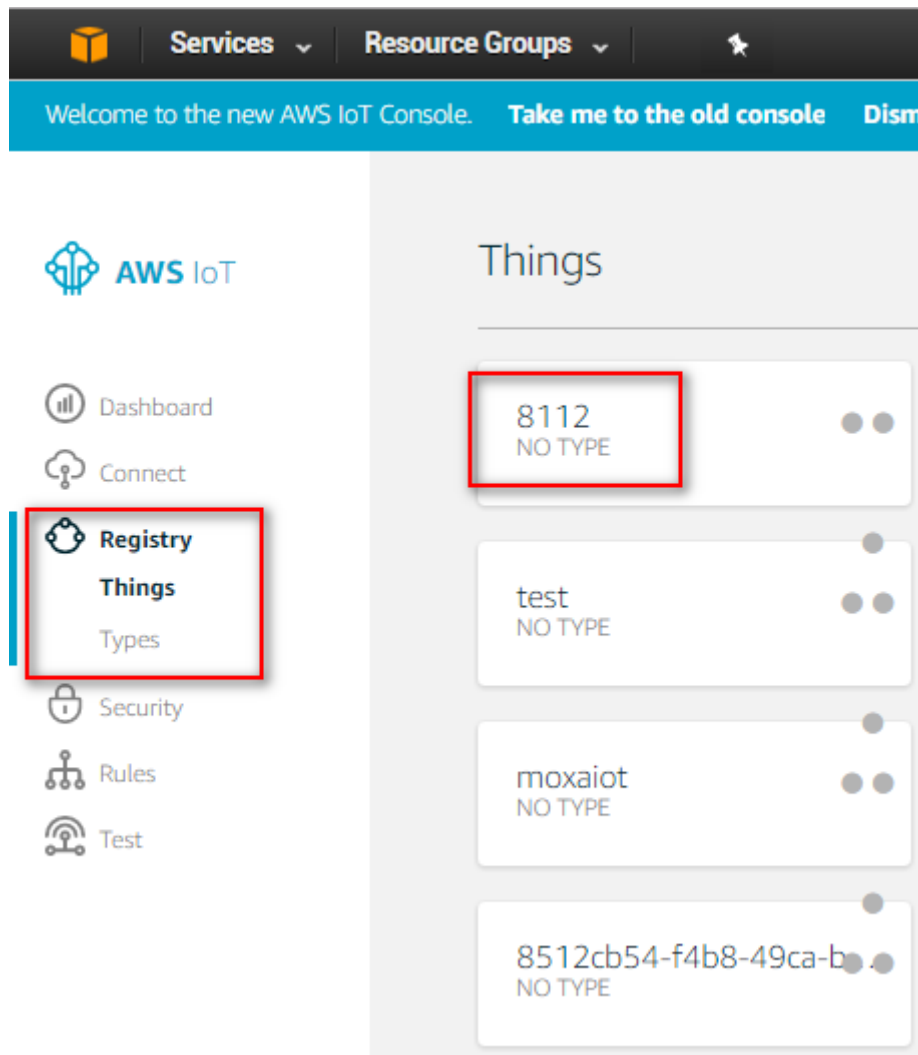
```
chmod +x start.sh
```

Step 3: Run the start script. Messages from your thing will appear below

```
./start.sh
```

```
Waiting for messages from your device
```

[Cancel](#) [Back](#) [Done](#)



14. Click on the thing that you created.
15. To fill in the AWS IoT parameters, do the following:
 - Click on the **Interact** link

- The **update to this thing shadow** is the **Topic** of ThingsPro AWS IoT connection UI.

The screenshot shows the AWS IoT console for a Thing named 8112. The sidebar on the left has 'Interact' selected. The main content area shows the following sections:

- Details:** This thing already appears to be connected. [Connect a device](#)
- HTTPS:** Update your Thing Shadow using this Rest API Endpoint. [Learn more](#)
adw9mfri5w1h2.iot.us-west-2.amazonaws.com
- MQTT:** Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow) [Learn more](#)
 - Update to this thing shadow:** `Saws/things/8112/shadow/update` (highlighted with a red box)
 - Update to this thing shadow was accepted:** `Saws/things/8112/shadow/update/accepted`
 - Update this thing shadow documents:** `Saws/things/8112/shadow/update/documents`

With above information, you should be able to fill up all required information for connecting to AWS IoT service.

AWS IoT ↻

Enable

Update on change

Target host *
xxxxxxx.iot.us-west-2.amazonaws.com

Port *
8883

Topic *
Saws/things/test/shadow/update

Client ID *
test

My Thing Name *
test

Root CA File ex: ****-G5.pem
root-CA.crt SELECT

Certificate File ex: ****-certificate.pem.crt
certificate.pem.crt SELECT

Private Key File ex: ****-private.pem.key
private.pem.key SELECT

SAVE

6 Additional Reading

[Getting Started with AWS IoT](#)