



lab title

Developing Serverless JavaScript DynamoDB Applications

V1.01



Course title

AWS Certified Associate



Table of Contents

Contents

Table of Contents.....	1
About the Lab	1
Registering an Amazon Developer App	1
Creating a DynamoDB Table	1
Creating an IAM Role for Federated Identity	1
Creating a Federated Identity Application.....	1
Querying DynamoDB Tables using the NodeJS SDK	Error! Bookmark not defined.



About the Lab

These lab notes are to support the instructional videos on Accessing AWS with Web Identity Federation in the BackSpace AWS Certified Developer course.

We will first create a DynamoDB table then create an IAM Role for Federated Identity access to the table. We will then create a Javascript application that will use the login with Amazon SDK to authenticate users against their Amazon account.

Please refer to the Login with Amazon SDK documentation at:

https://images-na.ssl-images-amazon.com/images/G/01/lwa/dev/docs/website-developer-guide_TTH_.pdf

Please refer to the AWS JavaScript SDK documentation at:

<http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/DynamoDB.html>

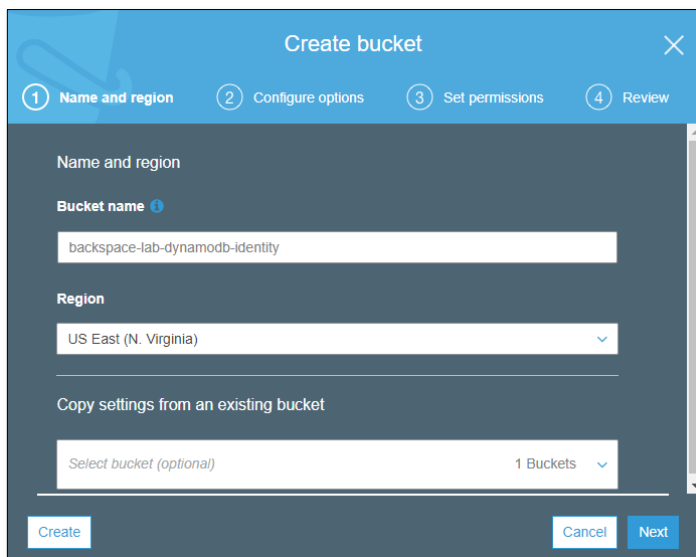
Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the latest version with any updates or corrections.

🎬 Upload Files to an S3 Bucket

In this section we will use the Amazon S3 service to upload our website files to a bucket.

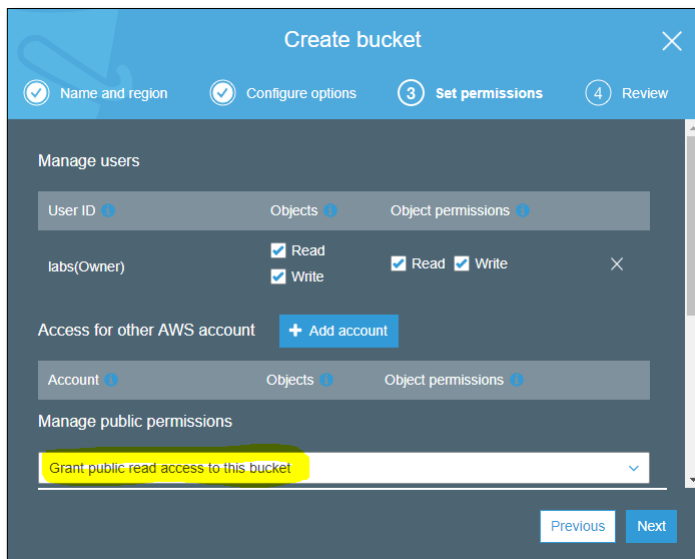
Go to the S3 management console.

Create a bucket with a unique name

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The window has a blue header with the title 'Create bucket' and a close button. Below the header is a progress bar with four steps: 1. Name and region, 2. Configure options, 3. Set permissions, and 4. Review. The first step, 'Name and region', is active. It contains three input fields: 'Bucket name' with the value 'backspace-lab-dynamodb-identity', 'Region' with a dropdown menu showing 'US East (N. Virginia)', and 'Copy settings from an existing bucket' with a dropdown menu showing 'Select bucket (optional)'. At the bottom of the form are three buttons: 'Create', 'Cancel', and 'Next'.

Click *Next*

Select *Grant public read access to this bucket*



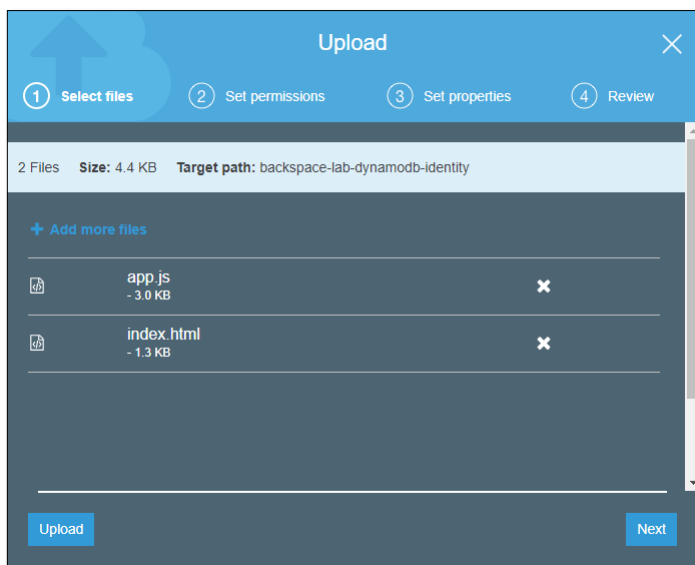
Click *Create bucket*

Download the following files:

