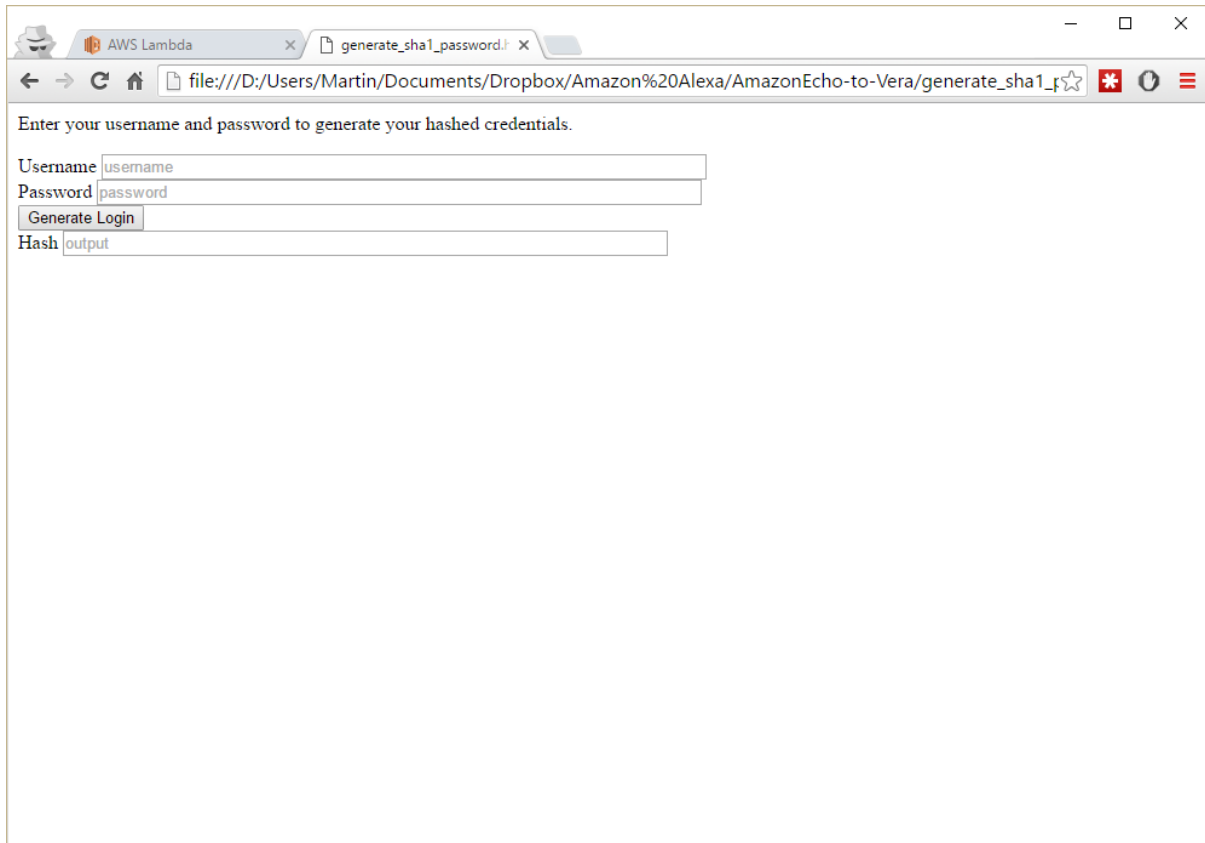


## Vera Credentials

First of all you need to get your vera remote access credentials. Download

[https://raw.githubusercontent.com/mmillmor/AmazonEcho-to-](https://raw.githubusercontent.com/mmillmor/AmazonEcho-to-Vera/master/generate_sha1_password.html)

[Vera/master/generate\\_sha1\\_password.html](https://raw.githubusercontent.com/mmillmor/AmazonEcho-to-Vera/master/generate_sha1_password.html) to your computer and open it with a web browser



The screenshot shows a web browser window with a single tab titled 'generate\_sha1\_password.html'. The address bar shows a file path: 'file:///D:/Users/Martin/Documents/Dropbox/Amazon%20Alexa/AmazonEcho-to-Vera/generate\_sha1\_password.html'. The page content includes a heading 'Enter your username and password to generate your hashed credentials.' followed by four input fields: 'Username' (containing 'username'), 'Password' (containing 'password'), 'Generate Login' (a button), and 'Hash' (containing 'output').

