AWS re:Invent

Builders Fair

re:Inventing Hot Desks

and flexible work-spaces

Ian Fayers

Technical Account Manager

AWS

Richard Westby-Nunn

Technical Account Manager

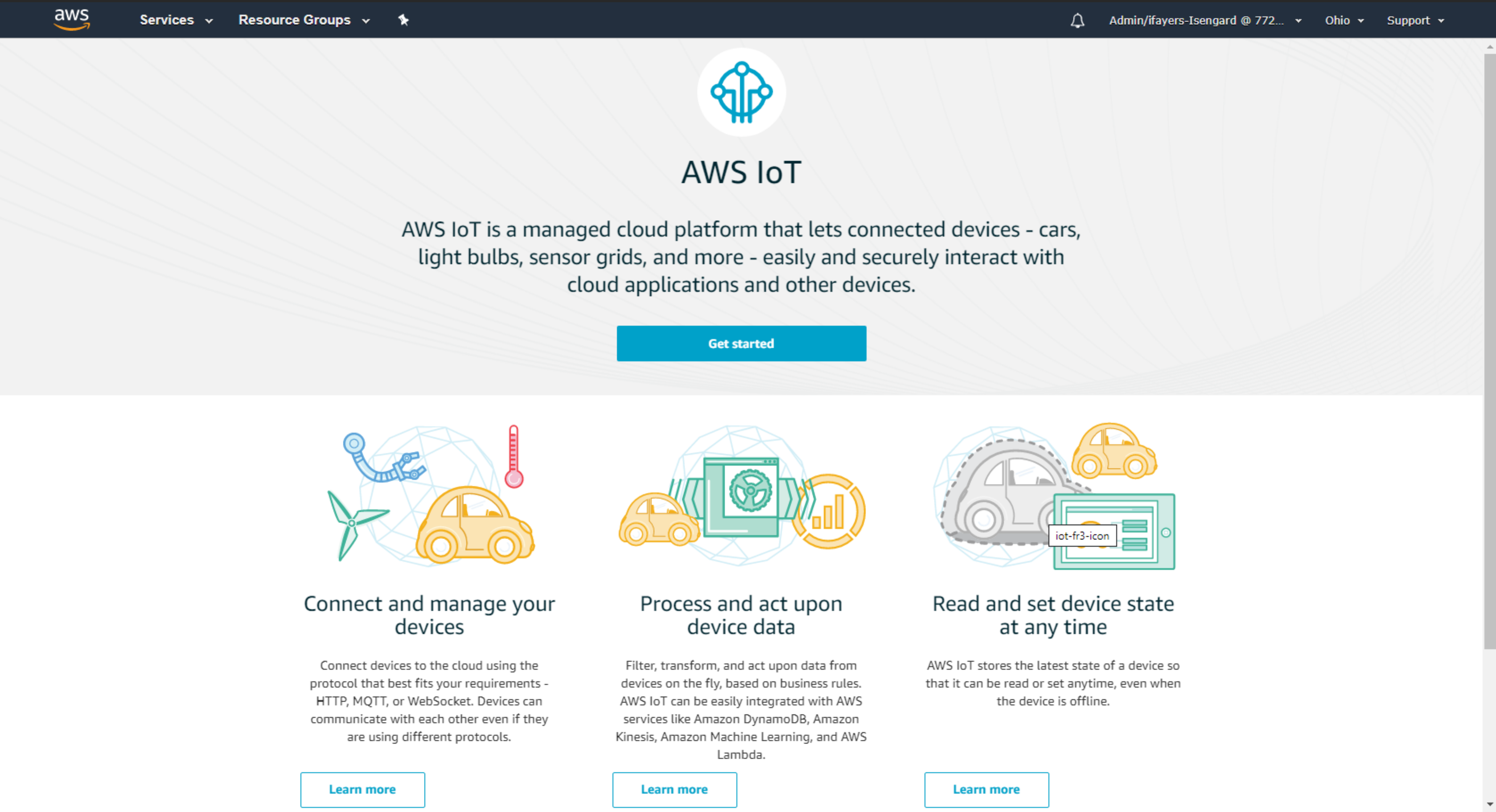
AWS

With extra special thanks to Thiha Soe (GUI) and Alex Fayers (Algorithm implementation)

Overview

To create the IoT backend, follow the steps detailed in this pack.

The front-end (GUI) code may follow later.



AWS IoT

AWS IoT is a managed cloud platform that lets connected devices - cars, light bulbs, sensor grids, and more - easily and securely interact with cloud applications and other devices.

Connect and manage your devices

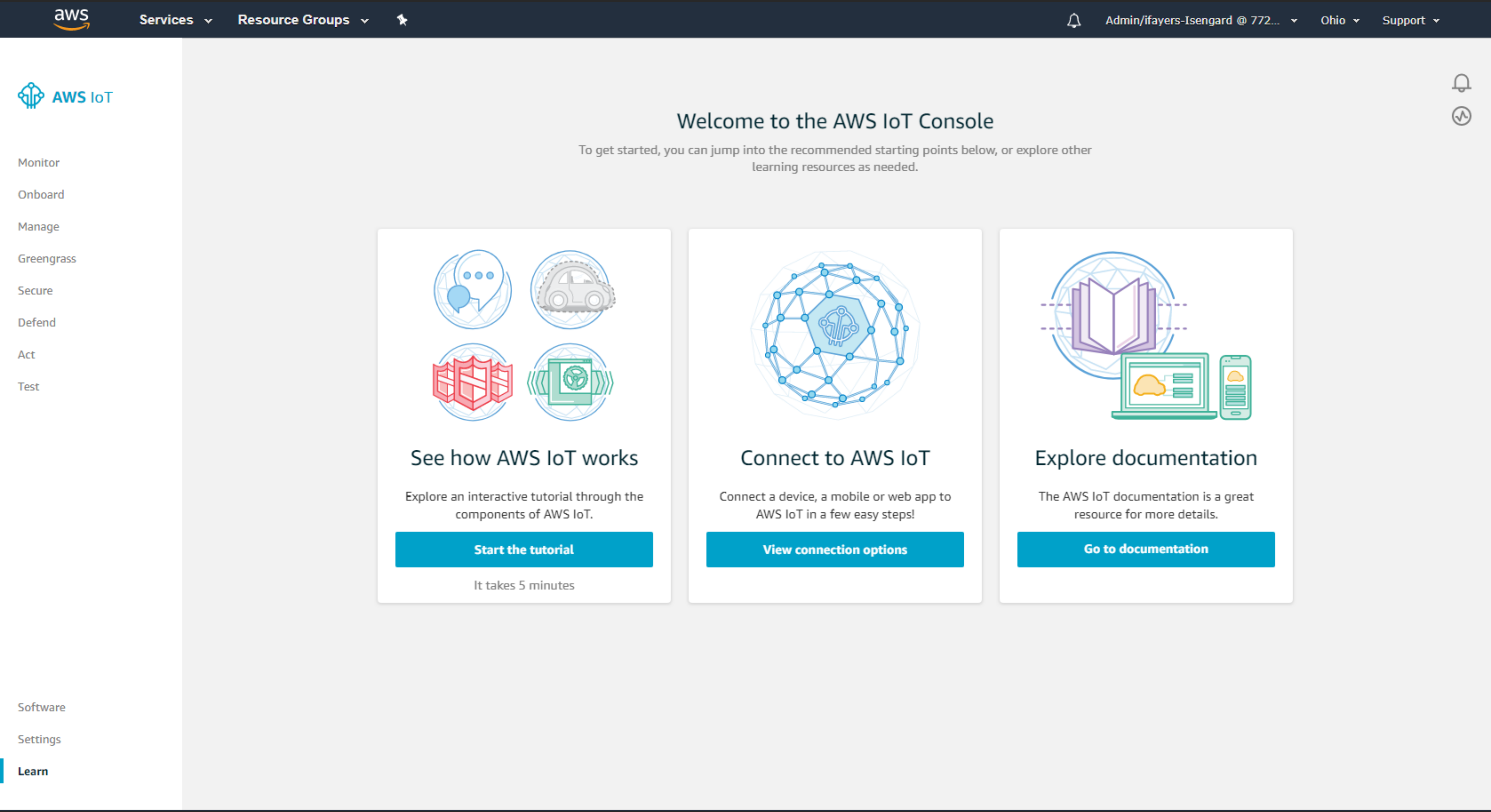
Connect devices to the cloud using the protocol that best fits your requirements - HTTP, MQTT, or WebSocket. Devices can communicate with each other even if they are using different protocols.

Process and act upon device data

Filter, transform, and act upon data from devices on the fly, based on business rules. AWS IoT can be easily integrated with AWS services like Amazon DynamoDB, Amazon Kinesis, Amazon Machine Learning, and AWS Lambda.

Read and set device state at any time

AWS IoT stores the latest state of a device so that it can be read or set anytime, even when the device is offline.



Welcome to the AWS IoT Console

To get started, you can jump into the recommended starting points below, or explore other learning resources as needed.

See how AWS IoT works

Explore an interactive tutorial through the components of AWS IoT.

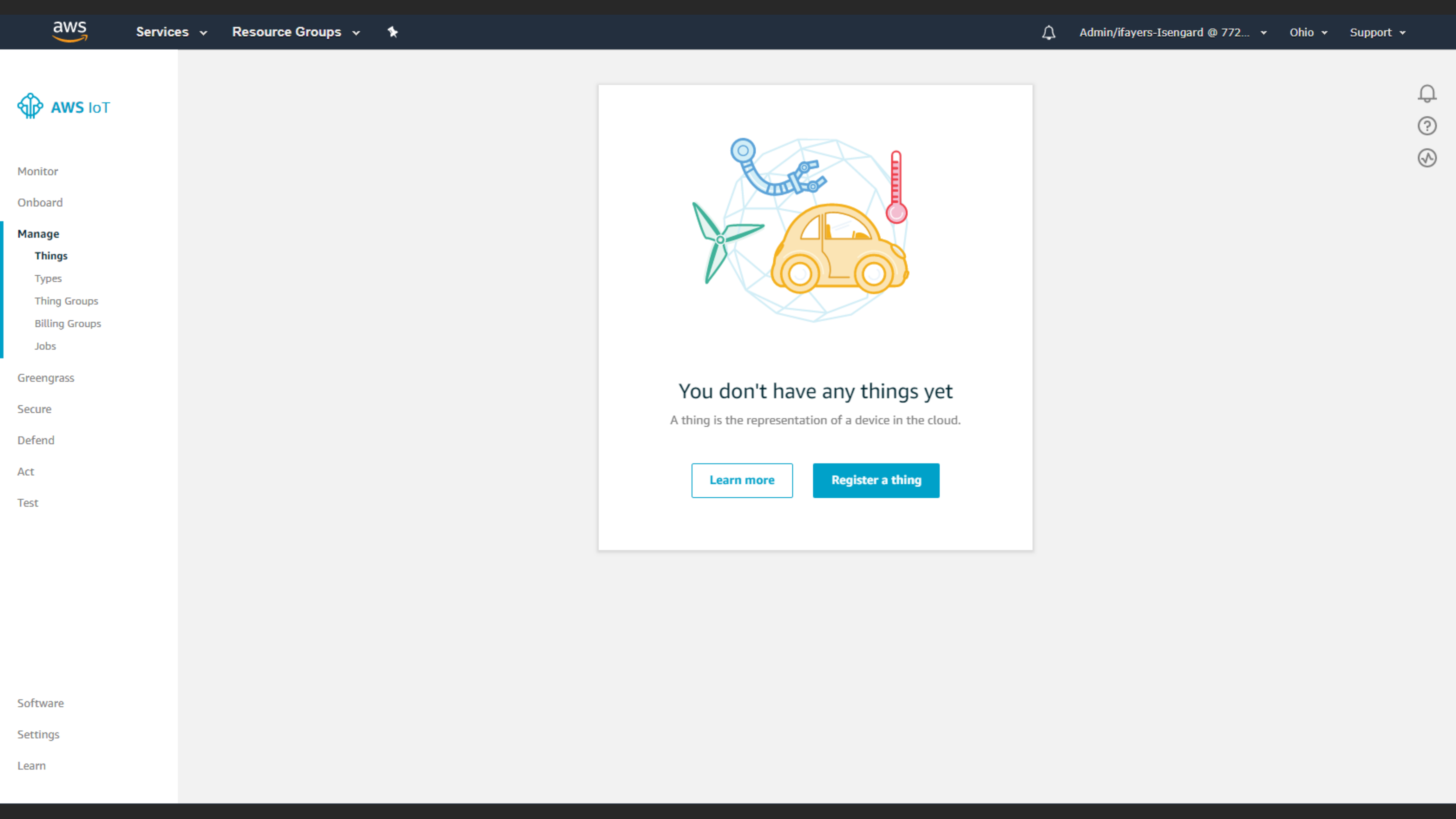
It takes 5 minutes

Connect to AWS IoT

Connect a device, a mobile or web app to AWS IoT in a few easy steps!

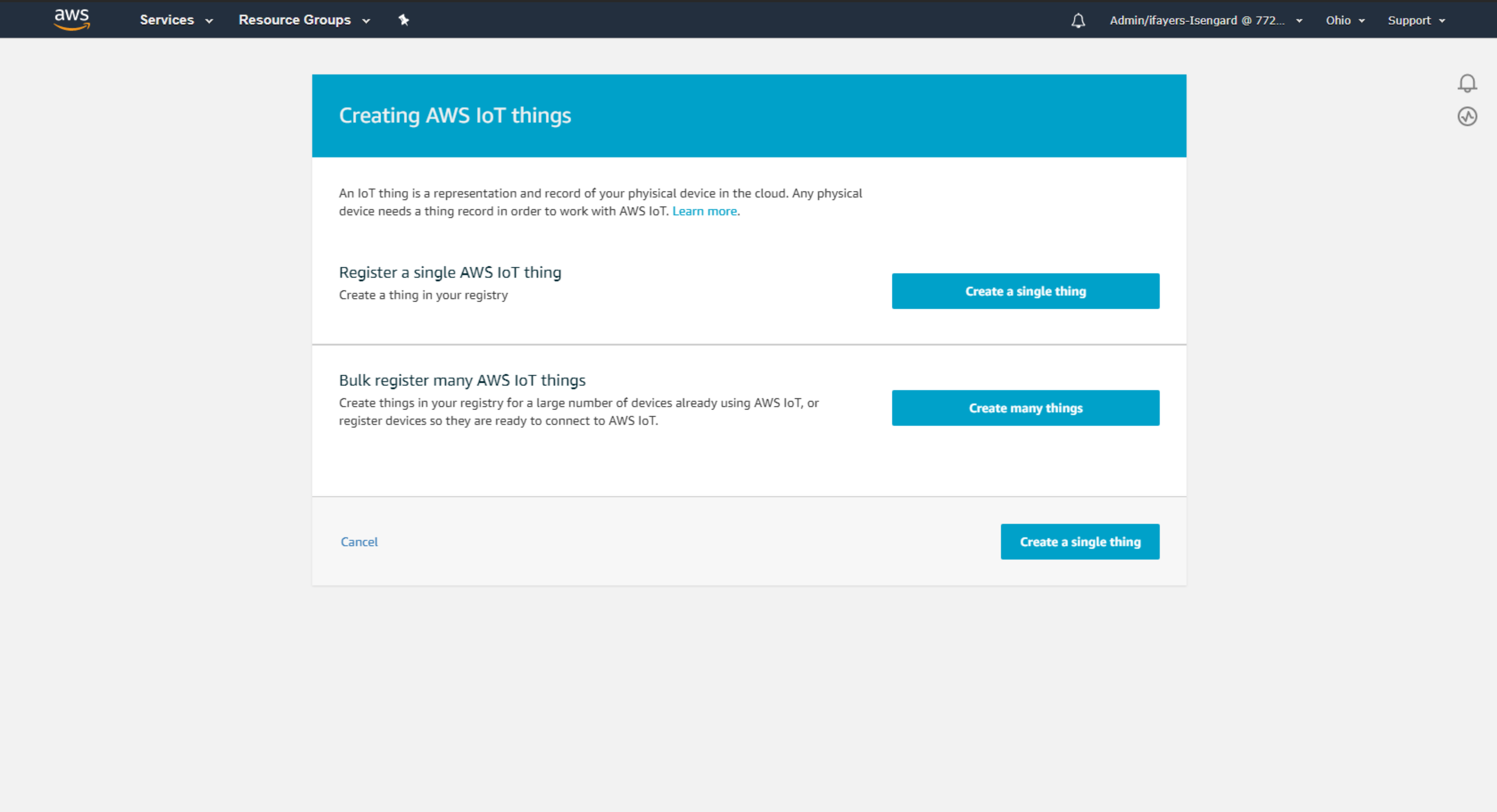
Explore documentation

The AWS IoT documentation is a great resource for more details.



You don't have any things yet

A thing is the representation of a device in the cloud.



