Instructions

Alexa example skill for Cisco Room kit

These instructions are to build a sample Alexa skill using JavaScript with Node.js that demonstrates how to control a Cisco Room kit using the xAPI.

Setup the Alexa Smart Home Sample via the Alexa Developer Console

1. Setup the Sample Resources
2. Get the Sample Source Code
3. Setup Cisco Webex integration
4. Create the Skill
5. Setup the Lambda Function
6. Configure the Skill
7. Setup the DynamoDB Table
8. Deploy the Sample Code
9. Test the Skill
10. Review the Skill Logs

Setup the Sample Resources

To complete this sample, check for the required accounts and tools as well as prepare a setup scratch file to store configuration variables.

Check for the Required Accounts

To work with this sample, you will need both an Amazon Developer account and an Amazon Web Services account.

- An Amazon Developer account. This is required to create and configure Alexa skills.
- An Amazon Web Services (AWS) account. This is required to host the skill logic in an AWS Lambda function.

Create a Working Directory

Create a folder on your computer Desktop called `example-cisco-skill`. This folder will store the code and configuration files needed to complete the sample.

Create a Setup Configuration File

The `setup.txt` configuration file is used to store IDs and other values during configuration of the environment.

1. Create your own setup.txt file in that directory and copy the contents outlined in the next step.
2. Open and review the `setup.txt` file. The file should contain:

```
 [Cisco Client ID]
amzn1.application-oa2-client.XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[Cisco Client Secret]
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[Alexa Skill Application ID]
amzn1.ask.skill.XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

[AWS Lambda ARN]
arn:aws:lambda:us-east-1:XXXXXXXXXXXX:function:cisco-skill

[Redirect URLs]
https://pitangui.amazon.com/api/skill/link/XXXXXXXXXXXXXX
https://layla.amazon.com/api/skill/link/XXXXXXXXXXXXXX
https://alexa.amazon.co.jp/api/skill/link/XXXXXXXXXXXXXX
```

These placeholders represent the configuration entities to be collected for the sample environment.

> Keep secrets safe. If a Client Secret is compromised or needs to be reset, you will have to discard the secret and regenerate the Client ID and Secret again or recreate the profile. This will immediately sever the existing access relationships and customers will have to re-authenticate or re-link their account or skill.

Install Optional Tools

The use of these tools is optional as they are useful, but not required.

Alexa Skills Kit Command Line Interface (ASK CLI)

This ASK CLI can be used to create, deploy, and maintain a Alexa skills. If you have the ASK CLI installed already you can can verify its version by running the command:

```
ask --version
```

For the purposes of this sample, the ask version should be 1.3.1 or greater.

If you do not have the ASK CLI installed or need to update to a version greater than 1.3, follow theQuick Start Alexa Skills Kit Command Line Interface (ASK CLI) instructions.

Get the Sample Source Code

Download and unzip the zipped contents of the sample from into your desktop working directory: `cisco-`

`skill`.

Setup Cisco integration

For Smart Home Skills it is required to have an OAuth account linking flow set up.

Create a new integration in Cisco developer portal.

To facilitate account linking in the Smart Home sample code, a registered integration is needed to generate a Client ID and Client Secret to use during the configuration of the Alexa skill.