Enter your username and password, and press Generate Login. That will calculate your hashed credentials, which you will need later on. These are the same credentials that you would use at [home.getvera.com](https://home.getvera.com)

## Set up the remote code

1) Sign up for a free AWS account at

[https://portal.aws.amazon.com/gp/aws/developer/registration/index.html?nc2=h\\_ct](https://portal.aws.amazon.com/gp/aws/developer/registration/index.html?nc2=h_ct)



## Sign In or Create an AWS Account

What is your e-mail or mobile number?

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?](#)

### New AWS Accounts Include:

#### 12 months of access to the AWS Free Tier

Amazon EC2: 750 hrs/month of Windows and Linux t2.micro instance usage  
Amazon S3: 5GBs of Storage  
Amazon RDS: 750 hrs/month of Micro DB Instance usage  
Amazon DynamoDB: 25 GB of storage, up to 200 million requests/month

#### AWS Basic Support Features

Customer Service: 24x7x365  
Support Forums  
Documentation, White Papers, and Best Practice Guides

Visit [aws.amazon.com/free](https://aws.amazon.com/free) for full offer terms.

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.



## Login Credentials

Use the form below to create login credentials that can be used for AWS as well as Amazon.com.

My name is:

My e-mail address is:

Type it again:

note: this is the e-mail address that we will use to contact you about your account

Enter a new password:

Type it again:

Create account

### About Amazon.com Sign In

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services. Your use of this site is governed by our [Terms of Use](#) and [Privacy Policy](#) linked below.

Enter your name, email and password

## Contact Information

☐ Company Account ☒ Personal Account

\* Required Fields

Full Name\*

Country\*

Address\*


Apartment, suite, unit, building, floor, etc.

City\*

State / Province or Region\*

Postal Code\*

Phone Number\*

Security Check 

Enter contact details

Contact Information

Payment Information

Identity Verification

Support Plan

Confirmation

## Payment Information

Please enter your payment information below. You will be able to try a broad set of AWS products for free via the Free Usage Tier. We will only bill your credit or debit card for usage that is not covered by our Free Usage Tier.

AWS Free Usage Tier	Compute Amazon EC2	Storage Amazon S3	Database Amazon RDS
free for 1 year	750hrs/month*	5GB	750hrs/month*

[\\*View full offer details »](#)


Credit/Debit Card Number


Expiration Date

01 ▼ 2016 ▼


Cardholder's Name


Enter a credit card number. Note, you will never be billed for the usage describe here - the Lambda service is free under 1 million hits a month (<https://aws.amazon.com/lambda/pricing/>)





English  [Sign Out](#)


Amazon Web Services Sign Up

 Contact Information

 Payment Information

 Identity Verification

 Support Plan

 Confirmation

## Identity Verification

You will be called immediately by an automated system and prompted to enter the PIN number provided.


### 1. Provide a telephone number


Please enter your information below and click the "Call Me Now" button.


Country Code

Phone Number

Ext

United Kingdom (+44) 







Call Me Now


2. Call in progress


Amazon will verify your identity





English  [Sign Out](#)


Amazon Web Services Sign Up

 Contact Information

 Payment Information

 Identity Verification


 Support Plan


 Confirmation



## Support Plan

All customers receive free support. Choosing a paid support plan will allow you to receive one-on-one technical assistance from experienced engineers and access many other support features. Please see below.

Please Select One

 **Basic (Free)**  
Contact Customer Service for account and billing questions, receive help for resources that don't pass system health checks, and access the AWS Community Forums.