<https://raw.githubusercontent.com/backspace-academy/aws-dynamodb-login-with-amazon/master/app.js>

<https://raw.githubusercontent.com/backspace-academy/aws-dynamodb-login-with-amazon/master/index.html>

Upload the files to the bucket



Make sure you select public permissions *Grant public read access to this object(s)*

Upload

Select files

2 Set permissions

3 Set properties

4 Review

labs(Owner)

☒ Read

☒ Write

☒ Read

☒ Write

X

Access for other AWS account

+ Add account

Account

Objects

Object permissions

Manage public permissions

Grant public read access to this object(s)

This object(s) has public read access.

Everyone in the world will have read access to this object(s).

Upload

Previous

Next

▶ Creating a CloudFront Distribution

In this section we will use the Amazon CloudFront service to cache our website and enable https.

Go to the CloudFront console

Click *Create Distribution*

Select *Web*

Select the bucket you created for *Origin Domain Name*

Create Distribution

Origin Settings

Origin Domain Name

Origin Path

Origin ID

Origin Custom Headers

Select *Redirect HTTP to HTTPS*

Default Cache Behavior Settings

Path Pattern

Viewer Protocol Policy

Select *Default CloudFront Certificate (*.cloudfront.net)*

Distribution Settings

Price Class

AWS WAF Web ACL

Alternate Domain Names (CNAMEs)

SSL Certificate

Enter index.html for *Default Root Object*

Click *Create Distribution*

Default Root Object ⓘ

Logging ☐ On ☒ Off ⓘ

Bucket for Logs ⓘ

Log Prefix ⓘ

Cookie Logging ☐ On ☒ Off ⓘ

Enable IPv6 ☒ ⓘ
[Learn more](#)

Comment ⓘ

Distribution State ☒ Enabled ☐ Disabled ⓘ

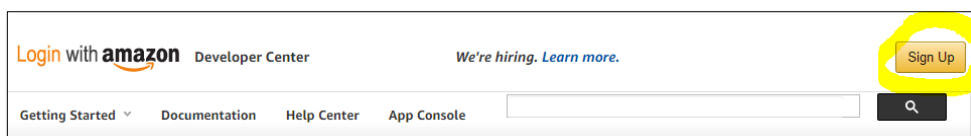
[Cancel](#) [Back](#) [Create Distribution](#)

▶ Registering an Amazon Developer App

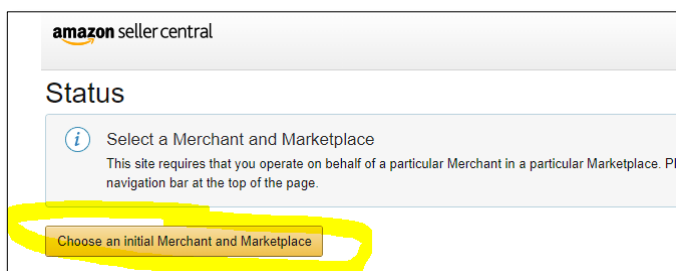
In this section we will use the Amazon Developer Portal to register a Login with Amazon app.

Go to <https://login.amazon.com>

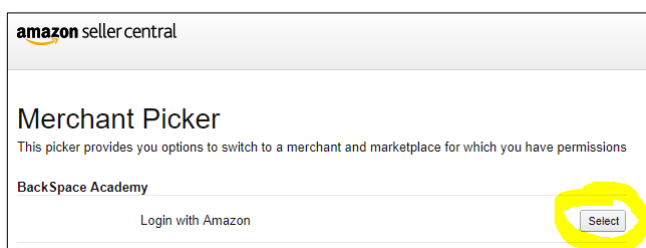
Use your Amazon login details to register as an Amazon developer



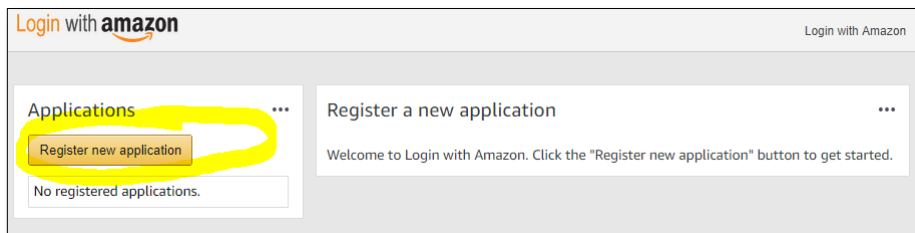
You will be redirected to Amazon Seller Central. Click *Choose an initial Merchant and Marketplace*



Click *Select*



You will be redirected to Login with Amazon. Click *Register new application*



Give your app a name and description

If you don't have a privacy policy you can use a sample at <http://cdn.backspace.academy/courses/aws-certification/03/075/privacy-policy-template.pdf>

You can optionally upload an image if you wish

Click **Save**

Select **Web Settings**

Click **Edit**

Go to the CloudFront management console and copy the domain of your distribution.