1. In your web browser, go to https://developer.webex.com/docs/integrations.
2. Click the **Create an Integration** button.
3. On the *New integration* page, enter `Cisco skill` for the Integration Name, contact email address, logo, and description.
4. In the redirect URL, copy-paste the redirect URLs found in
5. Select *spark:xapi_commands*, *spark:xapi_statuses*, *spark-admin:devices_read* as scope.
6. Save the integration and copy
7. Copy the displayed Client ID and Client Secret values and save them to the `setup.txt` file in the `working-cisco-skill` directory replacing the format example entries for [Cisco Client ID] and [Cisco Client Secret] respectively.

```
 [Cisco Client ID]
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[Cisco Client Secret]
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Create the Skill

Create an Alexa Smart Home Skill that will respond to Smart Home commands.

Create a Skill via the Alexa Skills Kit Developer Console

1. In a web browser to the *Alexa Skills Kit Developer Console* at https://developer.amazon.com/alexa/console/ask. If not already authenticated, you may have to Sign In with your Amazon Developer Account.
2. Click the **Create Skill** button.
3. For the *Skill name*, enter `cisco-skill`.
4. Leave the *Default language* as **English (US)**.
5. Under *Choose a model to add to your skill* select the **Smart Home model**.
6. Click the **Create a skill** button. When completed, you should be at the configuration page for a newly created *cisco-skill* page.
7. In the *Smart Home service endpoint* section, locate the *Your Skill ID* value and click the **Copy to**

clipboard link to copy and then paste the Alexa Skill Application ID into the [Alexa Skill Application ID] section of your working `setup.txt` file in your `working-cisco-skill` directory.

Setup the Lambda Function

Create a new AWS Lambda function that will handle the directives from Alexa.

Create a Policy for the Lambda Function

1. Navigate to the IAM Management Console policies at https://console.aws.amazon.com/iam/home?region=us-east-1#/policies.
2. Click the **Create policy** button.
3. Select the *JSON* tab and then copy & paste the following policy into the text area:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "dynamodb:UpdateItem",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Click the **Review policy** button.
5. In the *Review policy* section, set the **Name** of the policy to `cisco-skill`.
6. Click the **Create policy** button to create the policy.

Create a Lambda Execution Role

1. Navigate to the IAM Management Console roles at https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles.
2. Click the **Create Role** button.
3. On the *Create role* page, select **AWS Service** for the type of trusted entity and then select **Lambda** from the list of services.
4. Click the **Next: Permissions** button.
5. When prompted to *Attach permissions policies*, filter and find the previously create *cisco-skill* policy and select its check box.

6. Click the **Next: Review** button.
7. In the *Review* section, set the **Role name** to `lambda_cisco_skill`.
8. Click **Create role** to create the execution role.

Create a Lambda Function

1. Navigate to the AWS Lambda console at https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1.
2. Verify you are in N. Virginia (us-east-1).
3. Click on the **Create function** button to start creating an AWS Lambda function.
4. On the *Create function* page, select the **Author from scratch** option.
5. For the Name of the function, enter `cisco-skill`.
6. For the Runtime, select **Node.js 10.x**.
7. Choose an existing role and select *lambda_cisco_skill* from the **Existing Role** options.
8. Click the **Create function** button to create the AWS Lambda function.
9. From the Lambda function page, copy the ARN from the top right of the page and save it into the `setup.txt` file in the [AWS Lambda ARN] section. The ARN should look something like: `arn:aws:lambda:us-east-1:XXXXXXXXXXXX:function:cisco-skill`

Add a Smart Home Trigger to the Lambda Function

1. On the Lambda function page for `cisco-skill`, select the *Configuration* tab if not already selected. Optionally, you can browse to https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/cisco-skill?tab=triggers. Note the *Add triggers* section on the left menu. This part of the designer allows you to add triggers to your Lambda function.
2. In the *Configuration* tab, select **Alexa Smart Home** from the left menu and it should be added to the `cisco-skill` function as a trigger. The Alexa Smart Home trigger will report "Configuration required" until the corresponding Alexa Skill Application ID is entered as part of the configuration.
3. Select the **Alexa Smart Home - Configuration required** box and locate the *Configure triggers* section at the bottom of the page. Paste the Alexa Skill Application ID value from the setup.txt file [Alexa Skill Application ID] section into the Application ID text box. The Alexa Skill Application ID should look something like: `amzn1.ask.skill.XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX`
4. Verify **Enable trigger** checkbox is checked and then click **Add**.
5. The added trigger will have unsaved changes, to finish enabling the trigger, click **Save** for the function at the top right of the page. Once saved, the Alexa Smart Home trigger should report and ID and be associated with the Alexa Skill Application ID.

Configure the Skill

Configure the Alexa skill and fill out the required settings.

Configure the Smart Home settings

1. Return to the Alexa Skills Kit Developer Console at https://developer.amazon.com/alexa/console/ask

and open your cisco skill.
2. On the *SMART HOME* tab, leave the Payload version at v3.
3. For the *Smart Home service endpoint*, enter the ARN saved in your `setup.txt` file as [AWS Lambda ARN] into the *Default endpoint* field. The ARN should look something like: `arn:aws:lambda:us-east-1:XXXXXXXXXXXX:function:cisco-skill`
4. Click **Save**. If you get an error stating "Failed to save skill information, Please make sure that "Alexa Smart Home" is selected for the event source type..." return to your AWS Lambda for the *cisco-skill* function and verify you correctly set and saved the Smart Home trigger.

Configure the Account Linking settings

1. Either select the *ACCOUNT LINKING *tab on the left or the**Setup Account Linking** button at the bottom of the page.
2. On the *Account Linking* page, for the *Authorization URI*, enter `https://api.ciscospark.com/v1/authorize`.
3. For the Access Token URI, enter `https://api.ciscospark.com/v1/access_token`.
4. For the Client ID, copy and paste the previously saved [Cisco Client ID] value from the `setup.txt` file.
5. For the Client Secret, copy and paste the previously saved [Cisco Client Secret] value from the `setup.txt` file.