Creating AWS IoT things

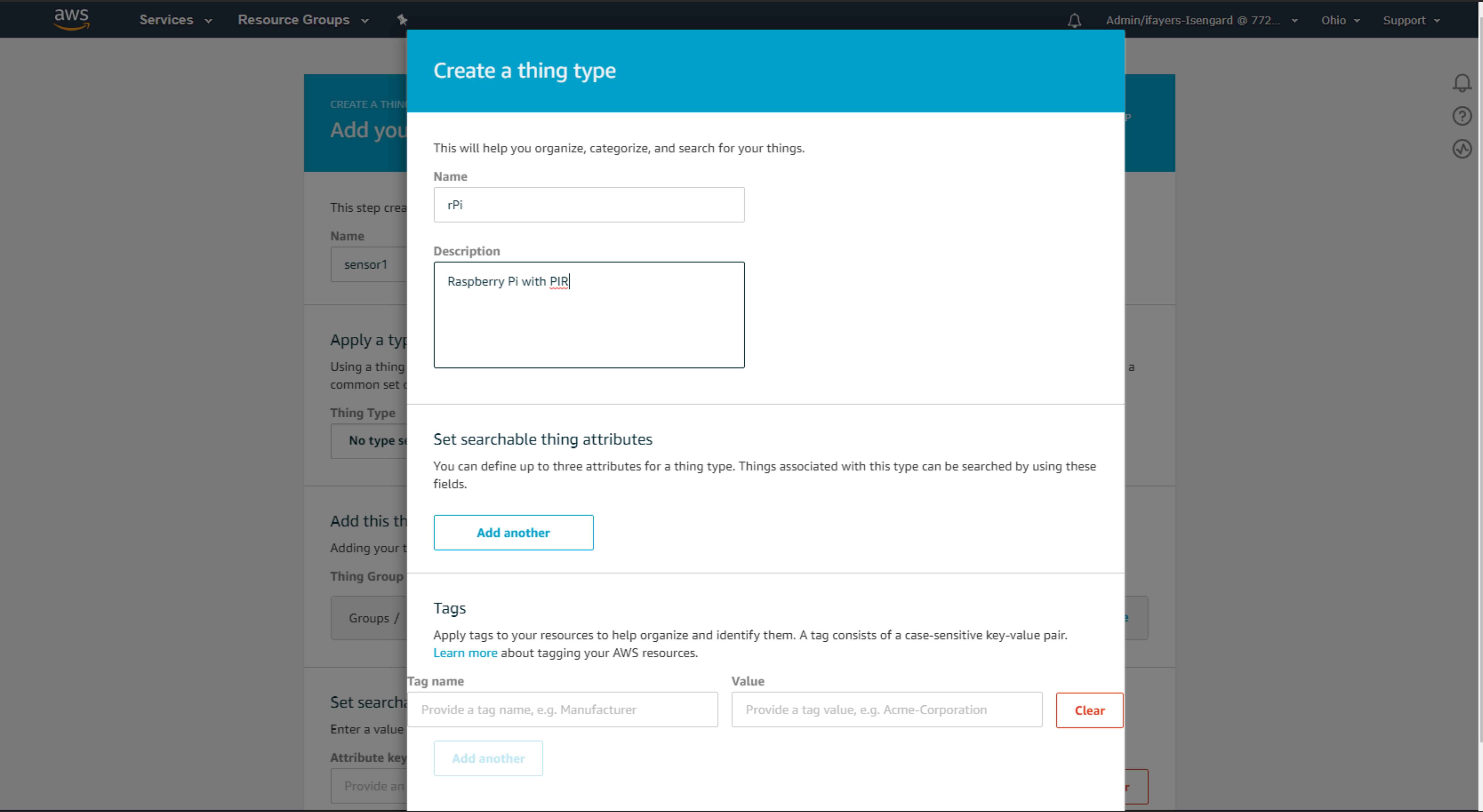
An IoT thing is a representation and record of your phyisical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. Learn more.

Register a single AWS IoT thing

Create a thing in your registry

Bulk register many AWS IoT things

Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.



Create a thing type

This will help you organize, categorize, and search for your things.

Name: rPi

Description: Raspberry Pi with PIR

Set searchable thing attributes

You can define up to three attributes for a thing type. Things associated with this type can be searched by using these fields.

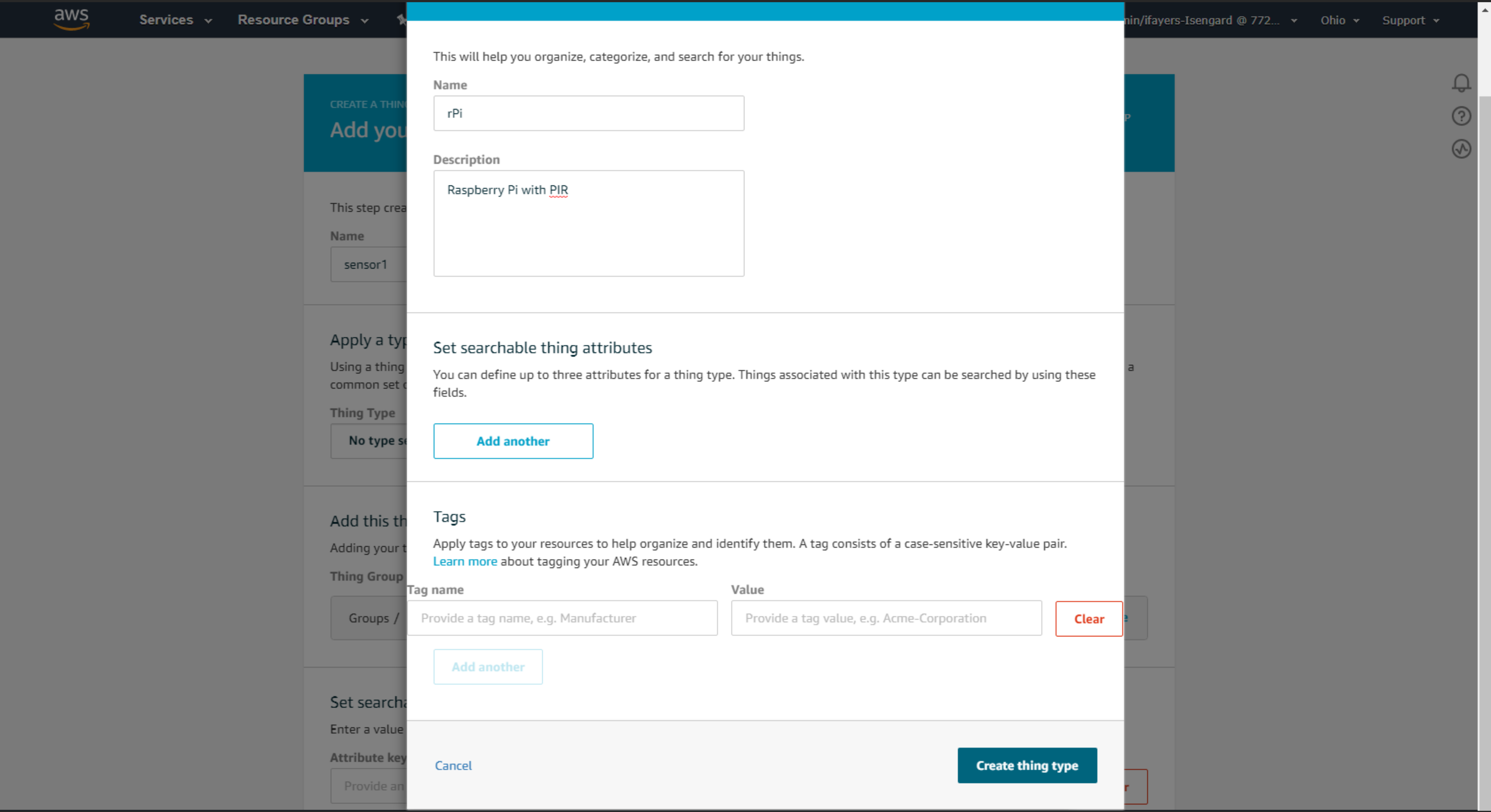
Add another

Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. Learn more about tagging your AWS resources.

Tag name	Value	
Provide a tag name, e.g. Manufacturer	Provide a tag value, e.g. Acme-Corporation	Clear

Add another



This will help you organize, categorize, and search for your things.

Name

rPi

Description

Raspberry Pi with PIR

Set searchable thing attributes

You can define up to three attributes for a thing type. Things associated with this type can be searched by using these fields.

Add another

Tags

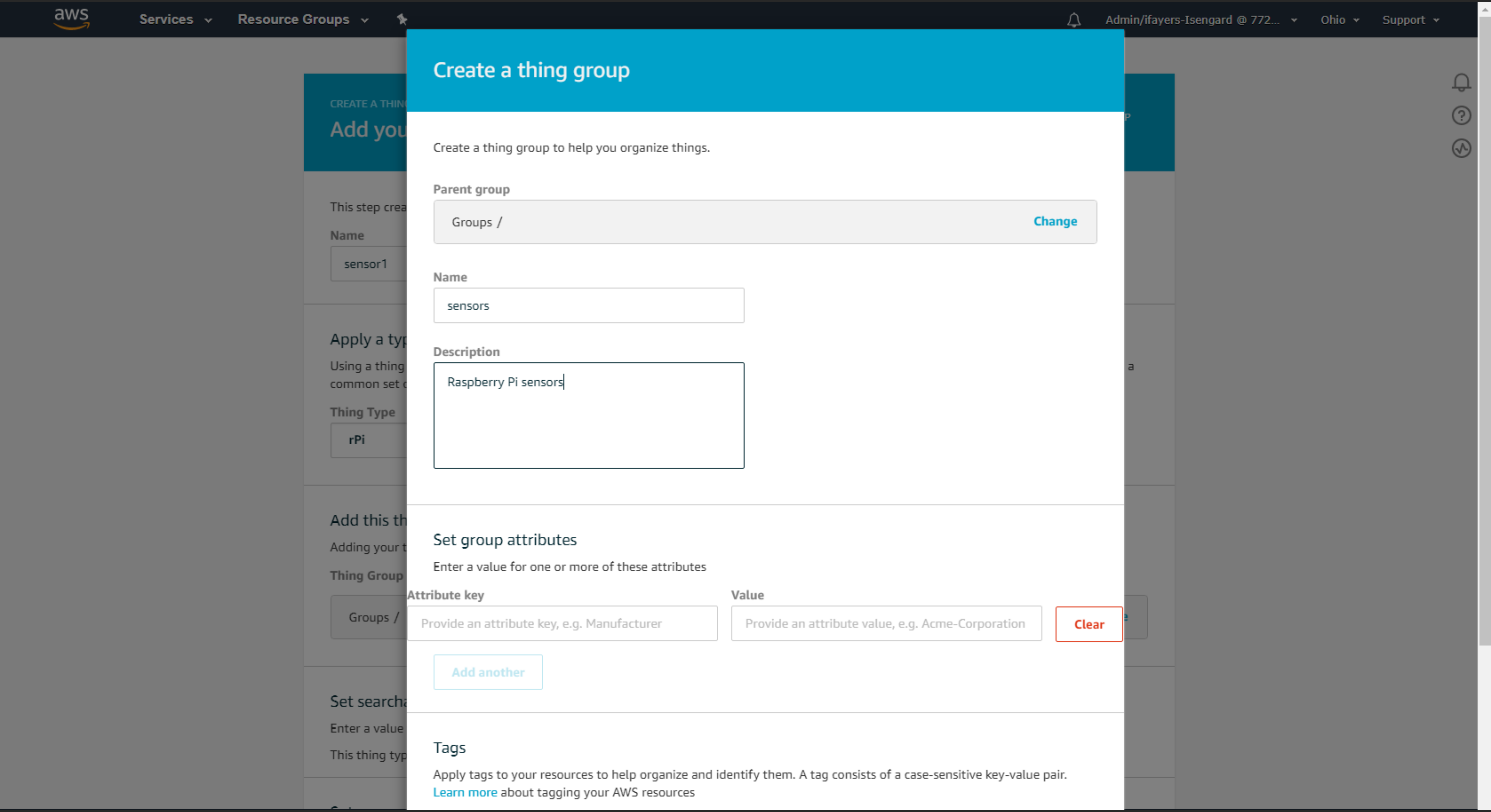
Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. Learn more about tagging your AWS resources.

Tag name	Value	
Provide a tag name, e.g. Manufacturer	Provide a tag value, e.g. Acme-Corporation	Clear

Add another

Cancel

Create thing type



Create a thing group

Create a thing group to help you organize things.

Parent group

Groups / Change

Name

sensors

Description

Raspberry Pi sensors

Set group attributes

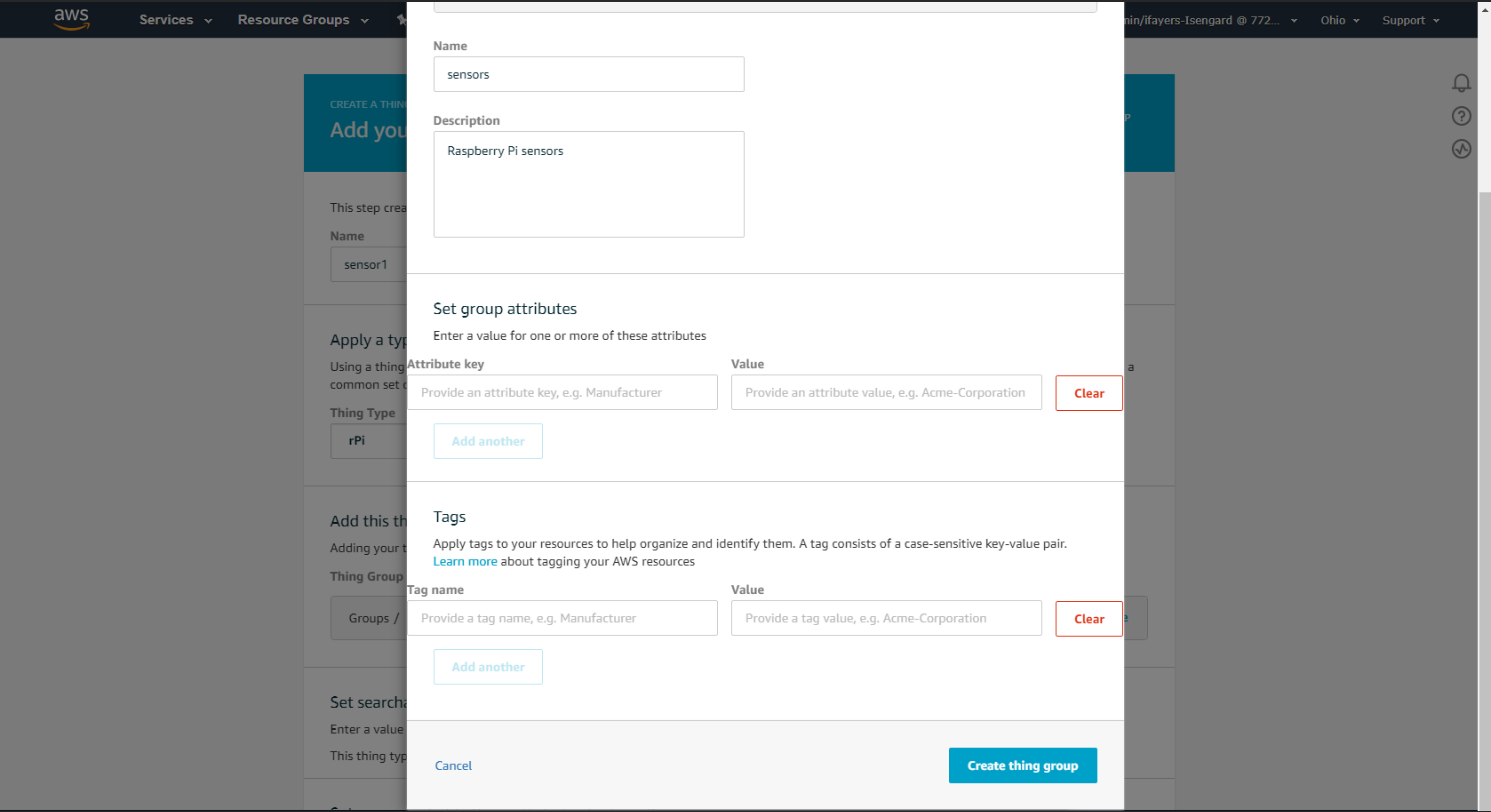
Enter a value for one or more of these attributes

Attribute key	Value	
Provide an attribute key, e.g. Manufacturer	Provide an attribute value, e.g. Acme-Corporation	Clear

Add another

Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. Learn more about tagging your AWS resources