CloudFront Distributions					
Create Distribution Distribution Settings Delete Enable Disable					
Viewing: Any Delivery Method Any State 					
Delivery Method	ID	Domain Name	Comment	Origin	
Web	EXOOEH5WX4D3X	d2p8ito5zl9htl.cloudfront.net	-	backspace-lab-dynar	

Add the https domain name **with https://** of your CloudFront distribution for *Allowed JavaScript Origins*

Click Save

Web Settings

Client ID: amzn1.application-oa2-client.2fc4d46a16704a0c887e02cb41ec8711
 Client Secret: [Show Secret](#)

Allowed JavaScript Origins:
 (Optional) [Add Another](#)

Allowed Return URLs:
 (Optional) [Add Another](#)

[Cancel](#) [Save](#)

Your app is now registered

Applications

[Register new application](#)

DynamoDB Test App

DynamoDB Test App

Application ID: amzn1.application.2408a8dd425c4497848a22840a191939

[Settings](#) [Metrics](#)

Application Information

Name: DynamoDB Test App
 Description: Test app for DynamoDB Federated access
 Privacy Notice URL: <http://cdn.backspace.academy/courses/aws-certification/03/075/privacy-policy-template.pdf>
 Logo Image: (Optional) [Edit](#)

Web Settings

Client ID: amzn1.application-oa2-client.2fc4d46a16704a0c887e02cb41ec8711
 Client Secret: [Show Secret](#)
 Allowed JavaScript Origins:
 (Optional)
 Allowed Return URLs:
 (Optional) [Edit](#)

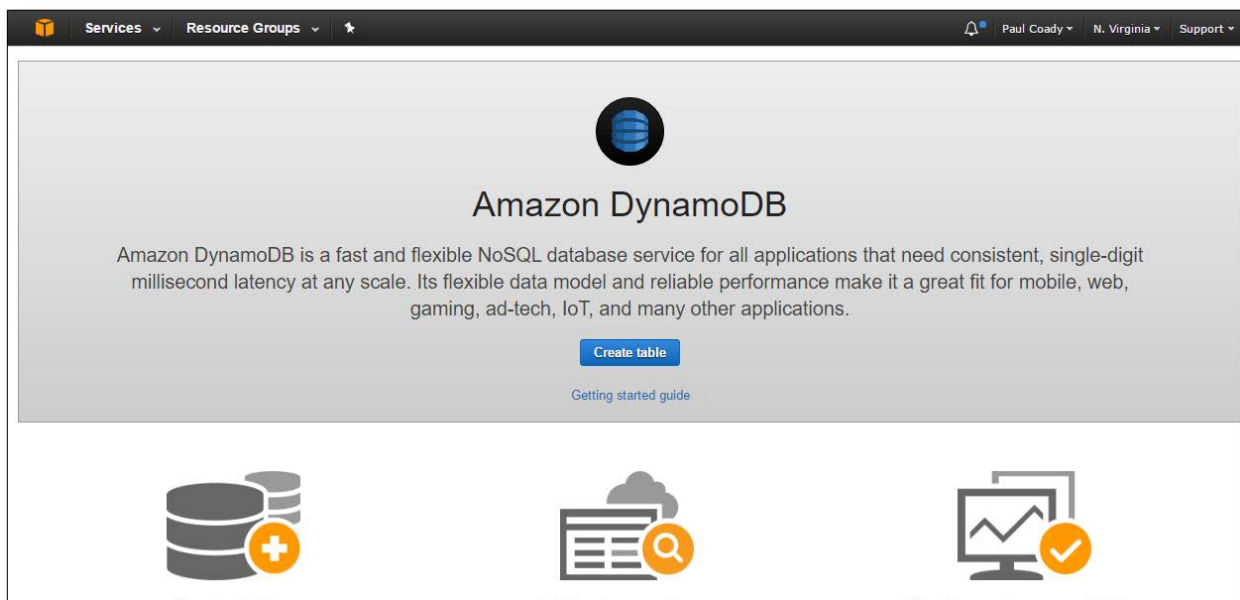
Android Settings

iOS Settings

▶ Creating a DynamoDB Table

In this section we will use the DynamoDB console to create a table.

Select the DynamoDB Console



Click "Create Table"

Enter the following details (enter exactly with correct case)

BE CAREFUL IF USING COPY/PASTE NOT TO INCLUDE ANY EXTRA SPACES ON THE END.

Table Name: login-with-amazon-test

Primary key Partition key: Customer (case sensitive)

Attribute Type: String

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key

ⓘ

☐ Add sort key

Uncheck *Use Default Settings*

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

Disable auto scaling

Auto Scaling

☐ Read capacity ☐ Write capacity

Change provisioned capacity to 1.

Provisioned capacity

Table

Read capacity units Write capacity units

Estimated cost \$0.59 / month ([Capacity calculator](#))

Leave encryption disabled.

Click Create.

Encryption At Rest

You may enable encryption for your DynamoDB table to help protect data at rest. [Learn more](#)

☐ Enable encryption

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel Create

Go to the *Tables* tab

Press refresh until table status is listed as active.