6. Under *Your Authentication Scheme* choose *Credentials in request body*.
7. Under *Scope*, click the **+ Add scope** link and then add the following scopes into the text box: `spark:xapi_commands`, `spark:xapi_statuses`, `spark-admin:devices_read*`
8. Under *Domain list*, add api.ciscospark.com and ciscospark.com.
9. Copy the three (3) redirect urls from the Redirect URLs section and save them to the [Redirect URLs] section of the `setup.txt` file.
10. Click **Save**.

Set the Allowed Return URLs

Using your account-specific values from the skill configuration section, collect the Redirect URLs and set them in the *Cisco Webex integrations* **Allowed Return URLs**.

1. Open https://developer.webex.com/my-apps/cisco-skill in another browser tab.
2. Select the **Cisco skill** integration.
3. In the Redirect URLs text input, add each of the saved Redirect URLs from the `setup.txt` file. You will need to click the **Add Another** link to add each Return/Redirect URL. Each of the Return/Redirect URLs will have a format similar to `https://pitangui.amazon.com/api/skill/link/XXXXXXXXXXXXXX`
4. Click **Save**.

Setup the Amazon DynamoDB Table

Create a new Amazon DynamoDB Table that will hold the state of the virtual switch.

Create a SampleSmartHome Table

1. Navigate to the Amazon DynamoDB Tables Console at https://console.aws.amazon.com/dynamodb/home?region=us-east-1#tables.
2. Click the **Create table** button.
3. For the **Table name**, enter `SampleSmartHome`.
4. For the **Primary key**, enter `ItemId`.
5. Click the **Create** button to create the table.

> It may take a moment for the table to be created.

Deploy the Sample Code

To deploy the sample code, it must first be packaged and then uploaded to the AWS Lambda as the function code.

Package the Sample Code

1. In your working directory, browse to the *cisco-skill/lambda/smarthome* directory.
2. In that directory and from the command-line, run the following command to install the dependencies: `npm install`.
3. Zip the contents of the *cisco-skill/lambda/smarthome* directory into a file named `package.zip` and save it to the `working-cisco-skill` folder. Make sure that the index.js file, alexa/ folder, and node_modules/ folder are at the root of the zip file. Do not include the full `cisco-skill/lambda/smarthome` path structure.

If you encounter any issues, refer to the AWS Documentation for Creating a Deployment Package (Node.js).

Upload the Sample Code

1. Navigate to https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/cisco-skill?tab=graph and locate the *Function code* section.
2. In the code entry type dropdown, select **Upload a .ZIP file**.
3. Click the **Upload** button and browser to the `working-cisco-skill` directory or where you zipped the packaged code.
4. Select the `package.zip` file to upload as the lambda function code.
5. Click **Save** at the top right of the page. If successful, the environment of the *Function code* section will update replacing the previous contents.

Test the Lambda Function

After the sample code is uploaded to the Lambda function, you can test that it is properly working via the **test events** functionality of the AWS Lambda console. This is done by creating a Test Event that contains a sample directive message that the function code should handle and respond to.

Create a Discovery Test Event

1. On the function page for *cisco-skill*, select the **Select a test event..** dropdown from the top menu of the function and click **Configure test events**.
2. In the dialog that opens, leave **Create new test event** selected and leave the default template.
3. For the *Event name* enter: `directiveDiscovery`.
4. Copy and paste the contents of https://raw.githubusercontent.com/alexa/alexa-smarthome/master/sample_messages/Discovery/Discovery.request.json into the text field at the bottom of the dialog replacing its contents.
5. Click **Create**.

Run a Discovery Test Event

1. With the **directiveDiscovery** Test Event created and selected in the dropdown, click the **Test** button.
2. If everything was successful, an "Execution result: succeeded" message will be returned and clicking on the **Details** link of the result should show a response that looks like the following:

Any errors with the code or packaging and deployment of the code will be reflected in the result response. Not that on a successful response, the event namespace is `Alexa.Discovery`. This is because the code is responding to a test event that sends in a Discovery directive.

Test the Skill

To test the smart home skill on a device, it must first be enabled and linked to your Amazon developer account.

Link the Alexa Smart Home Skill

1. Go to https://alexa.amazon.com, login with your developer account, and select **Skills** from the left menu.
2. Click **Your Skills** from the top right of the section.
3. On the Your Skills page, select the *DEV SKILLS* tab.
4. Click the *cisco-skill* skill.
5. On the *cisco-skill* page, click **Enable** in the top right and authenticate with your Cisco account. If you are already signed in, you will be presented with a dialog asking to allow the account linking".
6. On success, you should be presented with a window that reads "cisco-skill" has been successfully linked.
7. When redirected back to the Skill page, you will be prompted *Discover Devices*. Click **Discover Devices**.
8. After the discovery process completes, a new "Sample Switch" with the description of "Sample Endpoint Description" will be available in your *Devices* list.

Test the Skill through the Alexa Developer Console

The Alexa Developer Console has test functionality built-in and can be used to test your skills under development.

1. Navigate to https://developer.amazon.com/alexa/console/ask and select cisco-skill from the list of *Alexa Skills*.
2. Select the *Test* tab.
3. In the *Alexa Simulator* tab, leave English selected and type `Alexa, turn on cisco room kit` into the input box.
4. If successful, Alexa should respond with `ok`.

Test the Skill with an Alexa-Enabled Device

If you have an Amazon Echo or other Alexa-enabled device associated to your Amazon developer account, you can give that device the command “Alexa, turn on sample switch” and she should respond with “OK”.

Review the Skill Logs

Interactions between Alexa and the AWS Lambda function are recorded in AWS CloudWatch. Those logs are useful for investigating what information is sent back and forth to the Lambda function.

Review the logs on CloudWatch

1. Browse to https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logStream:group=/aws/lambda/cisco-skill
2. Click the latest **Log Stream** and review the *Messages*.
3. Note the values recorded in the logs. The inbound handler **request** and **context** values are from Alexa. The outbound handler **response** value is from the Lambda function.

Congratulations, if you have made it this far you have a working sample!