Name

sensors

Description

Raspberry Pi sensors

Set group attributes

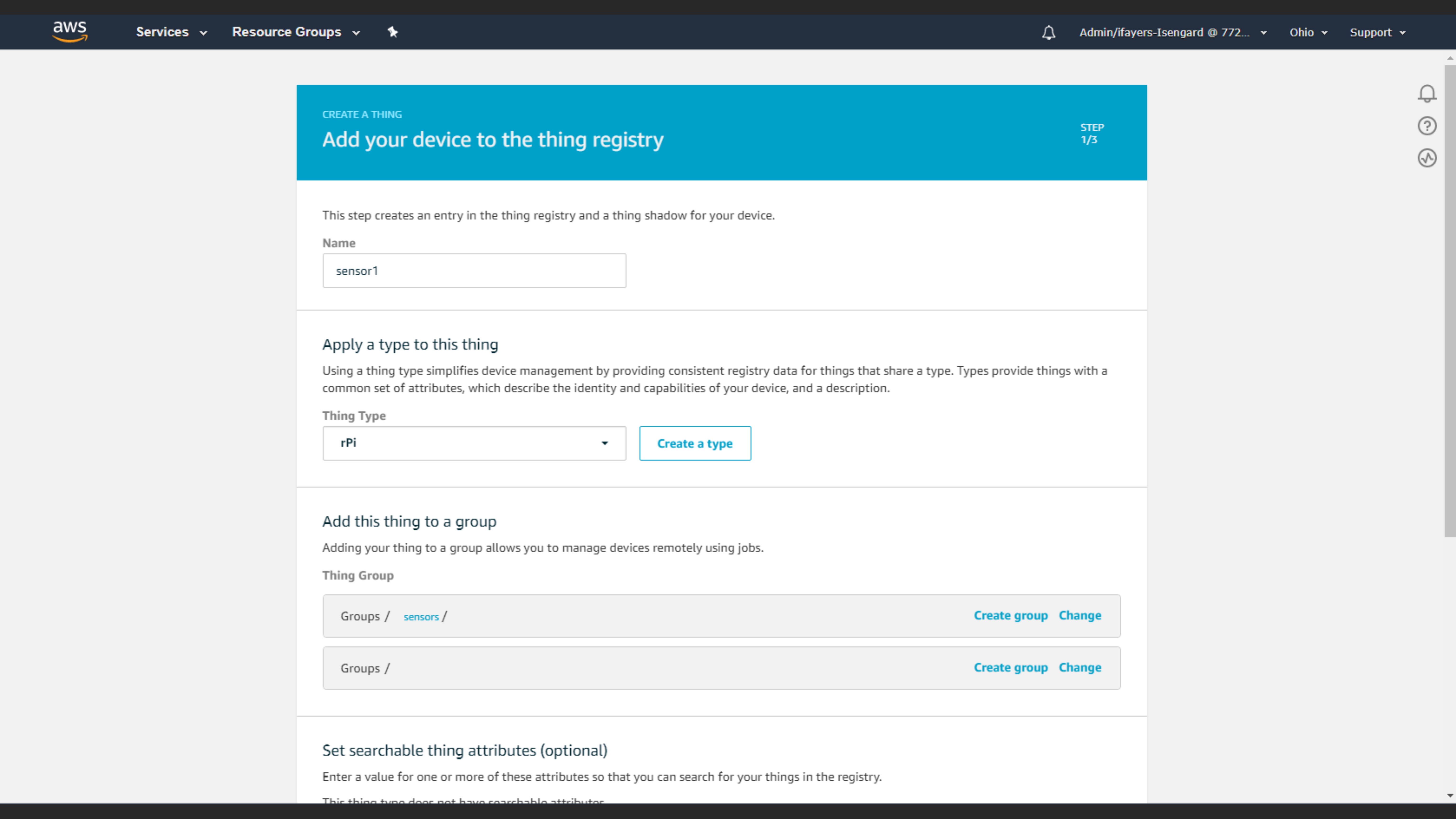
Enter a value for one or more of these attributes

Attribute key	Value
Provide an attribute key, e.g. Manufacturer	Provide an attribute value, e.g. Acme-Corporation

Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. Learn more about tagging your AWS resources

Tag name	Value
Provide a tag name, e.g. Manufacturer	Provide a tag value, e.g. Acme-Corporation



CREATE A THING

Add your device to the thing registry

STEP 1/3

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

Name: sensor1

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: rPi

Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

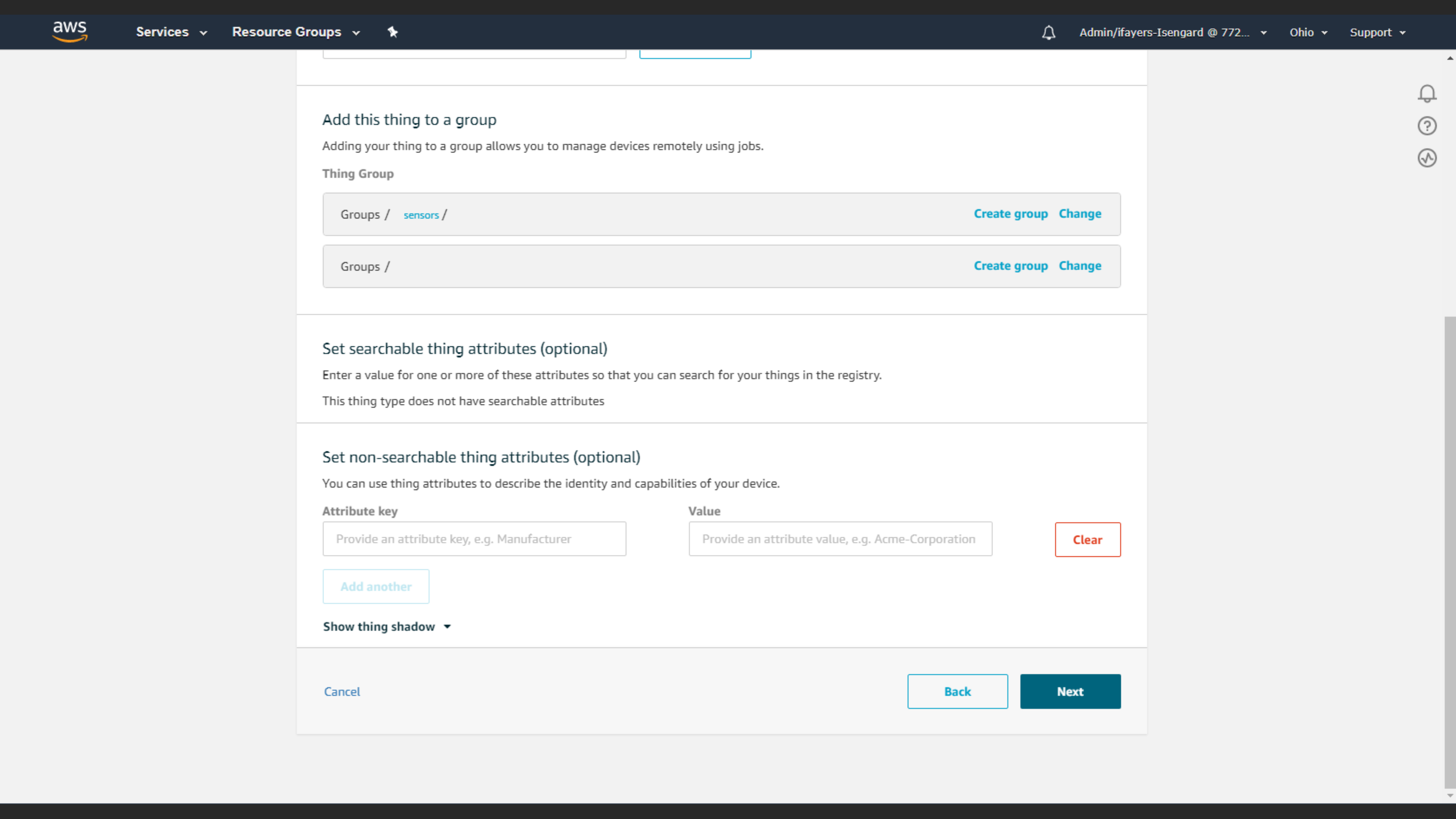
Groups / sensors /

Groups /

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.

This thing type does not have searchable attributes



Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

Groups / sensors /

Groups /

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.

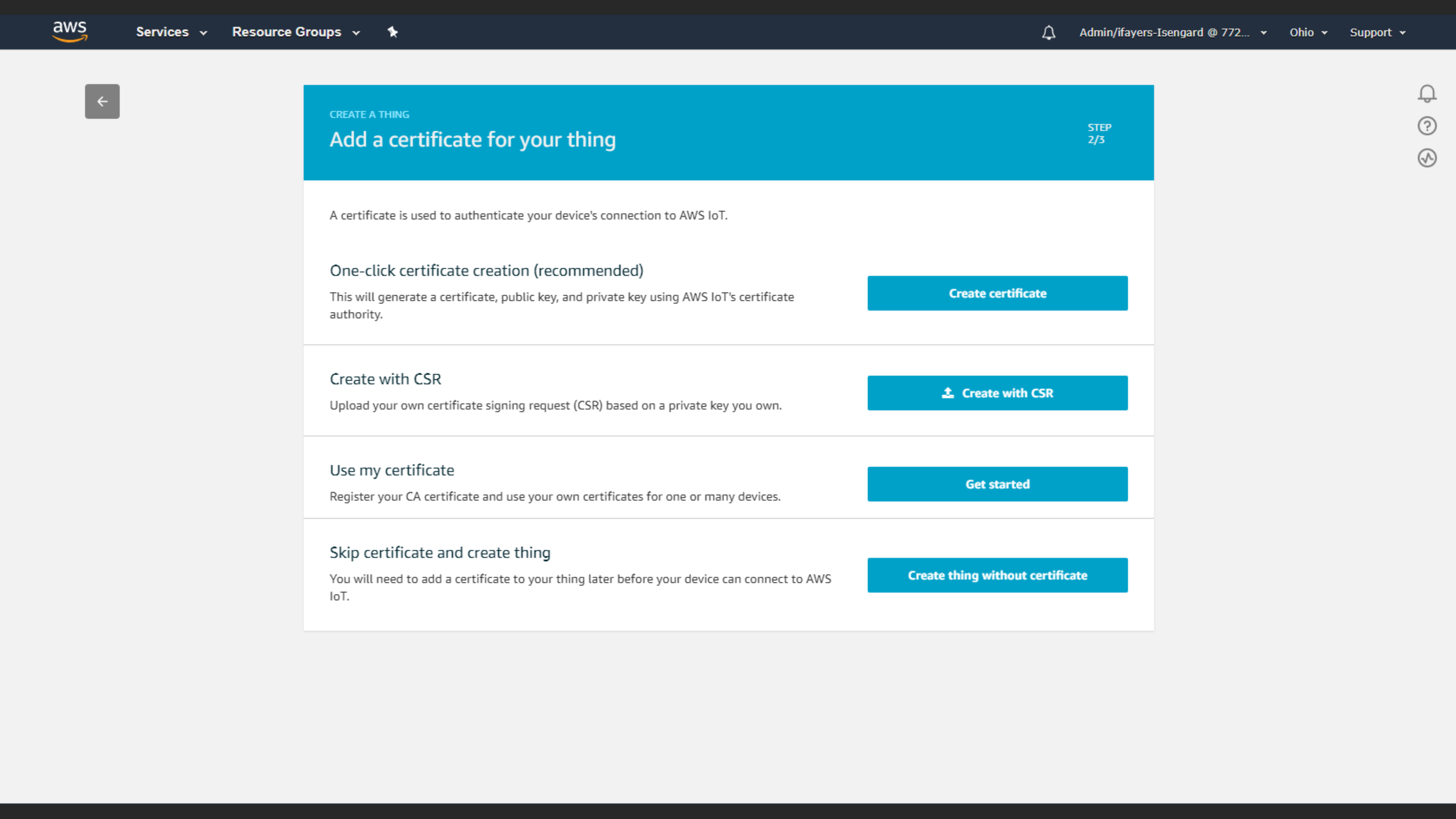
This thing type does not have searchable attributes

Set non-searchable thing attributes (optional)

You can use thing attributes to describe the identity and capabilities of your device.

Attribute key: Provide an an attribute key, e.g. Manufacturer

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



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 with CSR

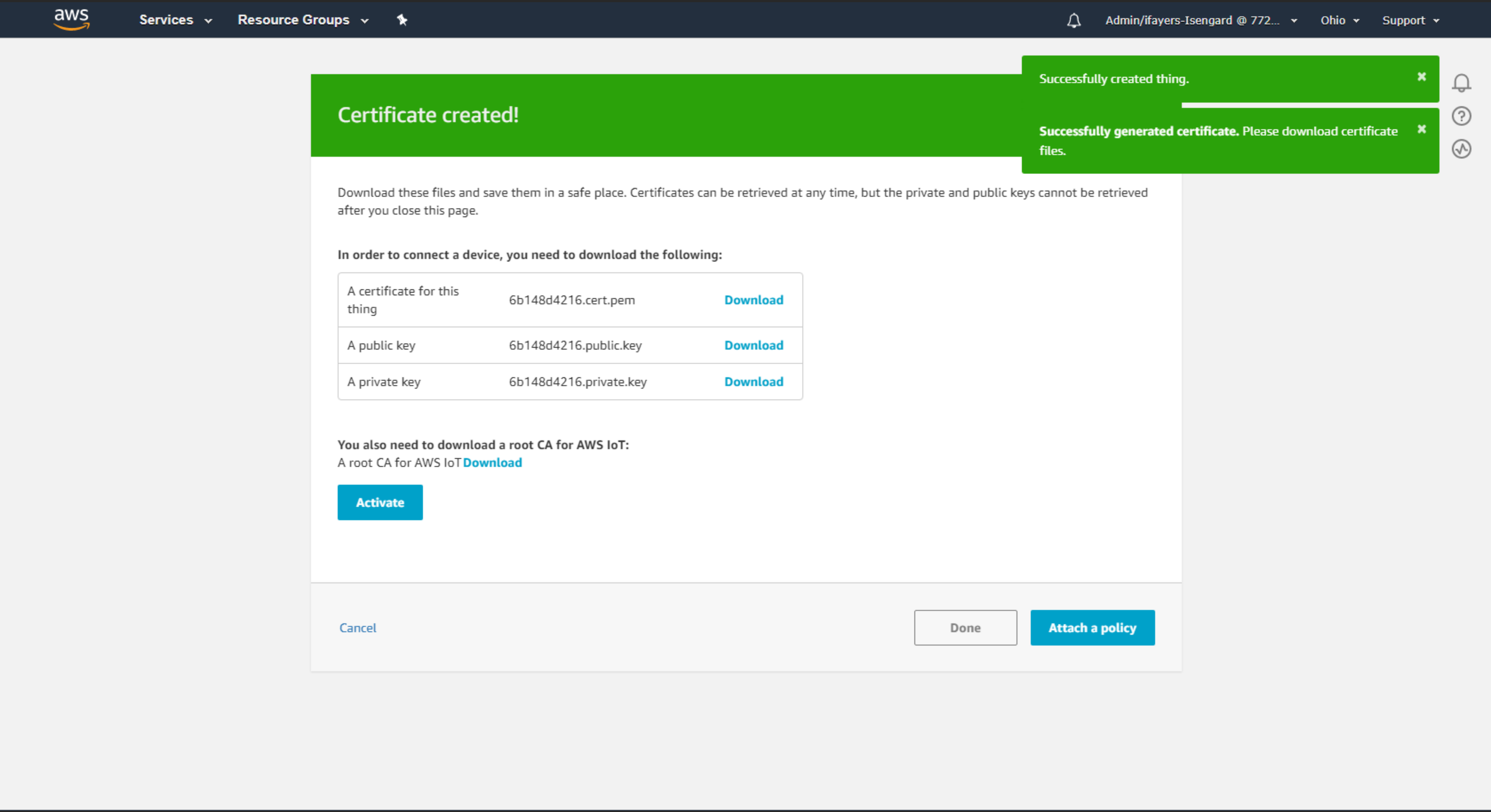
Upload your own certificate signing request (CSR) based on a private key you own.

Use my certificate

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

Skip certificate and create thing

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



Successfully created thing.

Successfully generated certificate. Please download certificate files.