DynamoDB

Dashboard
Tables
Backups
Reserved capacity
Preferences [Preview](#)

[Create table](#) [Delete table](#)

Filter by table name

Viewing 1 of 1 Tables

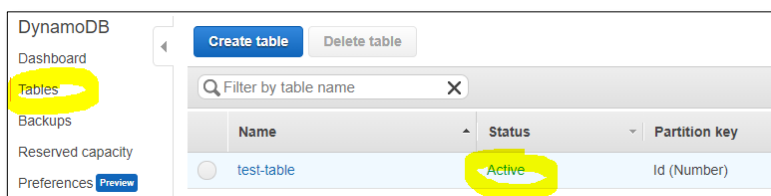
Name	Status	Partition key	Sort key
<input type="radio"/> login-with-amazon-test	Active	Customer (String)	-

▶ Creating an IAM Role for Federated Identity

In this section we will use the **DynamoDB console** to create an IAM policy to allow users to access DynamoDB with Login with Amazon. We will then use this policy with an IAM role using the IAM console.

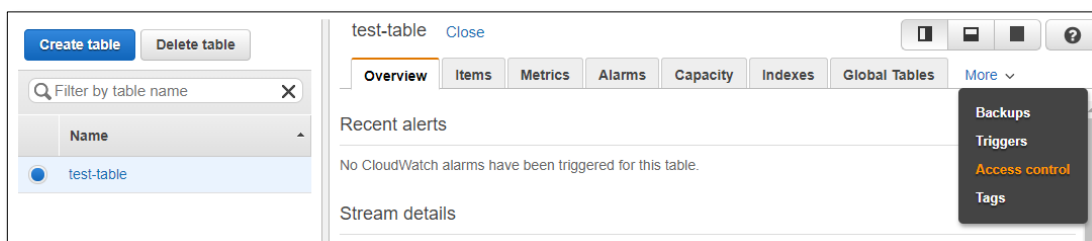
From the DynamoDB console select *Tables*

Wait until your table status is active



Select the table you created

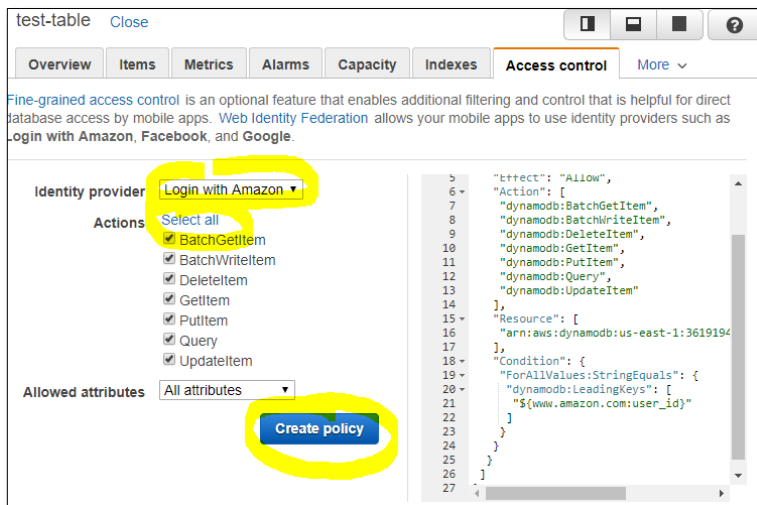
Select *Access control*



Select Login with Amazon

Select All Actions

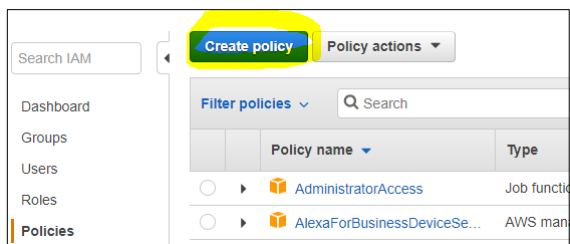
Click *Create policy*



Open the IAM console in another browser tab

Select *Policies*

Click *Create policy*



Select the *JSON* tab

Copy and paste the policy you created previously in the DynamoDB console



Click *Review policy*


```

15  "Resource": [
16    "arn:aws:dynamodb:us-east-1:361919435810:table/test-table"
17  ],
18  "Condition": {
19    "ForAllValues:StringEquals": {
20      "dynamodb:LeadingKeys": [
21        "${www.amazon.com:user_id}"
22      ]
23    }
24  }
25 }
26 }
27 }

```

Cancel **Review policy**

Give the policy a name and description

Review policy

Name*
Use alphanumeric and '+=, @, -, _' characters. Maximum 128 characters.

Description
Maximum 1000 characters. Use alphanumeric and '+=, @, -, _' characters.