 **Developer (\$49/month)**  
Get started on AWS - ask technical questions and get a response to your web case within 12 hours during local business hours.

 **Business (Starting at \$100/month - [Pricing Example](#)) - Recommended**  
24/7/365 real-time assistance by phone and chat, a 1 hour response to web cases, and help with 3rd party software. Access AWS Trusted Advisor to increase performance, fault tolerance, security, and potentially save money. 

Pick the free support plan

Menu



Products

More ▾

English ▾

My Account ▾

Complete Sign Up

## Welcome to Amazon Web Services

Thank you for creating an Amazon Web Services (AWS) Account. We are in the process of activating your account. For most customers, activation only takes a couple minutes, but it can sometimes take a few hours if additional account verification is required. We will notify you by email when your account is activated.

Sign In to the Console

Contact Sales

### Get Started Quickly



Launch a Linux Virtual Machine



Backup Files



Launch a WordPress Website



Deploy Web Apps

And that's your account created. Sign in with the account you just created



## Sign In or Create an AWS Account

What is your e-mail or mobile number?

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?](#)

Amazon

Aurora

Enterprise-class database at 1/10th the cost.



"10 times faster than our MySQL environment. It just works!"

-Alfresco

Learn more

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.



AWS

Services

Edit

Martin

Oregon

Support

# Amazon Web Services

Compute

**EC2**  
Virtual Servers in the Cloud

**EC2 Container Service**  
Run and Manage Docker Containers

**Elastic Beanstalk**  
Run and Manage Web Apps

**Lambda**  
Run Code in Response to Events

Storage & Content Delivery

**S3**  
Scalable Storage in the Cloud

**CloudFront**  
Global Content Delivery Network

**Elastic File System** PREVIEW  
Fully Managed File System for EC2

**Glacier**  
Archive Storage in the Cloud

**Import/Export Snowball**  
Large Scale Data Transport

**Storage Gateway**  
Hybrid Storage Integration

Database

**RDS**  
Managed Relational Database Service

**DynamoDB**

Developer Tools

**CodeCommit**  
Store Code in Private Git Repositories

**CodeDeploy**  
Automate Code Deployments

**CodePipeline**  
Release Software using Continuous Delivery

Management Tools

**CloudWatch**  
Monitor Resources and Applications

**CloudFormation**  
Create and Manage Resources with Templates

**CloudTrail**  
Track User Activity and API Usage

**Config**  
Track Resource Inventory and Changes

**OpsWorks**  
Automate Operations with Chef

**Service Catalog**  
Create and Use Standardized Products

**Trusted Advisor**  
Optimize Performance and Security

Security & Identity

Internet of Things

**AWS IoT**  
Connect Devices to the Cloud

Mobile Services

**Mobile Hub** BETA  
Build, Test, and Monitor Mobile apps

**Cognito**  
User Identity and App Data Synchronization

**Device Farm**  
Test Android, FireOS, and iOS Apps on Real Devices in the Cloud

**Mobile Analytics**  
Collect, View and Export App Analytics

**SNS**  
Push Notification Service

Application Services

**API Gateway**  
Build, Deploy and Manage APIs

**AppStream**  
Low Latency Application Streaming

**CloudSearch**  
Managed Search Service

**Elastic Transcoder**  
Easy-to-Use Scalable Media Transcoding

**SES**

Resource Groups

Create a Group

Tag Editor

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

Additional Resources

Getting Started

Read our [documentation](#) or view our [training](#) to learn more about AWS.

AWS Console Mobile App

View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

AWS Marketplace

Find and buy software, launch with 1-Click and pay by the hour.

AWS re:Invent Announcements

Explore the next generation of AWS cloud capabilities. [See what's new](#)

AWS

Services

Edit

Martin

N. Virginia

Support

Lambda > New function

Step 1: Select blueprint

Select blueprint

Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under [CC0](#).