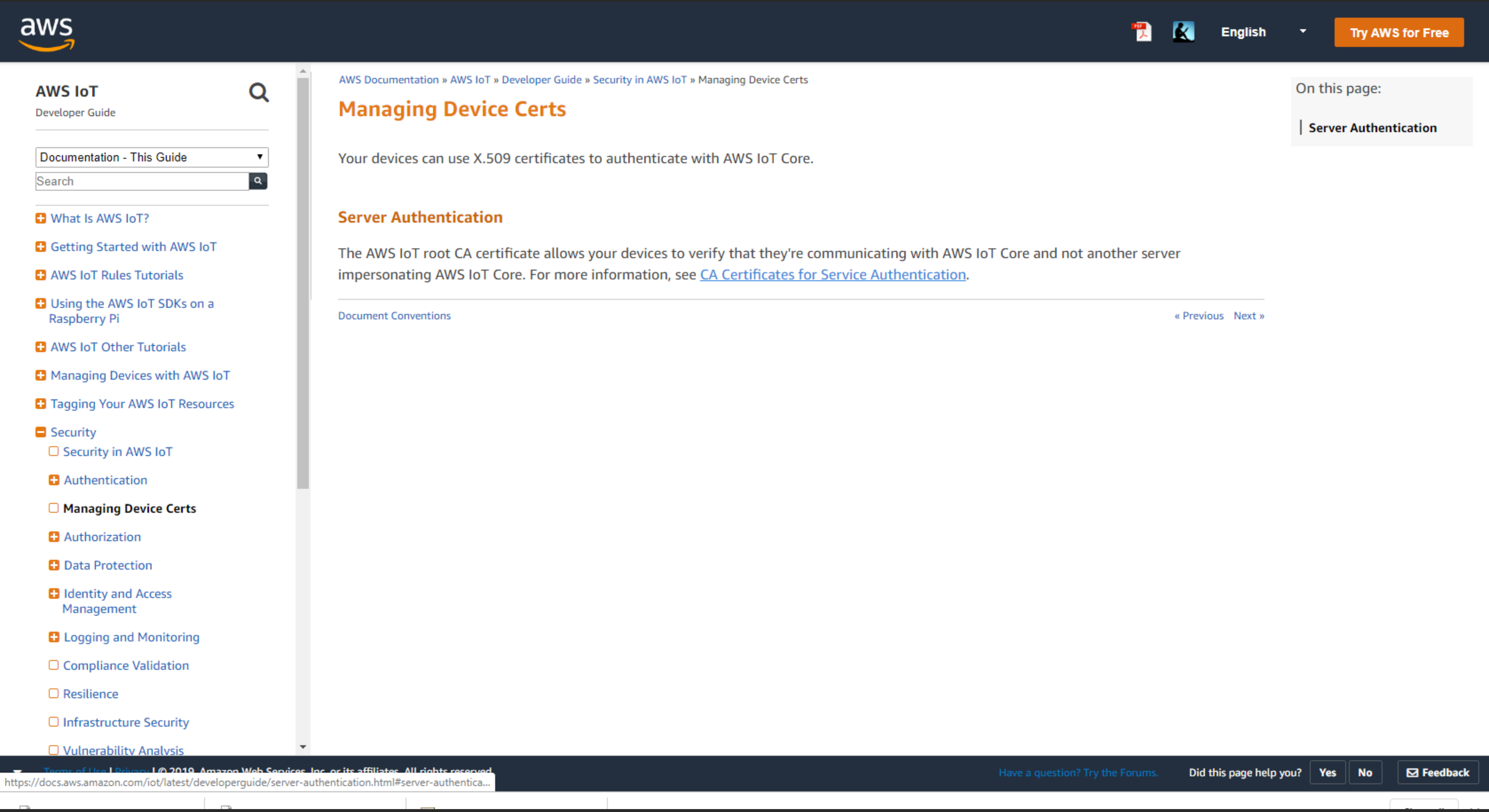
Certificate created!

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	6b148d4216.cert.pem	Download
A public key	6b148d4216.public.key	Download
A private key	6b148d4216.private.key	Download

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



AWS IoT
Developer Guide

Documentation - This Guide
Search

- What Is AWS IoT?
- Getting Started with AWS IoT
- AWS IoT Rules Tutorials
- Using the AWS IoT SDKs on a Raspberry Pi
- AWS IoT Other Tutorials
- Managing Devices with AWS IoT
- Tagging Your AWS IoT Resources
- Security
  - Security in AWS IoT
  - Authentication
  - Managing Device Certs
  - Authorization
  - Data Protection
  - Identity and Access Management
  - Logging and Monitoring
  - Compliance Validation
  - Resilience
  - Infrastructure Security
  - Vulnerability Analysis

AWS Documentation » AWS IoT » Developer Guide » Security in AWS IoT » Managing Device Certs

Managing Device Certs

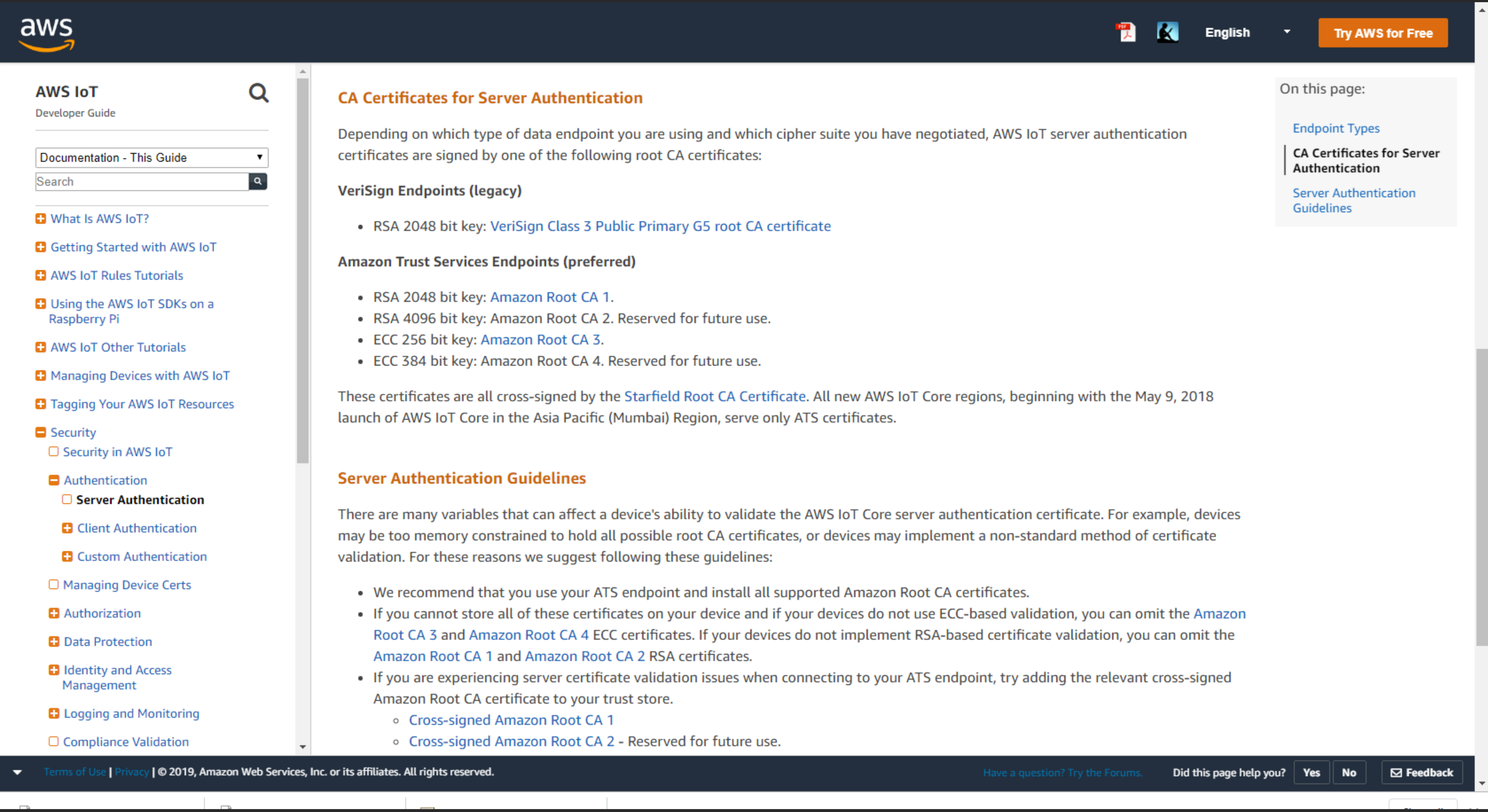
On this page:
Server Authentication

Your devices can use X.509 certificates to authenticate with AWS IoT Core.

Server Authentication

The AWS IoT root CA certificate allows your devices to verify that they're communicating with AWS IoT Core and not another server impersonating AWS IoT Core. For more information, see CA Certificates for Service Authentication.

Document Conventions
« Previous Next »



AWS IoT

Developer Guide

Documentation - This Guide

Search

CA Certificates for Server Authentication

Depending on which type of data endpoint you are using and which cipher suite you have negotiated, AWS IoT 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.

Server Authentication Guidelines

There are many variables that can affect a device's ability to validate the AWS IoT Core server authentication certificate. For example, devices may be too memory constrained to hold all possible root CA certificates, or devices may implement a non-standard method of certificate validation. For these reasons we suggest following these guidelines:

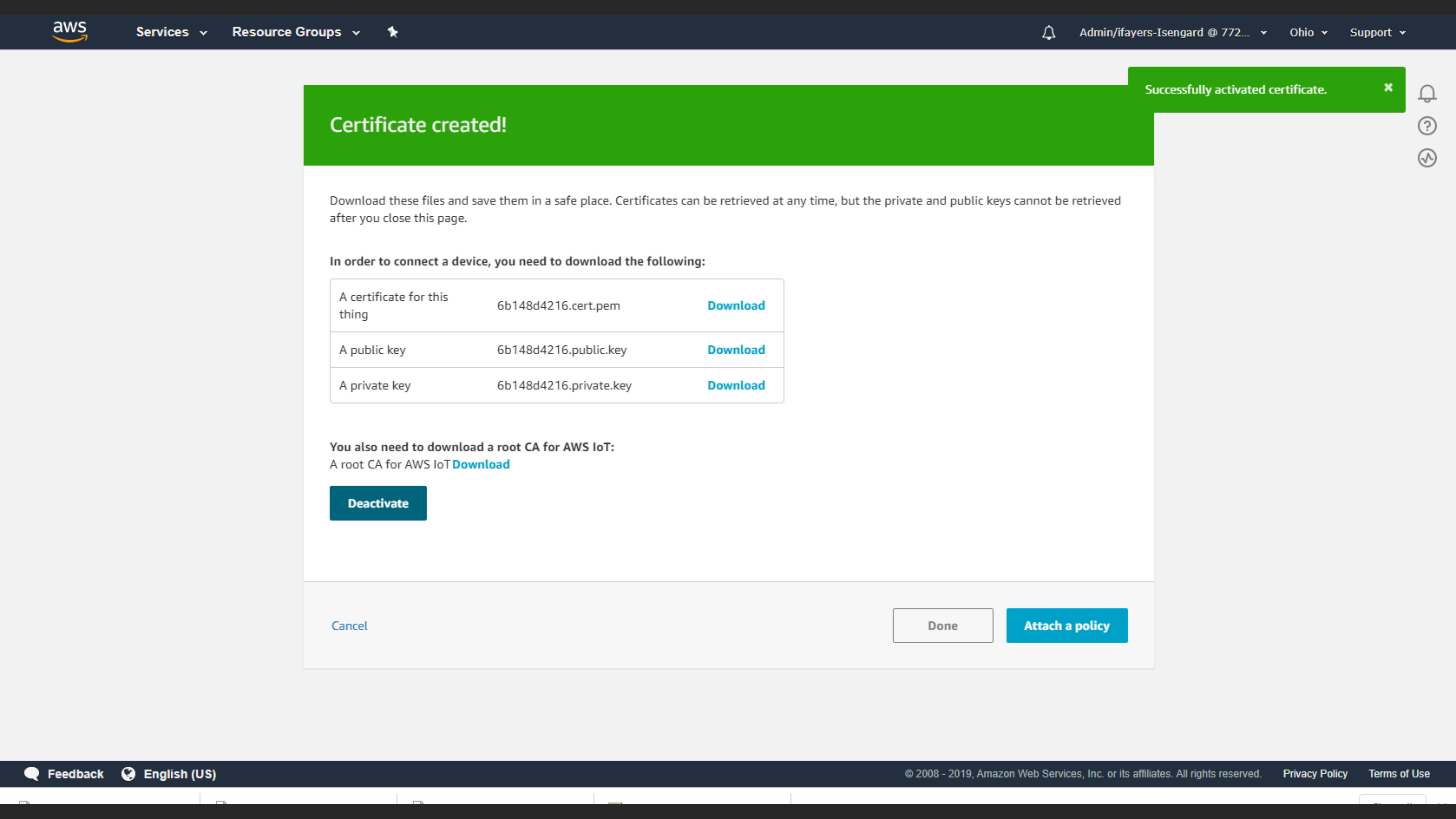
- We recommend that you use your ATS endpoint and install all supported Amazon Root CA certificates.
- If you cannot store all of these certificates on your device and if your devices do not use ECC-based validation, you can omit the Amazon Root CA 3 and Amazon Root CA 4 ECC certificates. If your devices do not implement RSA-based certificate validation, you can omit the Amazon Root CA 1 and Amazon Root CA 2 RSA certificates.
- If you are experiencing server certificate validation issues when connecting to your ATS endpoint, try adding the relevant cross-signed Amazon Root CA certificate to your trust store.
  - Cross-signed Amazon Root CA 1
  - Cross-signed Amazon Root CA 2 - Reserved for future use.

On this page:

Endpoint Types

CA Certificates for Server Authentication

Server Authentication Guidelines



Successfully activated certificate.

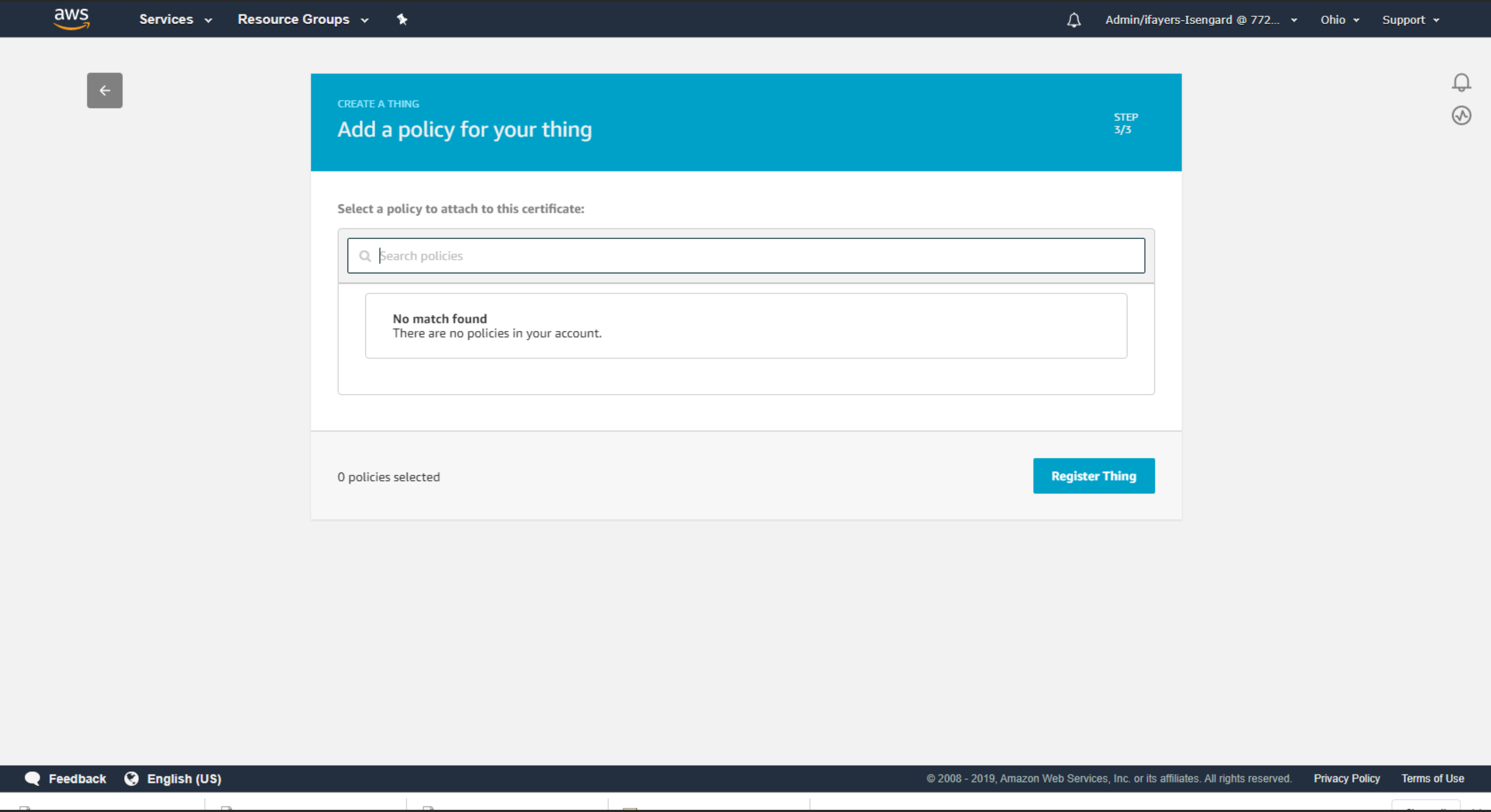
Certificate created!