Click *Create policy*

Summary

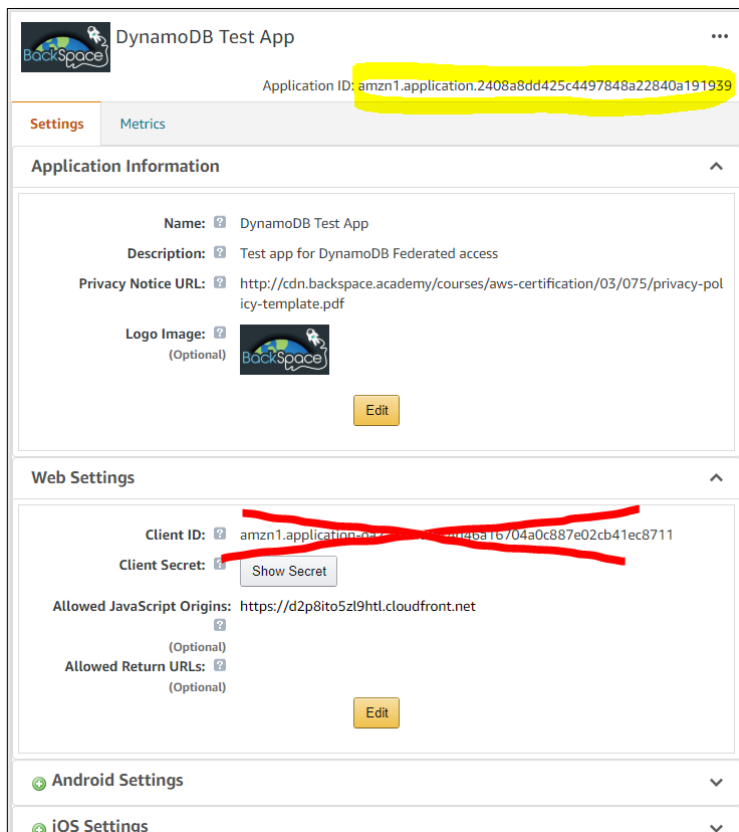
Service	Access level	Resource	Request condition
Allow (1 of 144 services) Show remaining 143			
DynamoDB	Limited: Read, Write	TableName string like login-with-amazon-test	dynamodb:LeadingKeys string like login-with-amazon-test

Cancel Previous **Create policy**

Go to <https://sellercentral.amazon.com>

Copy the *Application ID* of your app

PLEASE NOTE COPY THE APPLICATION ID NOT THE CLIENT ID



DynamoDB Test App ...

Application ID: **amzn1.application.2408a8dd425c4497848a22840a191939**


Settings Metrics

Application Information ^

Name: **DynamoDB Test App**

Description: **Test app for DynamoDB Federated access**

Privacy Notice URL: **http://cdn.backspace.academy/courses/aws-certification/03/075/privacy-policy-template.pdf**

Logo Image: **(Optional)** 

Edit

Web Settings ^

Client ID: **amzn1.application.2408a8dd425c4497848a22840a191939**

Client Secret: **amzn1.application.2408a8dd425c4497848a22840a191939** **Show Secret**

Allowed JavaScript Origins: **https://d2p8ito5z19htl.cloudfront.net**

(Optional)

Allowed Return URLs: **(Optional)**

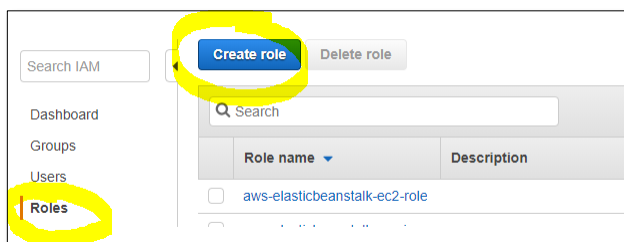
Edit

Android Settings v

iOS Settings v

Go to **Roles**

Click **Create role**



Search IAM

Create role Delete role

Dashboard

Groups

Users

Roles

Search

Role name	Description
<input type="checkbox"/> aws-elasticbeanstalk-ec2-role	


Select **Web identity**


Select **Login with Amazon**


Enter the Login with Amazon Application ID for **Application ID**


Click **Next: Permissions**

Select type of trusted entity

 **AWS service**
EC2, Lambda and others

 **Another AWS account**
Belonging to you or 3rd party

 **Web identity**
Cognito or any OpenID provider

 **SAML 2.0 federation**
Your corporate directory

Allows users federated by the specified external web identity or OpenID Connect (OIDC) provider to assume this role to perform actions in your account.
[Learn more](#)

Choose a web identity provider

Identity provider [Create new provider](#) [Refresh](#)

Application ID*

Condition [+ Add condition \(optional\)](#)

* Required

[Cancel](#) [Next: Permissions](#)

Search for the policy you created

Select the policy

Click *Next: Review*

Create role

1 2 3

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#) [Refresh](#)

Filter policies Showing 1 result

	Policy name ▼	Used as	Description
<input checked="" type="checkbox"/>	login-with-amazon-dynamodb	None	Login with Amazon Federated Identity for...

* Required

[Cancel](#) [Previous](#) [Next: Review](#)

Give the role a name and description

Click *Create role*

Review

Provide the required information below and review this role before you create it.

Role name*
Use alphanumeric and '+=, @-.' characters. Maximum 64 characters.

Role description
Maximum 1000 characters. Use alphanumeric and '+=, @-.' characters.

Trusted entities The identity provider `www.amazon.com:app_id`

Policies [login-with-amazon-dynamodb](#)

Permissions boundary Permissions boundary is not set

* Required

[Cancel](#) [Previous](#) [Create role](#)

Your role will now be created

The role `login-with-amazon-dynamodb` has been created.