Filter

All languages

<< < Viewing 1-9 of 35 > >>

s3-get-object-python

An Amazon S3 trigger that retrieves metadata for the object that has been updated.

python2.7 · s3

config-rule-change-triggered

An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types.

nodejs · config

dynamodb-process-stream

An Amazon DynamoDB trigger that logs the updates made to a table.

nodejs · dynamodb

microservice-http-endpoint

A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway.

node-exec

Demonstrates running an external process using the Node.js child\_process module.

slack-echo-command-python

A function that handles a Slack slash command and echoes the details back to the user.

Enter "alexa-connected-home-driver" in the filter box, and click on that box

AWS

Services

Edit

Martin

N. Virginia

Support

Lambda > New function

Step 1: Select blueprint

Select blueprint

Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under [CC0](#).

alexa-connected-home-driver

All languages

<< < Viewing 1-1 of 1 > >>

alexa-connected-home-driver

Demonstrates the control of supported Alexa Connected Home products

nodejs · iot · connected-home · alexa · light

Cancel

Skip

Feedback

English

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

The screenshot shows the AWS Lambda console interface for creating a new function using the 'alex-connected-home-driver' blueprint. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Edit' dropdown, and user/region information (Martin, N. Virginia, Support). The breadcrumb trail is 'Lambda > New function using blueprint alex-connected-home-driver'. The progress bar shows four steps: 'Step 1: Select blueprint', 'Step 2: Configure event sources' (highlighted), 'Step 3: Configure function', and 'Step 4: Review'. The main heading is 'Configure event sources'. Below it, a text prompt says 'Choose the appropriate event source for your Lambda function.' There is a dropdown menu for 'Event source type' with 'Alexa Connected Home' selected. An information icon is next to the dropdown. Below the dropdown, a paragraph explains that choosing 'Submit' will create a resource policy for the Amazon Alexa service and provides a link to the 'Alexa Developer' portal. At the bottom right, there are four buttons: 'Cancel', 'Previous', 'Skip', and 'Next'.