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	6b148d4216.cert.pem	Download
A public key	6b148d4216.public.key	Download
A private key	6b148d4216.private.key	Download

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



CREATE A THING

Add a policy for your thing

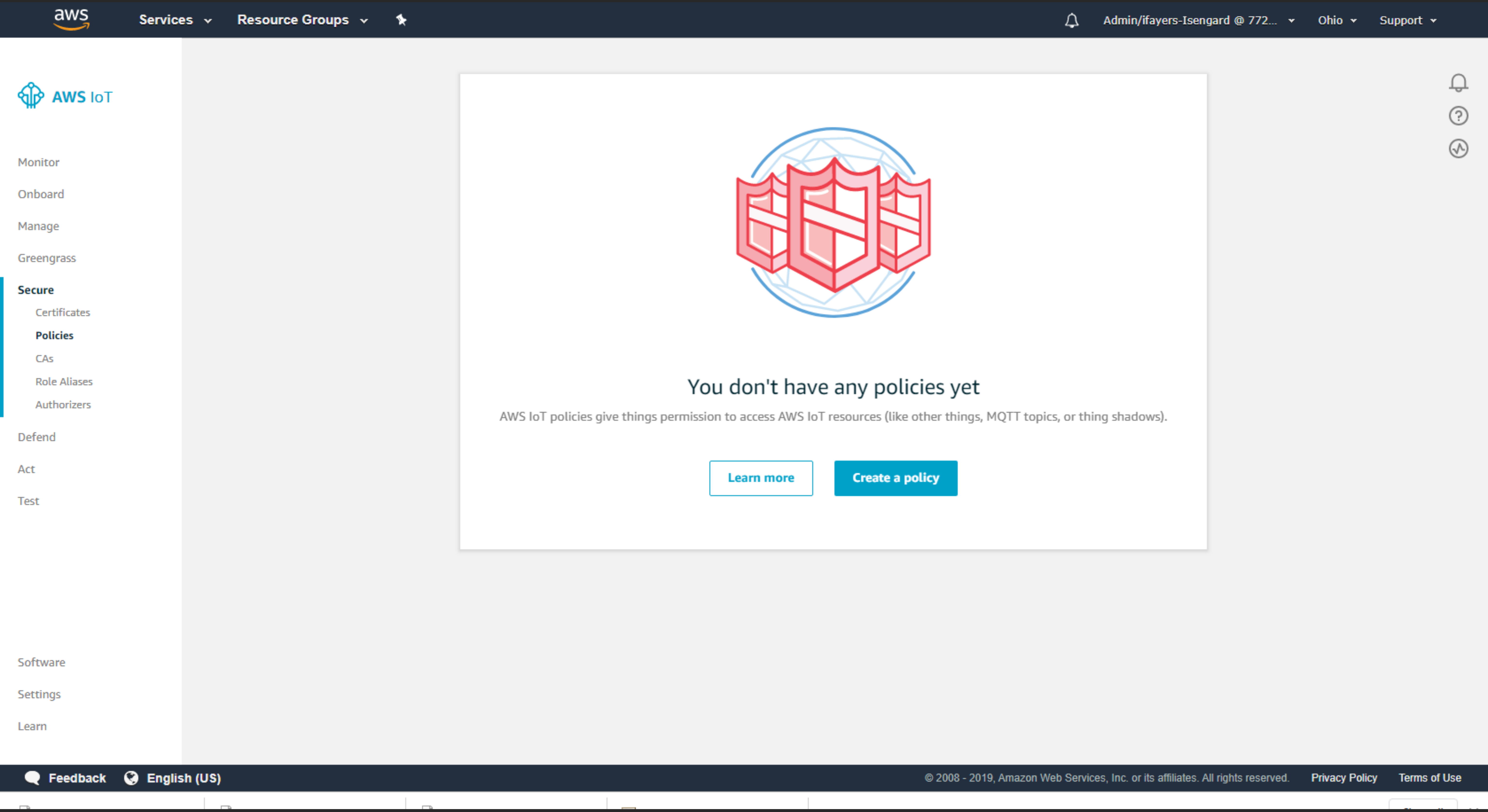
STEP 3/3

Select a policy to attach to this certificate:

No match found
There are no policies in your account.

0 policies selected

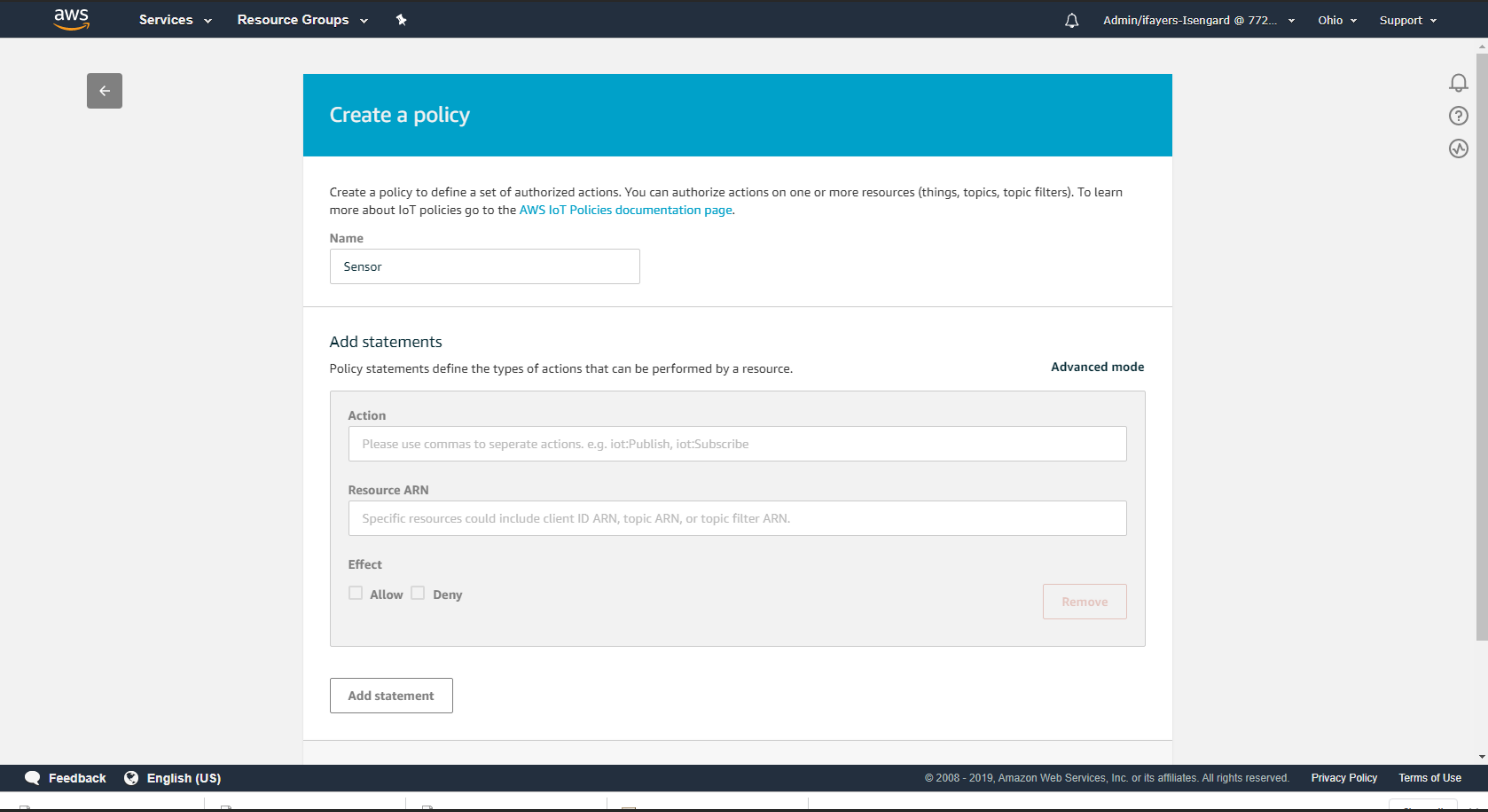
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



You don't have any policies yet

AWS IoT policies give things permission to access AWS IoT resources (like other things, MQTT topics, or thing shadows).

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



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: Sensor

Add statements

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

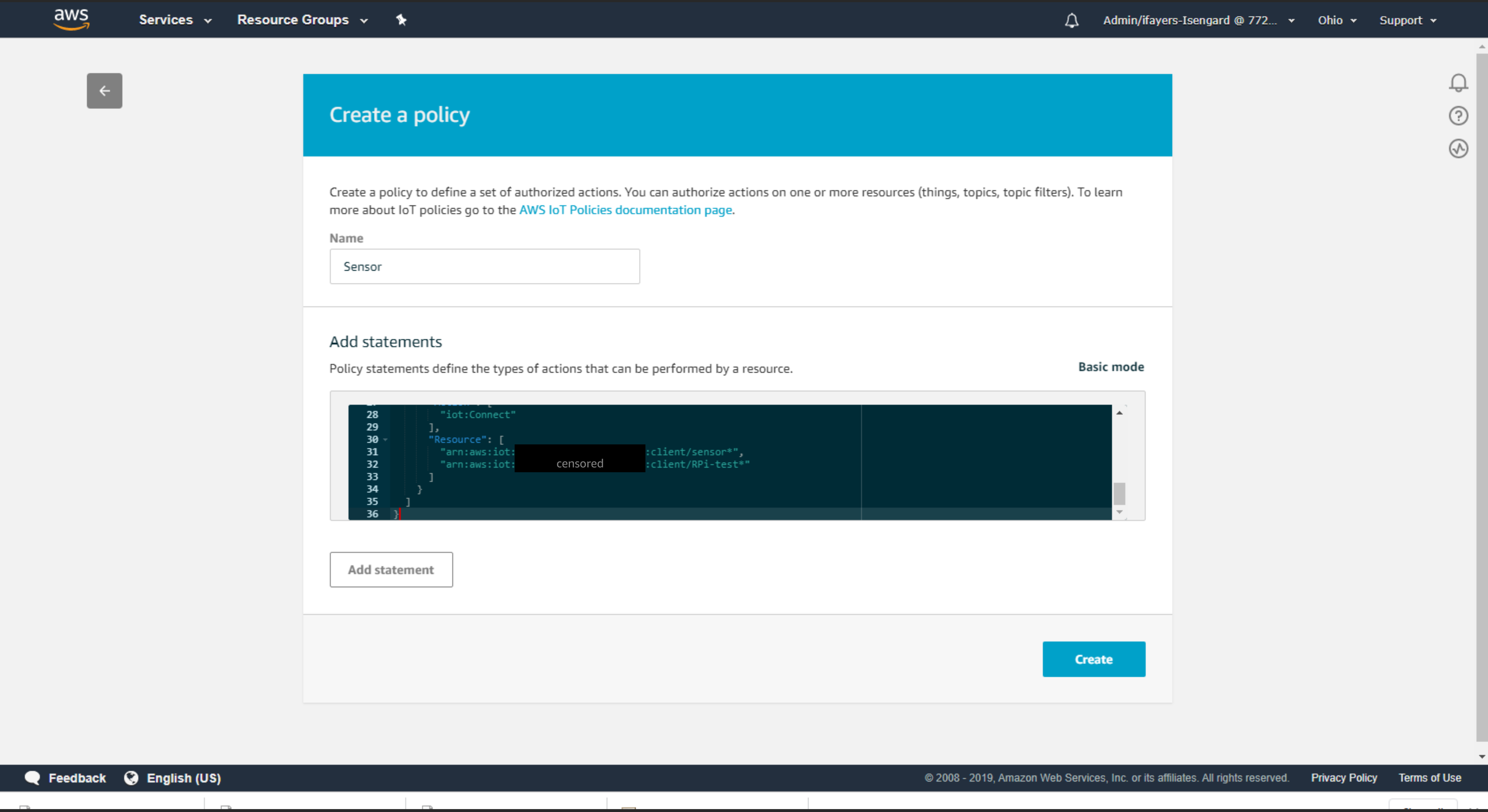
Action: Please use commas to seperate actions. e.g. iot:Publish, iot:Subscribe

Resource ARN: Specific resources could include client ID ARN, topic ARN, or topic filter ARN.

Effect

□ Allow □ Deny

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Create a policy

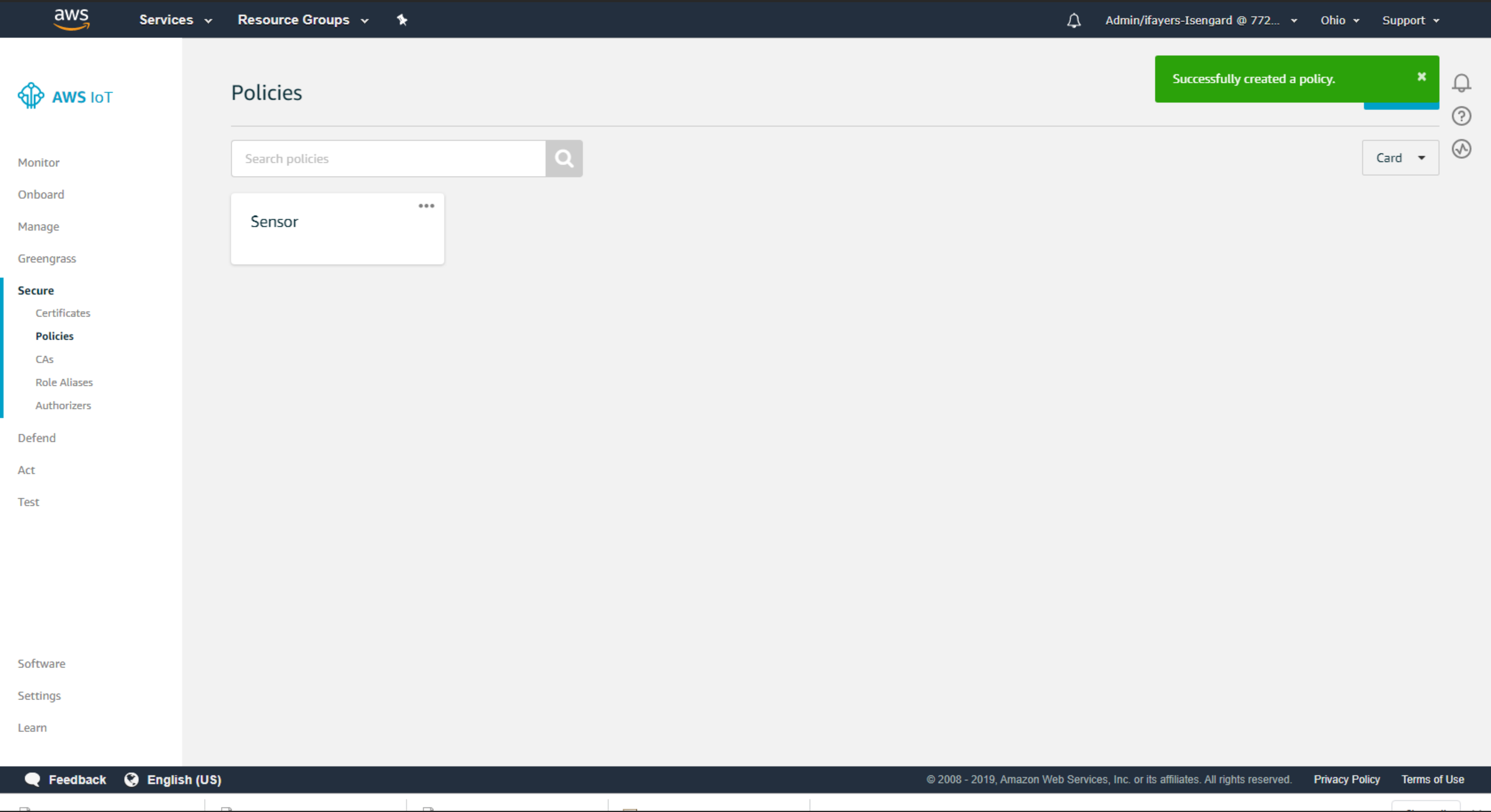
Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.