[Create role](#) [Delete role](#)

Showing 1 result

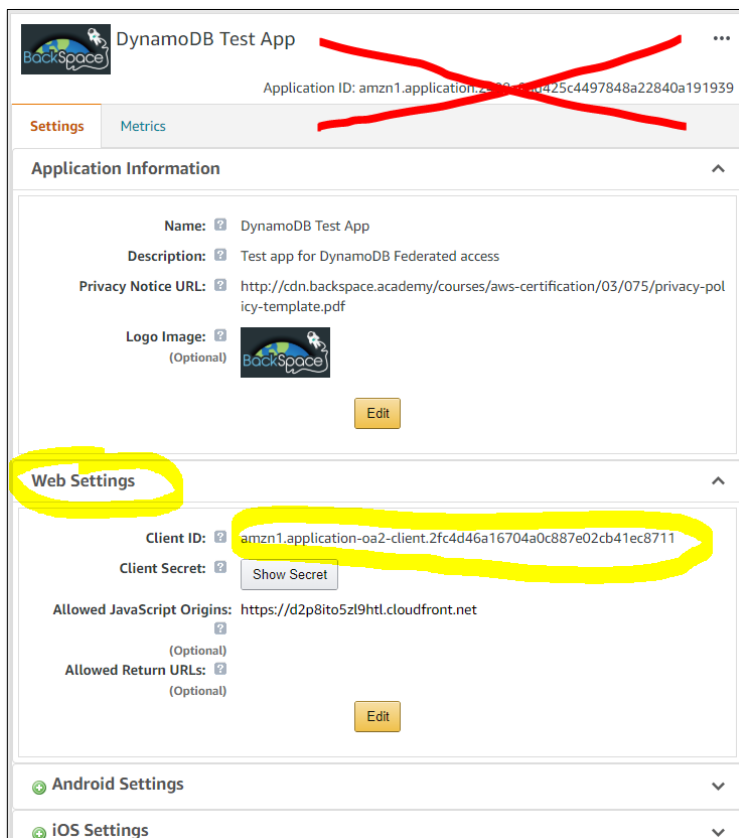
Role name ▾	Description	Trusted entities
<input type="checkbox"/> login-with-amazon-dynamodb	Login with Amazon Federated Identity for DynamoDB	Identity Provider: <code>www.amazon.com</code>

▶ Creating a Federated Identity Application

In this section we will use the **AWS JavaScript SDK** to create an application that uses **Login with Amazon** to verify the identity of users.

Go to <https://sellercentral.amazon.com>

Copy the Web Client ID (NOT YOUR APPLICATION ID) for your application



Open the index.html file using an editor

Paste your Web Client ID into the code:

```
<script type="text/javascript">
```

```

window.onAmazonLoginReady = function() {
    // Put your Login with Amazon Client ID in here
    amazon.Login.setClientId('YOUR-CLIENT-ID-GOES-HERE');
};
(function(d) {
    var a = d.createElement('script'); a.type = 'text/javascript';
    a.async = true; a.id = 'amazon-login-sdk';
    a.src = 'https://assets.loginwithamazon.com/sdk/na/login1.js';
    d.getElementById('amazon-root').appendChild(a);
})(document);
</script>

```

Scroll down to the end of the file and change app.js to app_v1.js

```

<!-- App Code -->
<script src="app_v1.js"></script>

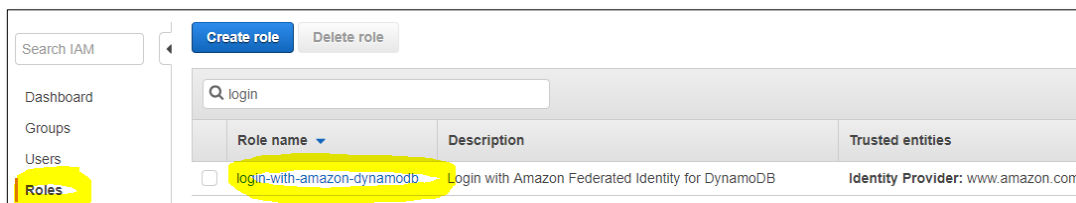
```

Save the file as index_v1.html (so that you don't have to invalidate the CloudFront distribution).