Leave the event source as Alexa Connected Home, and press Next

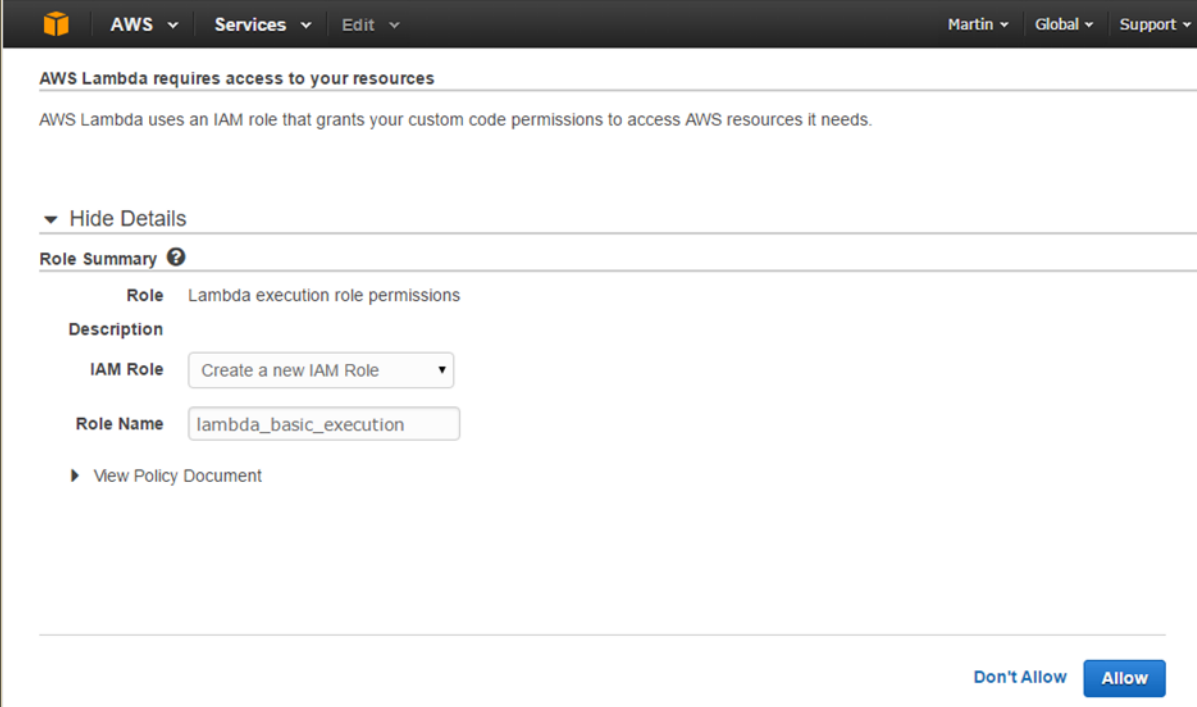
The screenshot shows the AWS Lambda console interface for the 'Configure function' step. The top navigation bar is the same as the previous screenshot. The breadcrumb trail is 'Lambda > New function using blueprint alex-connected-home-driver'. The progress bar shows four steps: 'Step 1: Select blueprint', 'Step 2: Configure event sources', 'Step 3: Configure function' (highlighted), and 'Step 4: Review'. The main heading is 'Configure function'. Below it, a text prompt says 'A Lambda function consists of the custom code you want to execute. Learn more about Lambda functions.' There are three input fields: 'Name\*' with 'myVeraIntegration', 'Description' with 'Integrates the Echo to my Vera', and 'Runtime\*' with 'Node.js' selected in a dropdown. Below these fields is the 'Lambda function code' section. It contains a text prompt: 'Provide the code for your function. Use the editor if your code does not require custom libraries (other than the aws-sdk). If you need custom libraries, you can upload your code and libraries as a .ZIP file. Learn more about deploying Lambda functions.' There are three radio buttons for 'Code entry type': 'Edit code inline' (selected), 'Upload a .ZIP file', and 'Upload a .ZIP from Amazon S3'. At the bottom, there is a code editor area with a line number '1' and a file name '/\*\*'.

Enter any name and description for your code. Leave the runtime as Node.js. Scroll down to the code entry. Copy and paste the entire contents of [https://raw.githubusercontent.com/mmillmor/AmazonEcho-to-Vera/master/lambda/alexa\\_lambda\\_amazon\\_oauth.js](https://raw.githubusercontent.com/mmillmor/AmazonEcho-to-Vera/master/lambda/alexa_lambda_amazon_oauth.js) in there.



Scroll down and replace the text {enter your username} with your vera username, and the text {enter your encoded password} with the hashed vera password from the first step.

Scroll down to the bottom of the page. For Role, pick Basic Execution Role\* and it will open a new window to create a role;



AWS Lambda requires access to your resources

AWS Lambda uses an IAM role that grants your custom code permissions to access AWS resources it needs.

▼ Hide Details

Role Summary ?