Name: Sensor

Add statements

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

```
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:          :client/sensor*",
        "arn:aws:iot: censored :client/RPi-test*"
      ]
    }
  ]
}
```

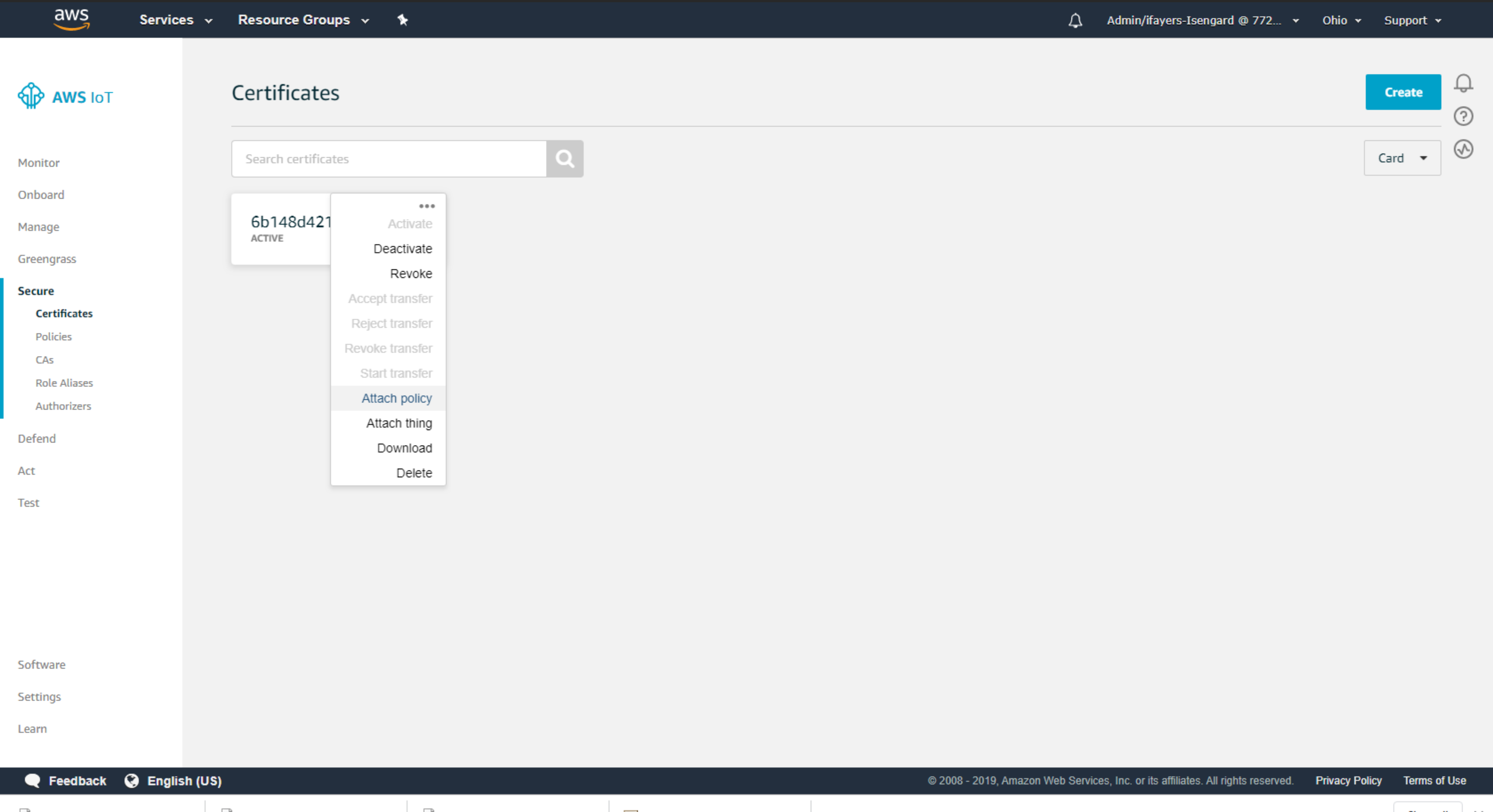


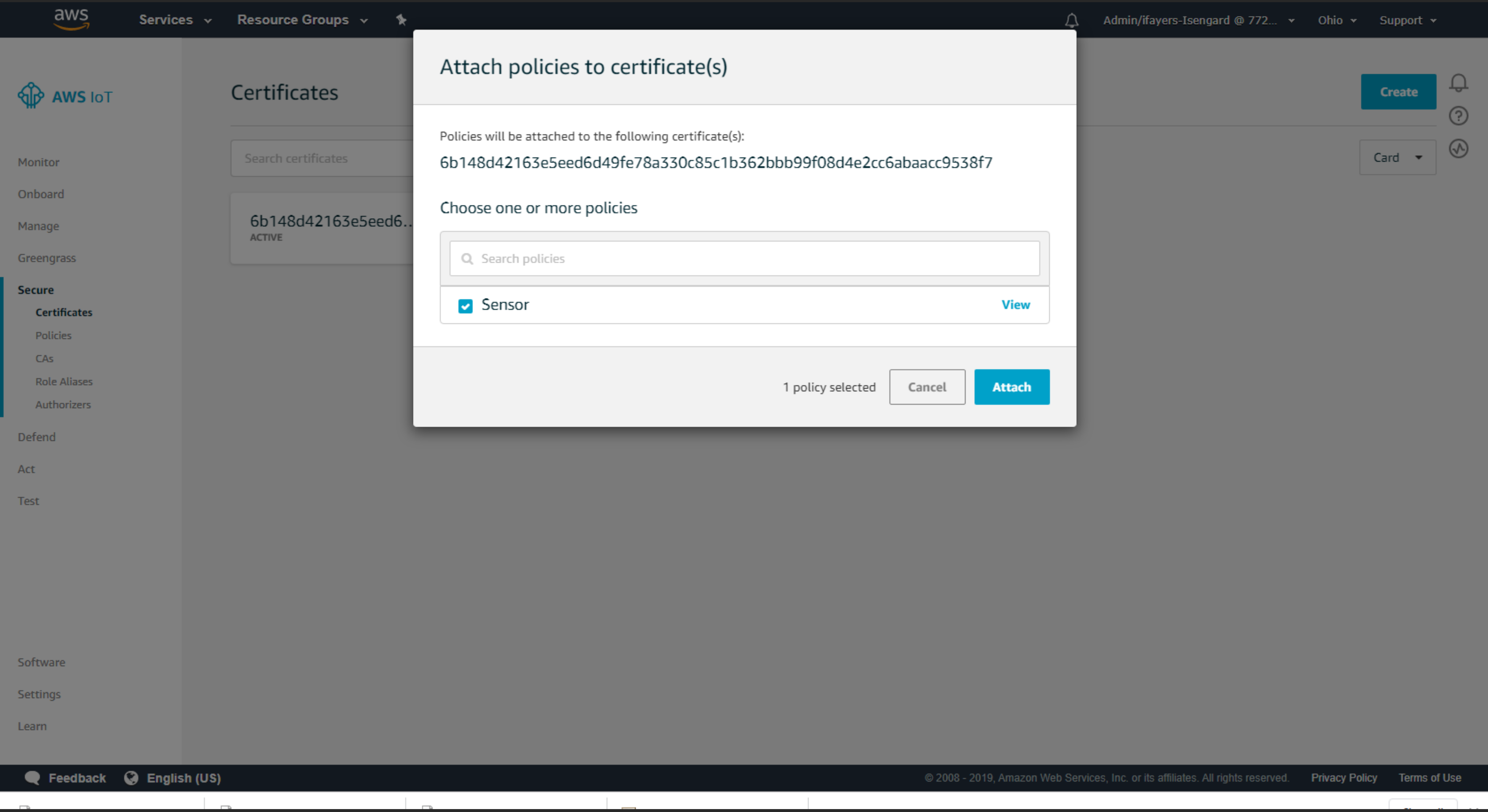
Policies

Successfully created a policy.

Sensor

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.


aws
Services
Resource Groups
Admin/ifayers-Isengard @ 772...
Ohio
Support
AWS IoT
Monitor
Onboard
Manage
Greengrass
Secure
Certificates
Policies
CAs
Role Aliases
Authorizers
Defend
Act
Test
Software
Settings
Learn
Certificates
Create
Search certificates
Card
6b148d421
ACTIVE
•••
Activate
Deactivate
Revoke
Accept transfer
Reject transfer
Revoke transfer
Start transfer
Attach policy
Attach thing
Download
Delete
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Attach policies to certificate(s)

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

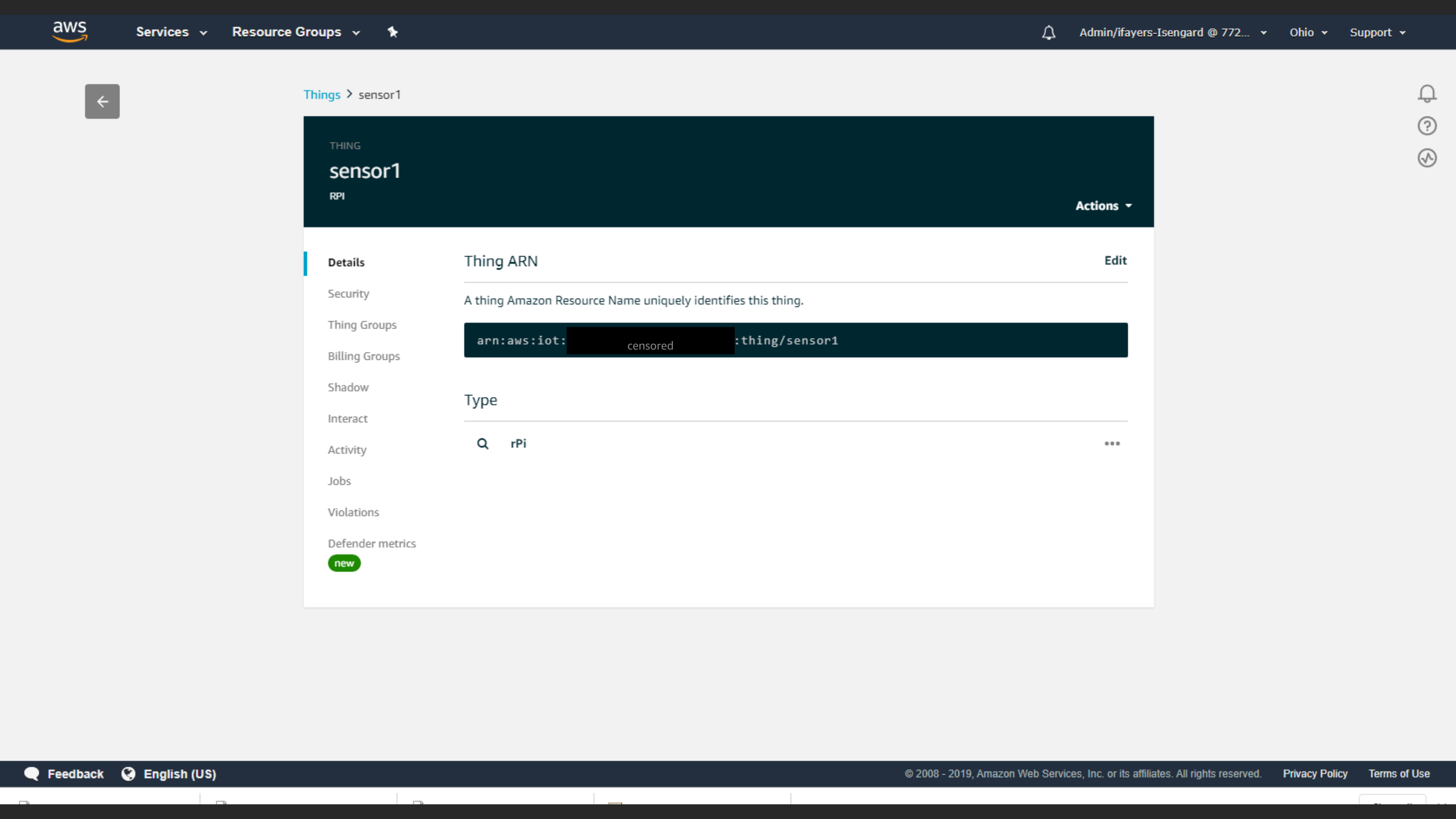
6b148d42163e5eed6d49fe78a330c85c1b362bbb99f08d4e2cc6abaacc9538f7

Choose one or more policies

☑ Sensor

1 policy selected

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



sensor1

RPI

Thing ARN

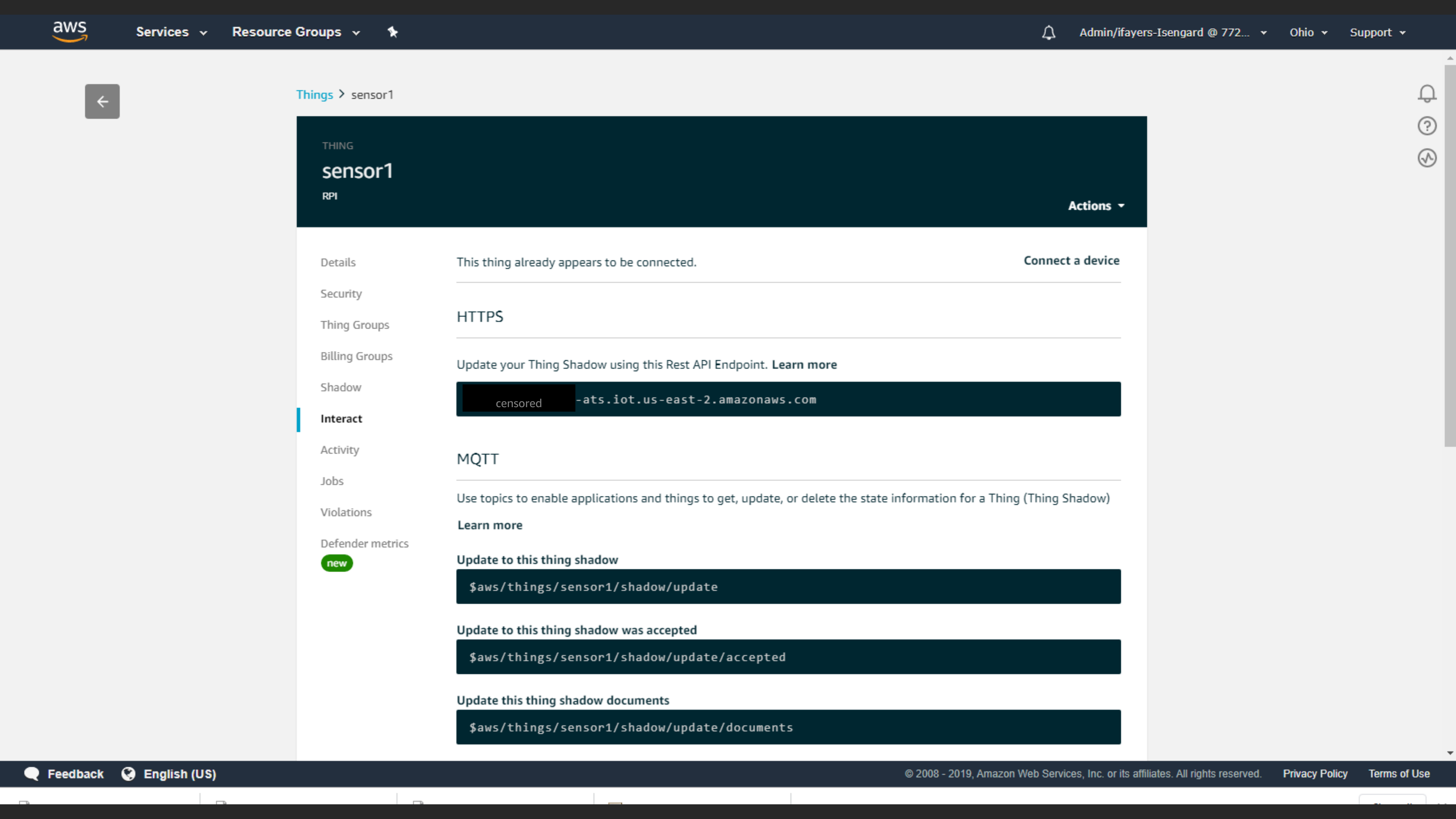
A thing Amazon Resource Name uniquely identifies this thing.

```
arn:aws:iot: censored :thing/sensor1
```

Type

rPi

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Things > sensor1

THING

sensor1

RPI

This thing already appears to be connected.