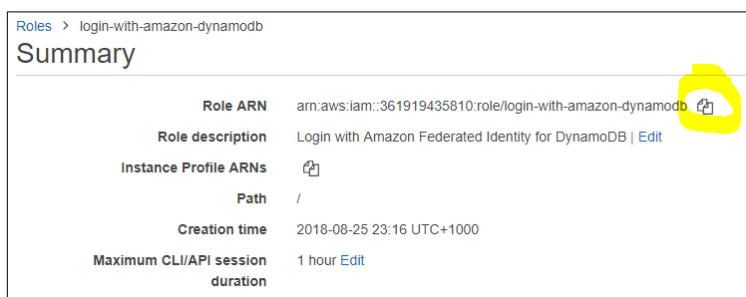
Go to the IAM management console

Select Roles

Click on the role you created



Copy the Role ARN



Open the app.js file using an editor

Paste your Role ARN into the code

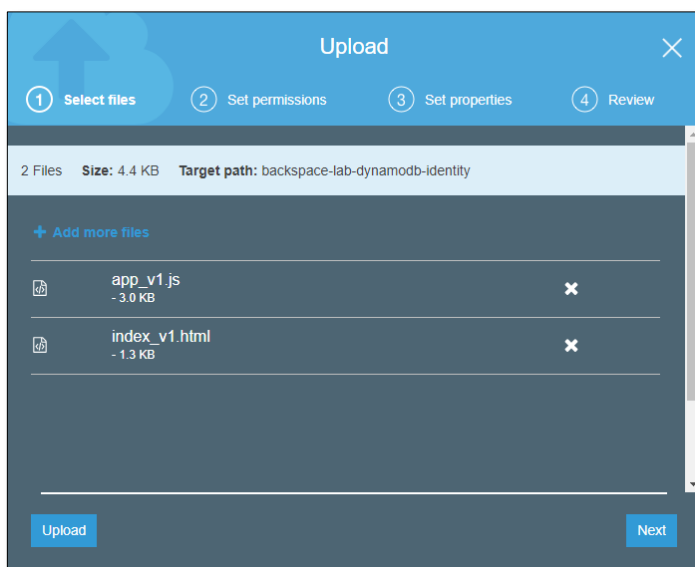
```
// Self-invoking anonymous function
(function($) {
    'use strict';

    // Region must be defined
    AWS.config.region = 'us-east-1';
    // Insert your IAM role arn here
    var roleArn = 'YOUR-ROLE-ARN-GOES-HERE';
```

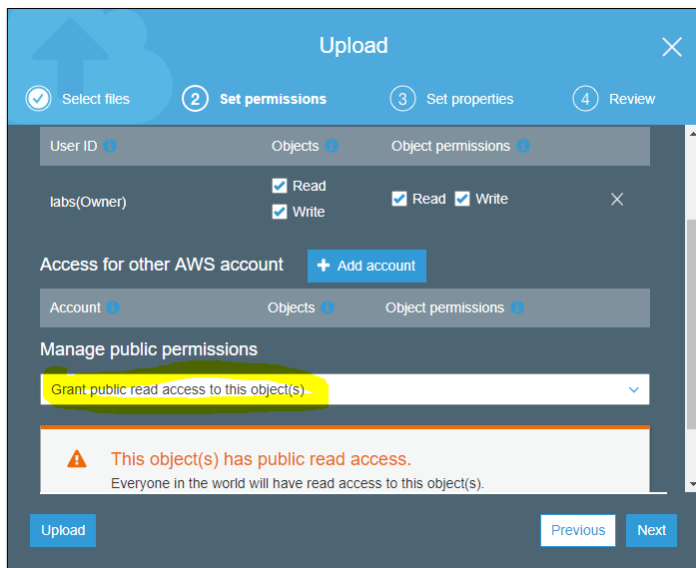
Save the file as *app_v1.js* (so that you don't have to invalidate the CloudFront distribution).

Go to the S3 management console

Upload *index_v1.html* and *app_v1.js*

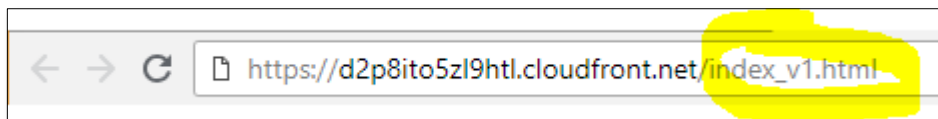


Make sure permissions set to public read.



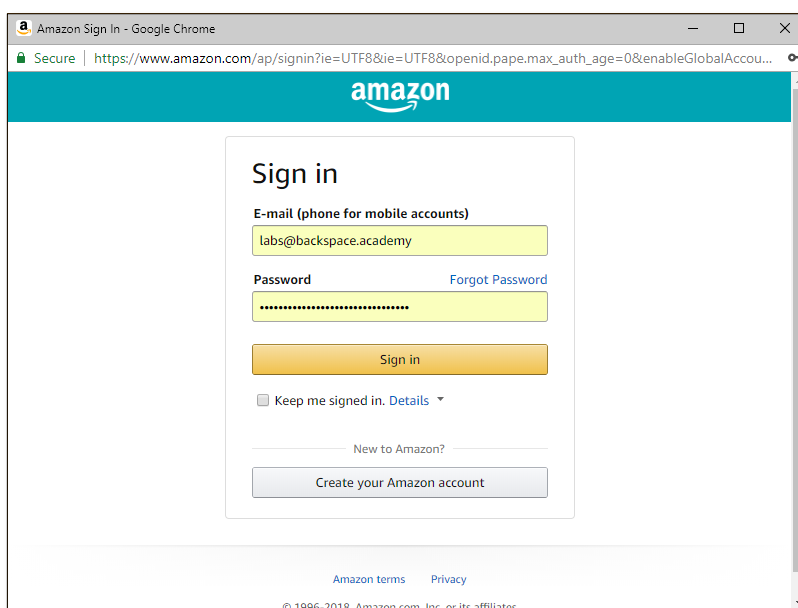
Open your browser to the URL of your CloudFront distribution

Navigate to index_v1.html



Click the Login with Amazon button

Enter your Amazon account details and sign in (if you don't see the pop-up then allow pop-ups in browser settings)



You should get an alert saying, "You are now signed in".