Role Lambda execution role permissions

Description

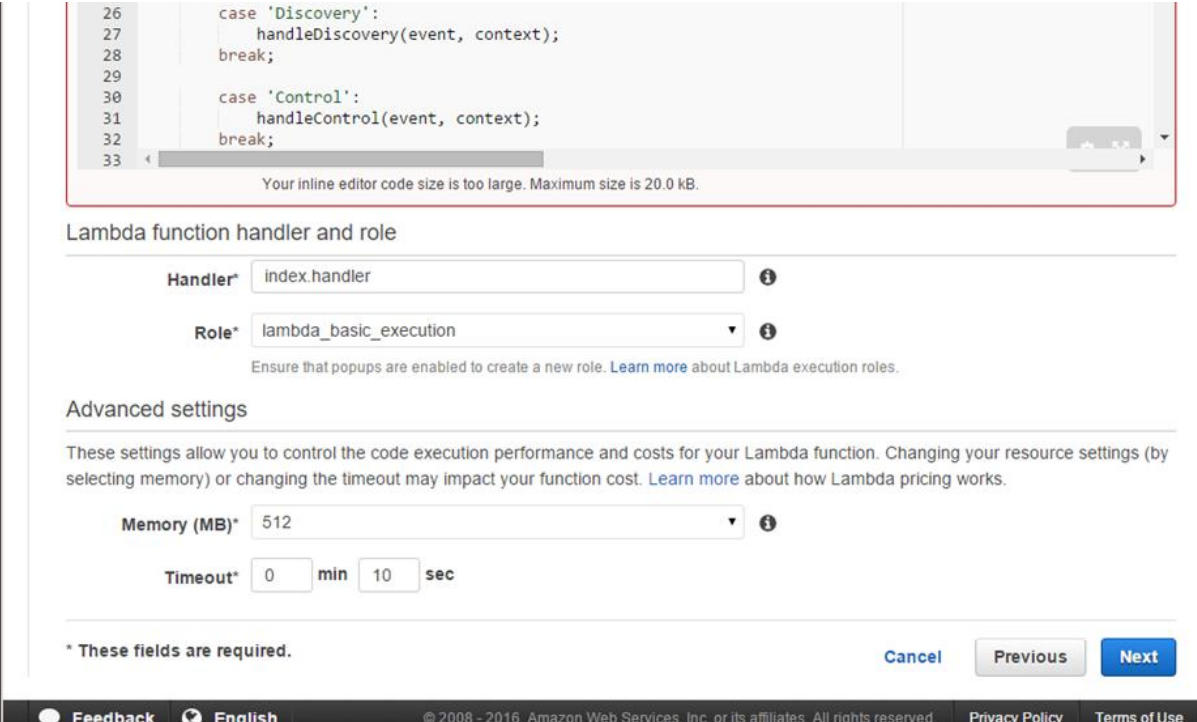
IAM Role Create a new IAM Role ▼

Role Name lambda\_basic\_execution

► View Policy Document

Don't Allow Allow

name the role lambda\_basic\_execution, and click Allow. This will go back to the previous page with the role populated. Leave everything else at the default values.



```
26     case 'Discovery':
27         handleDiscovery(event, context);
28         break;
29
30     case 'Control':
31         handleControl(event, context);
32         break;
33
```

Your inline editor code size is too large. Maximum size is 20.0 kB.

Lambda function handler and role

Handler\* index.handler ⓘ

Role\* lambda\_basic\_execution ⓘ

Ensure that popups are enabled to create a new role. [Learn more](#) about Lambda execution roles.

Advanced settings

These settings allow you to control the code execution performance and costs for your Lambda function. Changing your resource settings (by selecting memory) or changing the timeout may impact your function cost. [Learn more](#) about how Lambda pricing works.

Memory (MB)\* 512 ⓘ

Timeout\* 0 min 10 sec

\* These fields are required.

Cancel Previous Next

If it complains about the file being too big, remove the copyright text at the top of the page (everything between `/**` and `*/`)

Click Next

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with the AWS logo, 'AWS', 'Services', and 'Edit' dropdowns. On the right, it shows the user 'Martin', the region 'N. Virginia', and a 'Support' link. Below the navigation bar, the breadcrumb is 'Lambda > New function using blueprint alexa-connected-home-driver'. A list of steps is shown: 'Step 1: Select blueprint', 'Step 2: Configure event sources', 'Step 3: Configure function', and 'Step 4: Review' (which is highlighted with an orange bar). The 'Review' section has a title 'Review' and a paragraph: 'Please review your Lambda function details. You can go back to edit changes for each section. When you are ready, click **Create function** to complete the setup process.' There are two main sections for review: 'Event sources' and 'Lambda function'. The 'Event sources' section shows 'Alexa Connected' and 'Home' as event sources, with an 'Edit' button. The 'Lambda function' section shows 'Name' as 'myVeraIntegration', 'Description' as 'Integrates the Echo to my Vera', and 'Runtime' as 'NodeJS', also with an 'Edit' button.