HTTPS

Update your Thing Shadow using this Rest API Endpoint. **Learn more**

```
censored-ats.iot.us-east-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

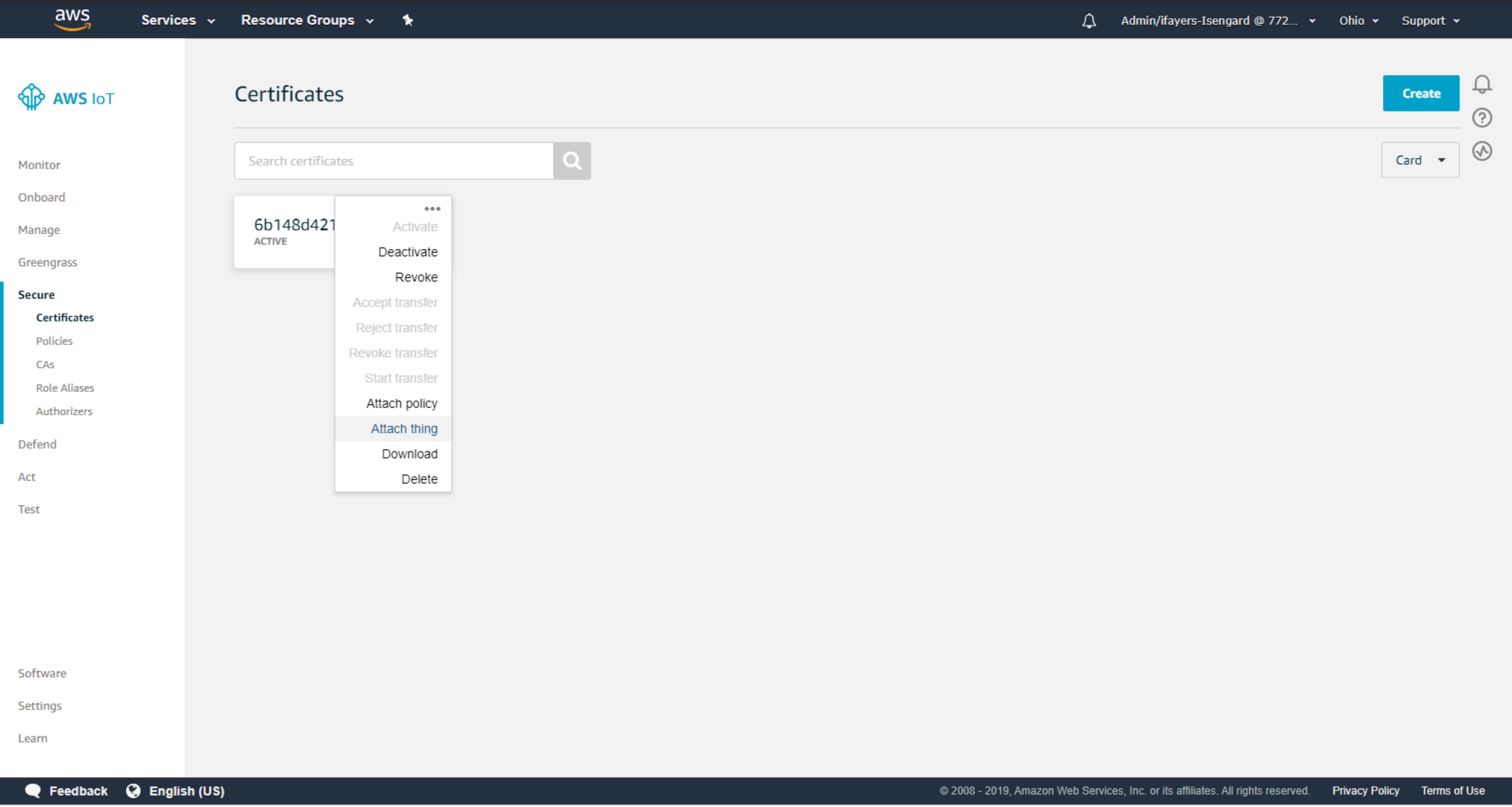
```
$aws/things/sensor1/shadow/update
```

Update to this thing shadow was accepted

```
$aws/things/sensor1/shadow/update/accepted
```

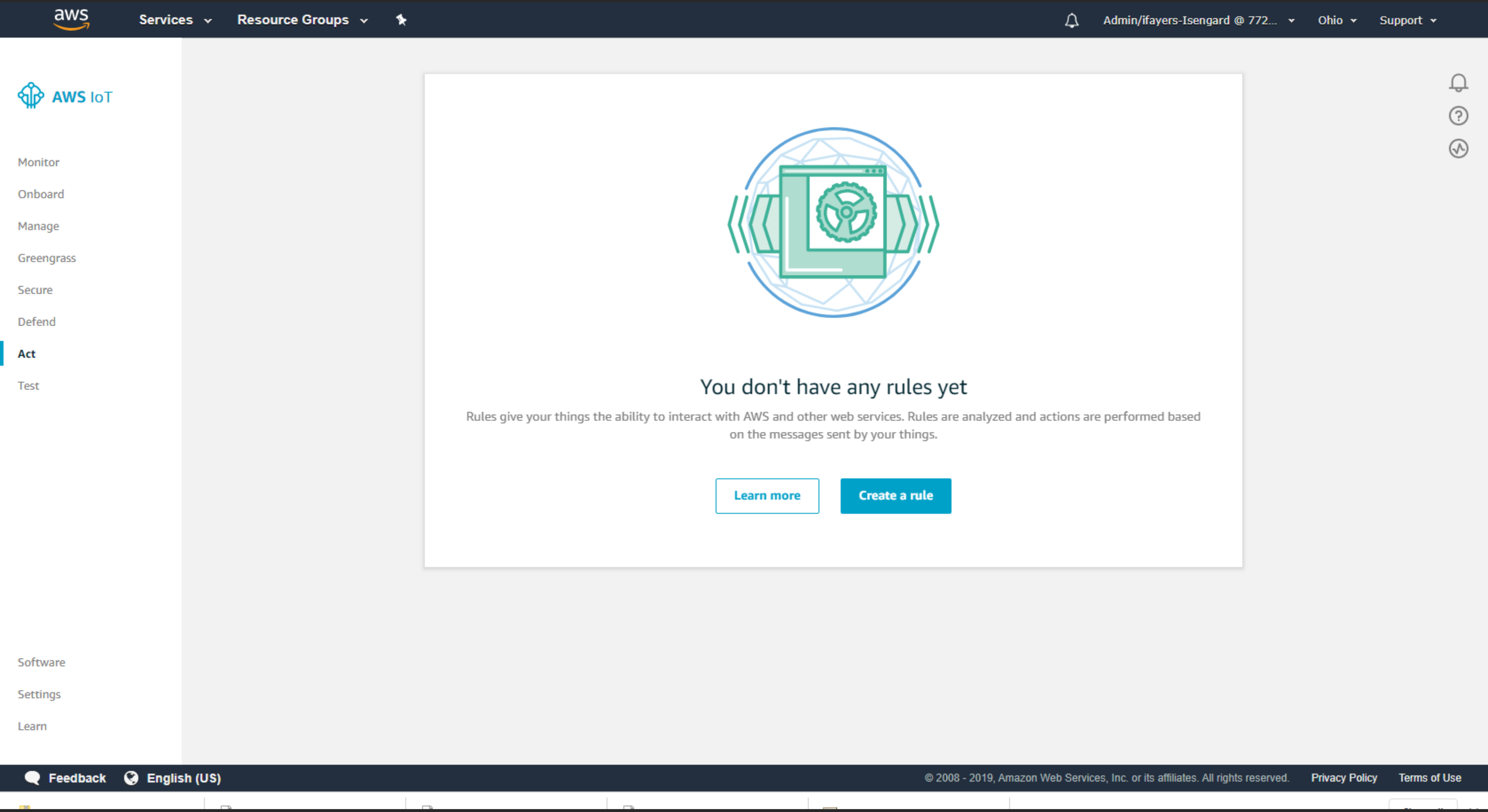
Update this thing shadow documents

```
$aws/things/sensor1/shadow/update/documents
```



Certificates

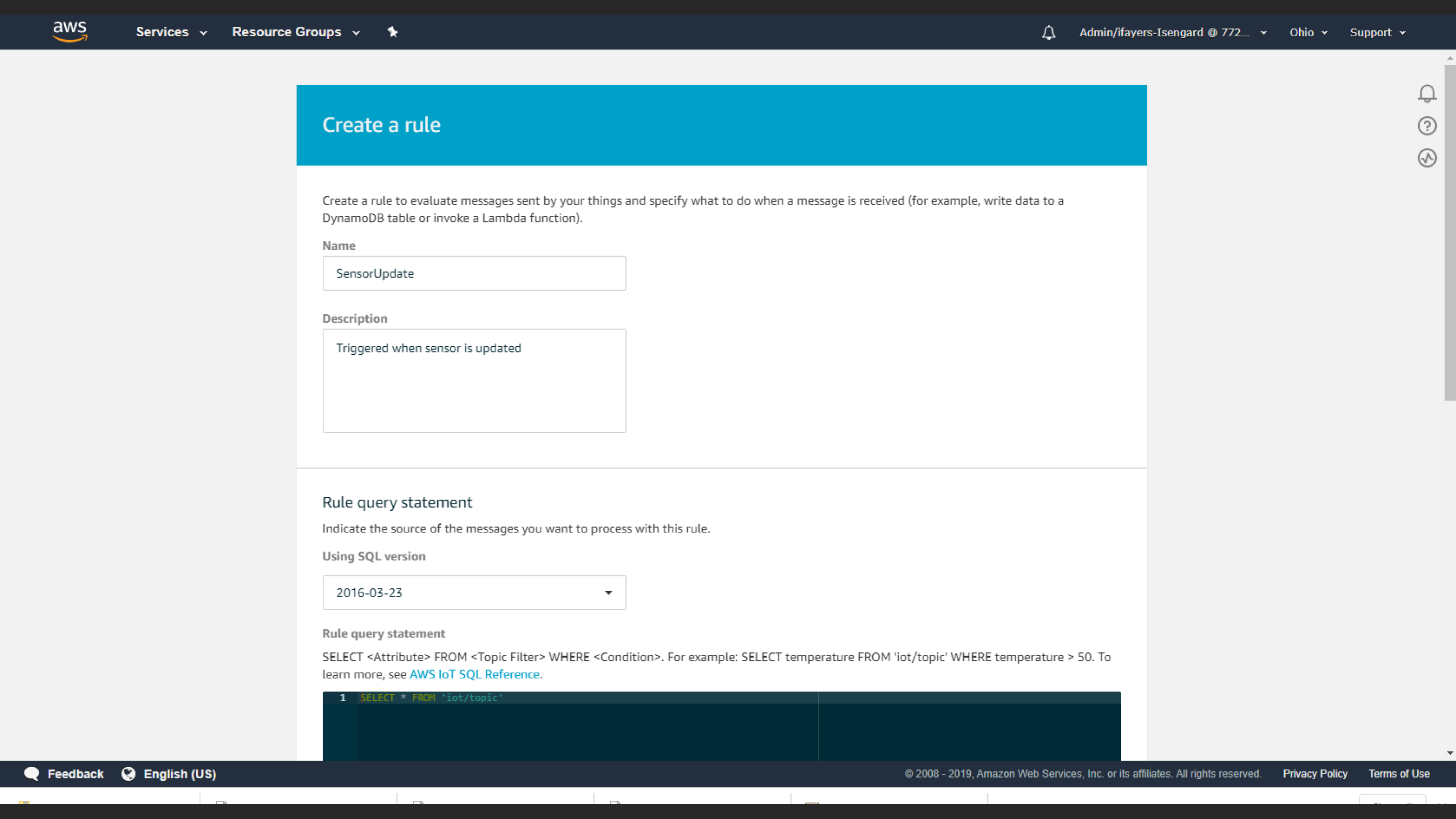
6b148d421
ACTIVE



You don't have any rules yet

Rules give your things the ability to interact with AWS and other web services. Rules are analyzed and actions are performed based on the messages sent by your things.

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

SensorUpdate

Description

Triggered when sensor is updated

Rule query statement

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

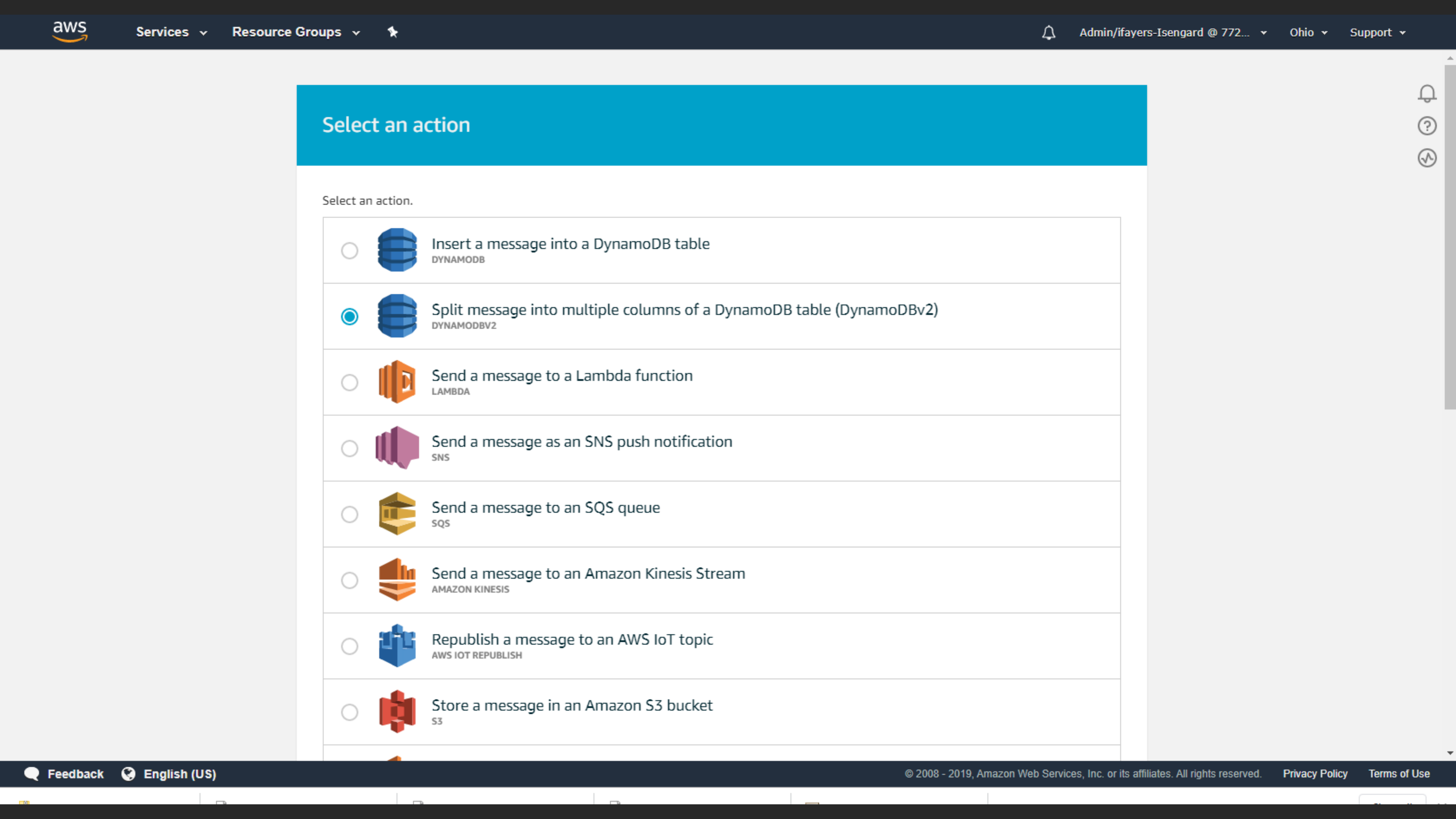
Using SQL version

2016-03-23

Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see AWS IoT SQL Reference.

