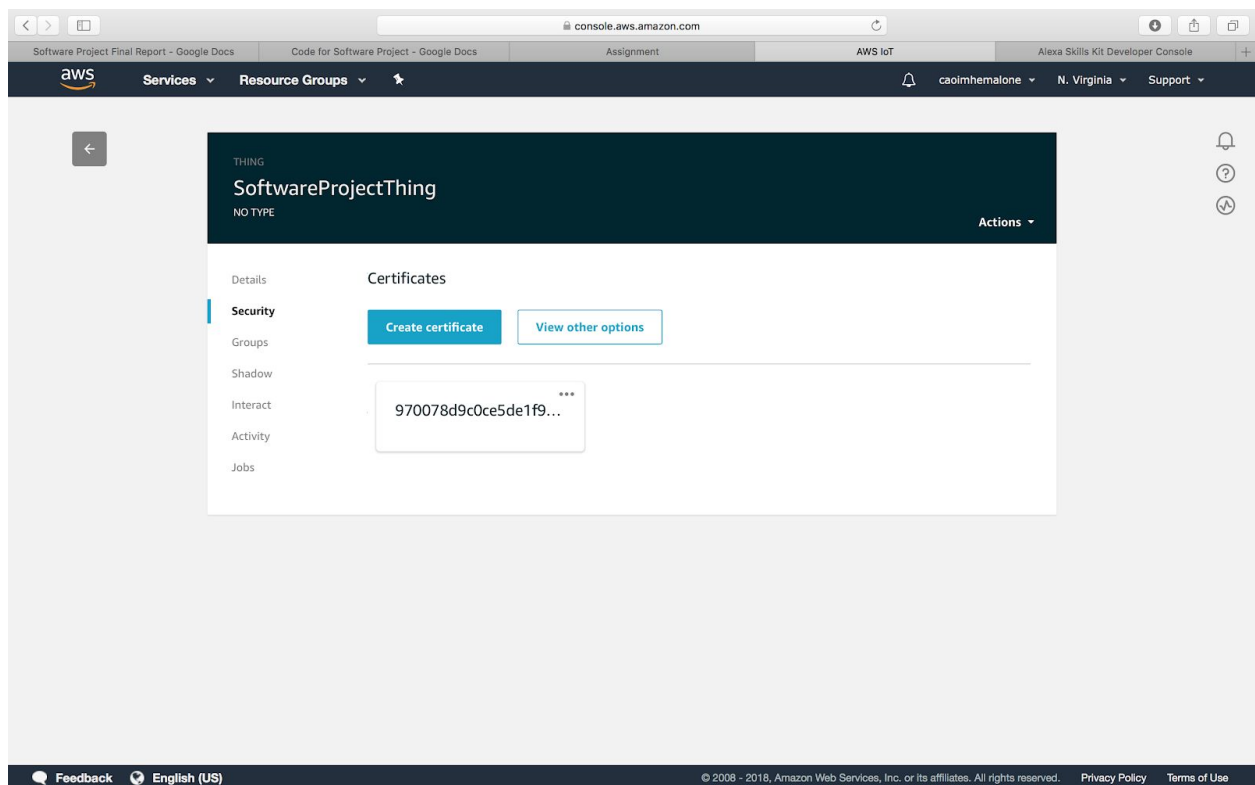


# Code for Software Project

Caoimhe Malone  
X14447022  
BSHC4 IOT  
Safe & Sound

Github link: <https://github.com/caoimhemalone/SoftwareProject>

AWS:



Alexa:

```
{  
  "interactionModel": {  
    "languageModel": {  
      "invocationName": "safeandsound",  
      "intents": [  
        {
```

```

    "name": "AMAZON.FallbackIntent",
    "samples": []
  },
  {
    "name": "AMAZON.CancelIntent",
    "samples": []
  },
  {
    "name": "AMAZON.HelpIntent",
    "samples": []
  },
  {
    "name": "AMAZON.StopIntent",
    "samples": []
  },
  {
    "name": "SystemOn",
    "slots": [],
    "samples": [
      "Alexa turn safeandsound on",
      "Alexa turn system on",
      "Alexa ask safeandsound to turn on ",
      "Alexa ask safeandsound to turn system on"
    ]
  },
  {
    "name": "SystemOff",
    "slots": [],
    "samples": [
      "Alexa ask safeandsound to turn off",
      "Alexa stop safeandsound",
      "Alexa turn system off",
      "Alexa ask safeandsound to stop"
    ]
  }
],
"types": []
}
}
}

```

Lambda:

```
import boto3
```

```
access_key = "AKIAIUHYRDMGFNB2JPQ"
```

```
access_secret = "c/gPYfztzoqq1aBdNbLzsTtwCx4C5QSjuhY5Jphc"
```

```
region = "us-east-1"
```

```
queue_url = " https://sqs.us-east-1.amazonaws.com/688000642875/AlexaScriptPi"
```

```
def build_speechlet_response(title, output, reprompt_text, should_end_session):
```

```
    return {
        'outputSpeech': {
            'type': 'PlainText',
            'text': output
        },
        'card': {
            'type': 'Simple',
            'title': "SessionSpeechlet - " + title,
            'content': "SessionSpeechlet - " + output
        },
        'reprompt': {
            'outputSpeech': {
                'type': 'PlainText',
                'text': reprompt_text
            }
        },
        'shouldEndSession': should_end_session
    }
```

```
def build_response(session_attributes, speechlet_response):
```

```
    return {
        'version': '1.0',
        'sessionAttributes': session_attributes,
        'response': speechlet_response
    }
```

```
def post_message(client, message_body, url):
```

```
    response = client.send_message(QueueUrl = url, MessageBody= message_body)
```

```
def lambda_handler(event, context):
```

```
    client = boto3.client('sqs', aws_access_key_id = access_key, aws_secret_access_key =
access_secret, region_name = region)
```

```
    intent_name = event['request']['intent']['name']
```

```
    if intent_name == "SystemOn":
```

```
    post_message(client, 'on', queue_url)
    message = "on"
elif intent_name == "SystemOff":
    post_message(client, 'off', queue_url)
    message = "off"
else:
    message = "Unknown"

speechlet = build_speechlet_response("System Status", message, "", "true")
return build_response({}, speechlet)
```