Scroll down and click Create Function.

Click the Code tab, then the Test button, and paste the following in to the text box;

```
{
  "header": {
    "namespace": "Discovery",
    "name": "DiscoverAppliancesRequest",
    "payloadVersion": "1"
  },
  "payload": {
    "accessToken": "dummy"
  }
}
```

Scroll down, and you should see a list of your devices;

```
198
199 }
200
201 /**
202  * Control events are processed here.
203  * This is called when Alexa requests an action (IE turn off appliance).
204  */
205 function handleControl(event, context) {
206
```

✓ Execution result: succeeded (logs)

The area below shows the result returned by your function execution using the context methods. [Learn more](#) about returning results from your function.

```
{
  "header": {
    "namespace": "Discovery",
    "name": "DiscoverAppliancesResponse",
    "payloadVersion": "1"
  },
  "payload": {
    "discoveredAppliances": [
      {
        "manufacturerName": "vera",
        "modelName": "vera scene",
        "version": "1"
      }
    ]
  }
}
```

Summary

At the top of the page you will see the text "ARN - " followed by a string like `arn:aws:lambda:us-east-1:XXX`. You will need that string later.

AWS Services Edit Martin N. Virginia Support

Lambda > Functions > myVeralIntegration ARN - arn:aws:lambda:us-east-1:73322796138:function:myVeralIntegration

Test Actions

Code Configuration Event sources API endpoints Monitoring

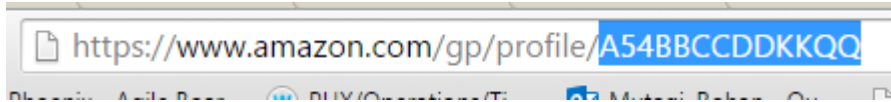
Code entry type ☒ Edit code inline ☐ Upload a .ZIP file ☐ Upload a .ZIP from Amazon S3

```
177 context.succeed(result),
178
179 });
180 });
181 } else {
182
183
184   var payload = {
185     exception: {
186       code: 'INVALID_ACCESS_TOKEN',
187       description: 'Could not find user'
188     }
189   };
190
191   var result = {
192     header: headers,
193     payload: payload
194   };
195   context.fail(result);
196 }
197 // }); // getLoginDetails
198
```

Well done - you are almost done!

## Amazon Account Details

Go to <https://amazon.com/profile> while logged in with the account that your Echo is connected to. The URL will change to have your customer ID in the URL. You will need that later



## Contact Amazon

The final step is to contact Amazon with your information so that they can wire it all together. Send an e-mail to [alexa-coho-submissions@amazon.com](mailto:alexa-coho-submissions@amazon.com) with the following details. Note the use of the milliesoft OAuth server. This is a dummy OAuth server that you need to get your Echo to think it has authenticated with the remote service (although in reality the authentication happens in the Lambda code). It does not store any details about you, or allow any access to your Vera or your Echo.

<i>Skill adapter display name</i>	<i>The name for your integration, e.g. "Martin's Vera System"</i>
<i>Skill adapter description</i>	<i>A description, e.g. "Integration with my Vera"</i>
<i>AWS Lambda function name</i>	<i>This is the ARN string from the end of the coding step</i>
<i>OAuth Client ID</i>	echo_vera
<i>OAuth Client Secret</i>	fNxzmWoy0fjb
<i>OAuth Scope</i>	profile
<i>OAuth authorization URL</i>	<a href="https://www.milliesoft.co.uk/auth/authorize.php">https://www.milliesoft.co.uk/auth/authorize.php</a>
<i>OAuth token URL</i>	<a href="https://www.milliesoft.co.uk/auth/token.php">https://www.milliesoft.co.uk/auth/token.php</a>
<i>Amazon Customer ID</i>	<i>This is the customer ID from the last step above</i>

Amazon will respond with confirmation that it has all been set up, and then you can run discover and control devices on your Echo.