```
SELECT * FROM 'iot/topic'
```



Select an action

Select an action.

○ Insert a message into a DynamoDB table
DYNAMODB

● Split message into multiple columns of a DynamoDB table (DynamoDBv2)
DYNAMODBV2

○ Send a message to a Lambda function
LAMBDA

○ Send a message as an SNS push notification
SNS

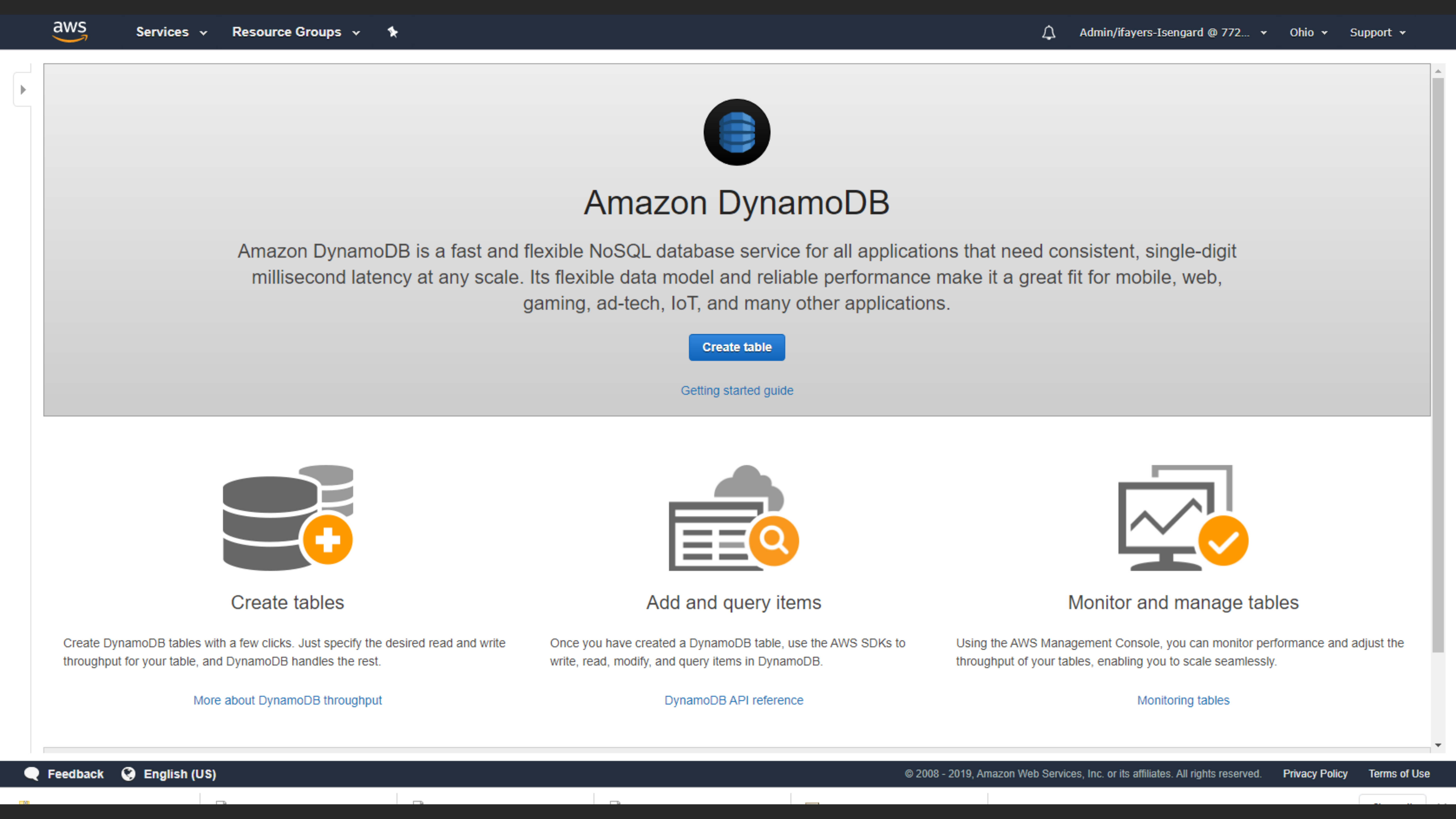
○ Send a message to an SQS queue
SQS

○ Send a message to an Amazon Kinesis Stream
AMAZON KINESIS

○ Republish a message to an AWS IoT topic
AWS IOT REPUBLISH

○ Store a message in an Amazon S3 bucket
S3

Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Amazon DynamoDB

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT, and many other applications.

Create tables

Create DynamoDB tables with a few clicks. Just specify the desired read and write throughput for your table, and DynamoDB handles the rest.

Add and query items

Once you have created a DynamoDB table, use the AWS SDKs to write, read, modify, and query items in DynamoDB.

Monitor and manage tables

Using the AWS Management Console, you can monitor performance and adjust the throughput of your tables, enabling you to scale seamlessly.

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

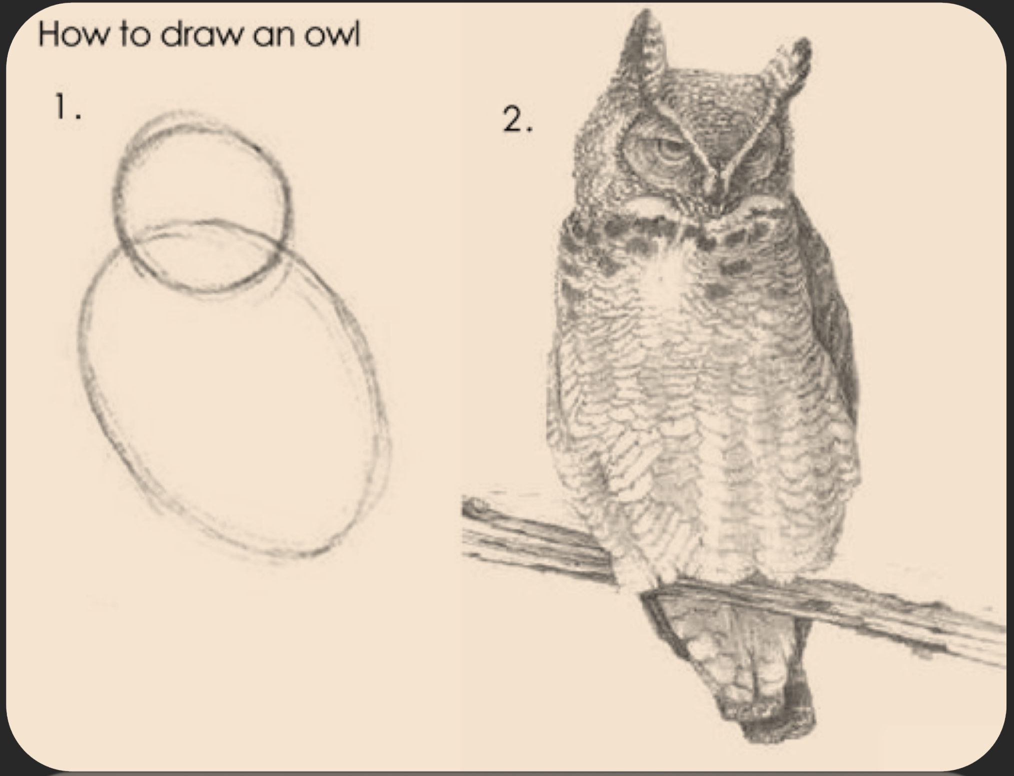
Once IoT is configured with a sensor and a rule, it will receive messages from a commissioned Raspberry Pi. DynamoDB,

Kinesis, S3 or other end-points is dependent upon implementation.

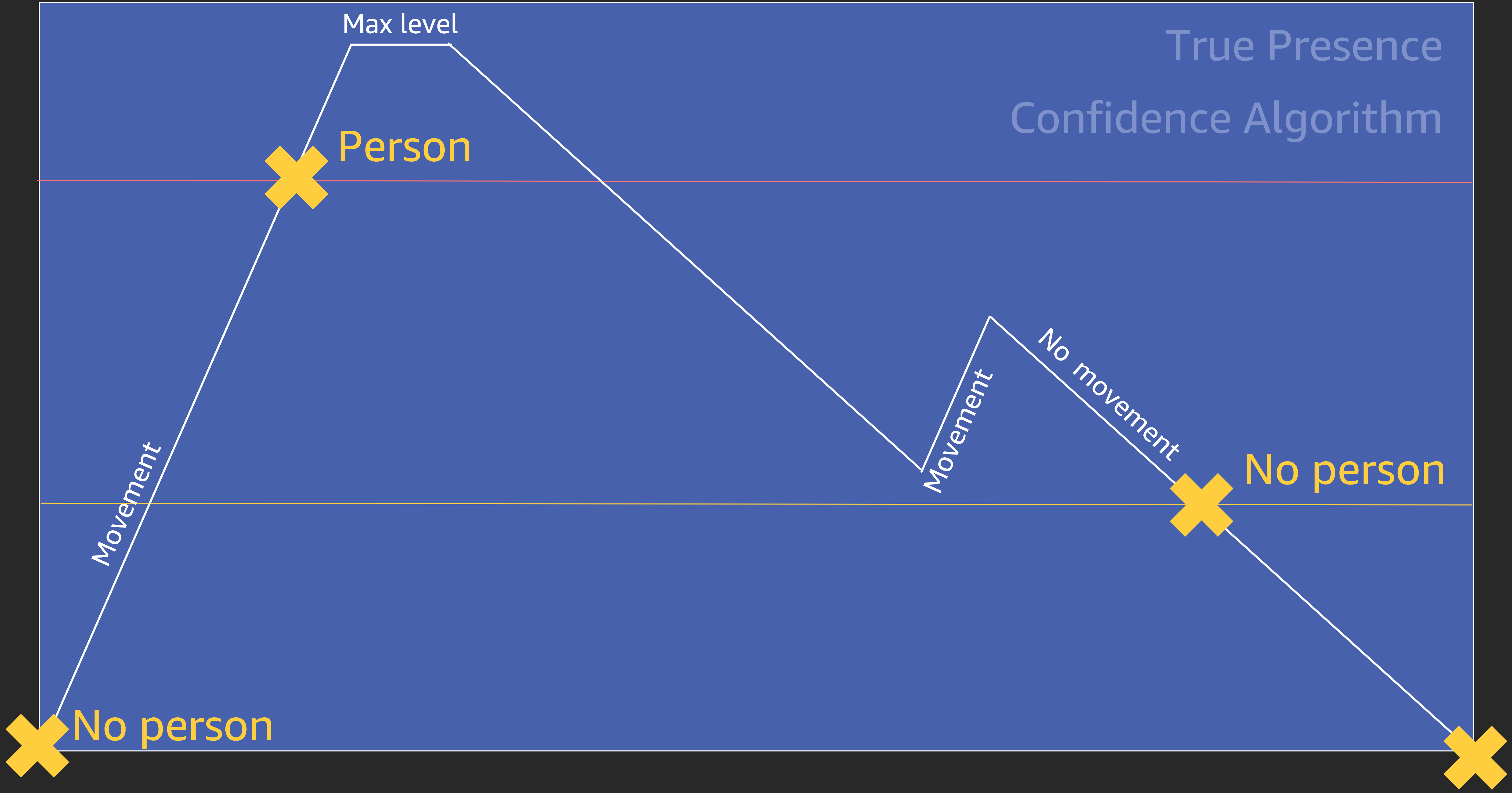
The Raspberry Pi Python code example uses a ‘confidence of person present’ algorithm and passes that data to the IoT core back-end.

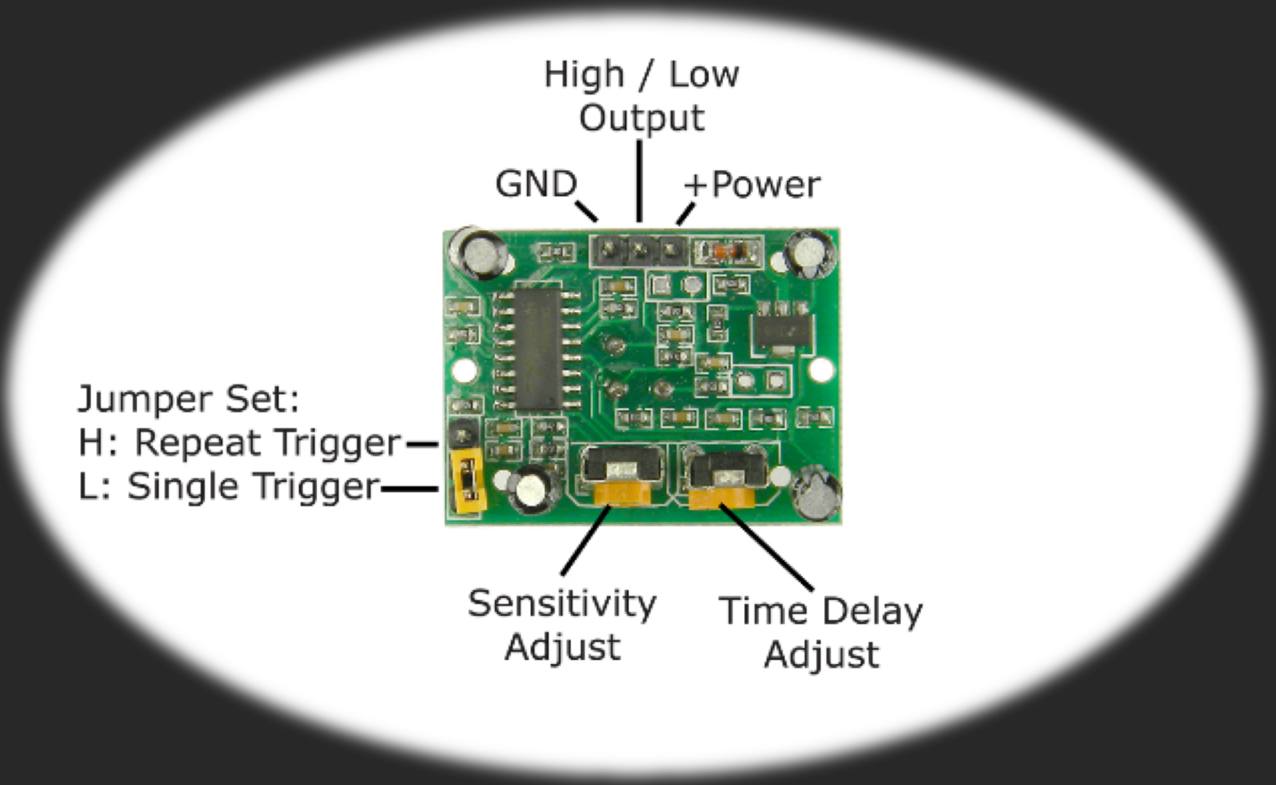
The algorithm depiction is next.

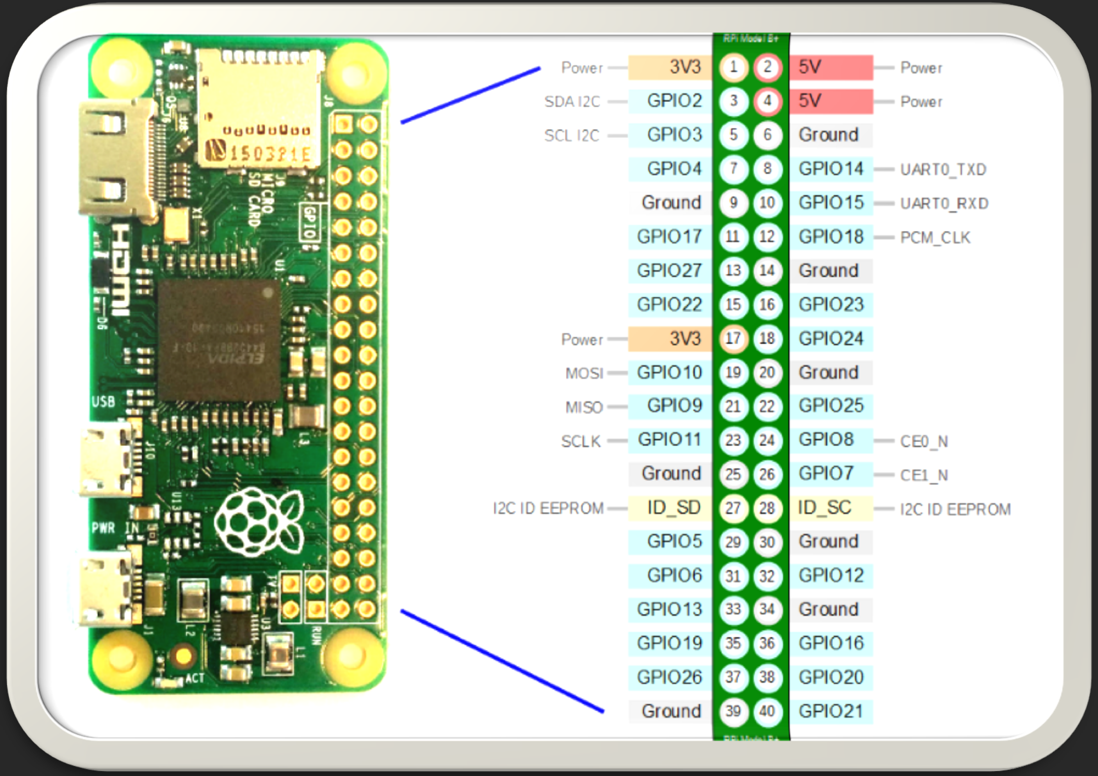
A hardware-helper finishes this pack.



True Presence Confidence Algorithm







Power	3V3	1	2	5V	Power
SDA I2C	GPIO2	3	4	5V	Power
SCL I2C	GPIO3	5	6	Ground	
	GPIO4	7	8	GPIO14	UART0_TXD
	Ground	9	10	GPIO15	UART0_RXD
	GPIO17	11	12	GPIO18	PCM_CLK
	GPIO27	13	14	Ground	
	GPIO22	15	16	GPIO23	
Power	3V3	17	18	GPIO24	
MOSI	GPIO10	19	20	Ground	
MISO	GPIO9	21	22	GPIO25	
SCLK	GPIO11	23	24	GPIO8	CE0_N
	Ground	25	26	GPIO7	CE1_N
I2C ID EEPROM	ID_SD	27	28	ID_SC	I2C ID EEPROM
	GPIO5	29	30	Ground	
	GPIO6	31	32	GPIO12	
	GPIO13	33	34	Ground	
	GPIO19	35	36	GPIO16	
	GPIO26	37	38	GPIO20	
	Ground	39	40	GPIO21	

The LED’s are optional, the motion-sensor is key.

We used a mix of sensors, above is the larger (adjustable) version.

Wiring of the sensor and LED’s is self-explanatory from the code.

Thank you for voting for us!