Press F12 for the Dev Tools to see the console output (don't worry about the Uncaught TypeError in Amazon's login code!)

```

ZOAIQVNI", "Name": "Paul Coady"}
Creating AWS Credentials for:
Role: arn:aws:iam::361919435810:role/login-with-amazon-dynamodb
Access Token: Atza|IwEBIKPLsmE3Qoh_z2VpiI-008-LVVFhbF8TyZt3dz-
JcytaQw3v8KNNQW8dSguMwvcwFAOERTzJdpQQadiw4Gd6i3eEtuttpTo01opubAA3r0zoYxvGVVZT-CjBxQ-
9STvTyP3Z85s_7hJMP1c1qNuB0PK0sehFIHIQ5IGrEbKyzHly7QmsivRzE7PUu5j3rR0WoyqsIdH0tmm_dKpF1F7NZZ6b8R-
_C6snzEoXXg17WuU7UcC1_F8QrXewrwCaMhd1nBoZLEEUndTzbBA8JQz09IUuo14DUpB3eZNh3DgvW6sq6Pkbq603TXpWAdXz8qC40ywkNO
jkQtpM1hR41i2h0w8x18n0b0NnoC1wX7T3o7h2OV68B9p__Bqr05EuoVwEudo7nUeY4tf-F0ZJgb1NvWRF1ZkpqiKp-
HbPG2v7Z4H1G1LRdu4PrYfBrFRXMMVbZZT48emSLx5j-OgsTmfjXQqkbwqiL_8gwX2Eyz1urY09zxjvpx1W-IrfZFUKN1wCqgzXNYrk1Bun6-
-891aAffb6L3C0h5b8XSuoqH-qIg
Successfully created AWS STS temporary credentials!
Uncaught TypeError: a.match is not a function
    at xa (login1.js:30)
    at d (login1.js:31)

```

Now Click "Write to DynamoDB"

You should get response *ConsumedCapacity*

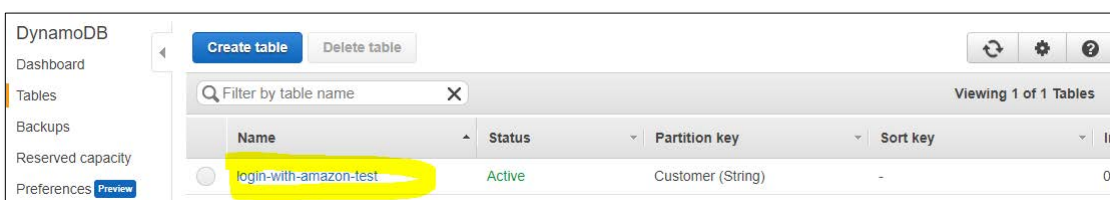
```

HbPG2v7Z4H1G1LRdu4PrYfBrFRXMMVbZZT48emSLx5j-OgsTmfjXQqkbwqiL_8gwX2Eyz1urY09zxjvpx1W-IrfZFUKN1wCqgzXNYrk1Bun6-
-891aAffb6L3C0h5b8XSuoqH-qIg
Successfully created AWS STS temporary credentials!
Writing to DynamoDB for CutoomerID:
"amzn1.account.AGSUB24IMQASKYABCFZOAIQVNI"
{"ConsumedCapacity":{"TableName":"login-with-amazon-test","CapacityUnits":1}}

```

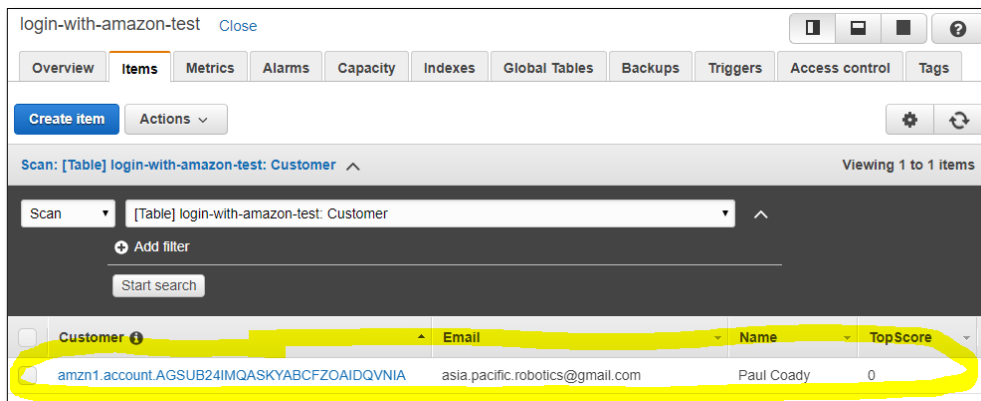
Go back to the DynamoDB management console

Select the table



Go to the Items tab

You will see an item created containing your Amazon profile details and a TopScore of 0.



login-with-amazon-test [Close](#)

Overview **Items** Metrics Alarms Capacity Indexes Global Tables Backups Triggers Access control Tags

Create item Actions

Scan: [Table] login-with-amazon-test: Customer Viewing 1 to 1 items

Scan [Table] login-with-amazon-test: Customer

Add filter

Start search

Customer	Email	Name	TopScore
amzn1.account.LAGSUB24IMQASKYABCFZOAIQVANIA	asia.pacific.robotics@gmail.com	Paul Coady	0

Clean Up

Do not delete the S3 website and CloudFront distribution you created as it will be needed in the lab on *Programming AWS Lambda*.

Go to the DynamoDB management console and delete the table

