lab

lab title

Developing Serverless JavaScript DynamoDB Applications

V1.01

Course title

AWS Certified Associate

Table of Contents

Contents

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 Fedrated 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

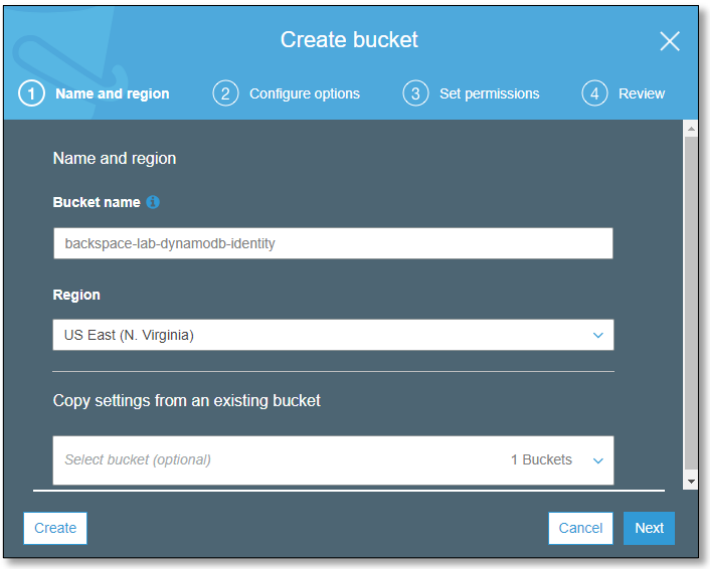
Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

Upload Files to an S3 Bucket

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

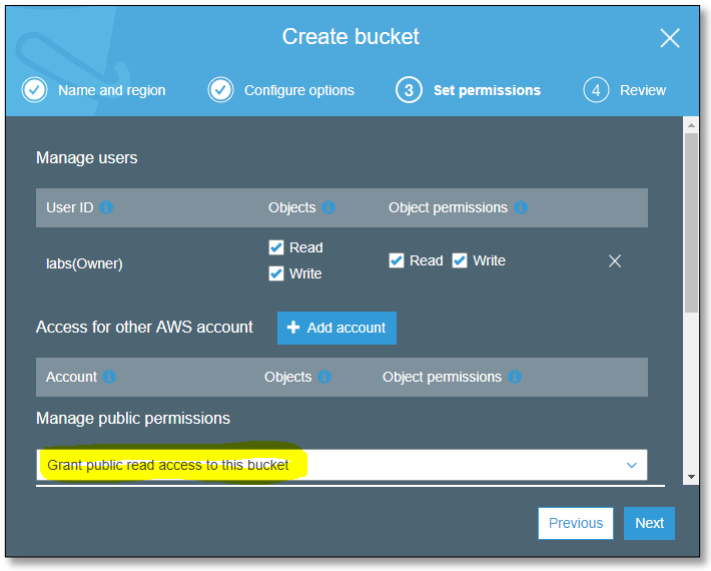
Go to the S3 management console.

Create a bucket with a unique name



Click *Next*

Select *Grant public read access to this bucket*



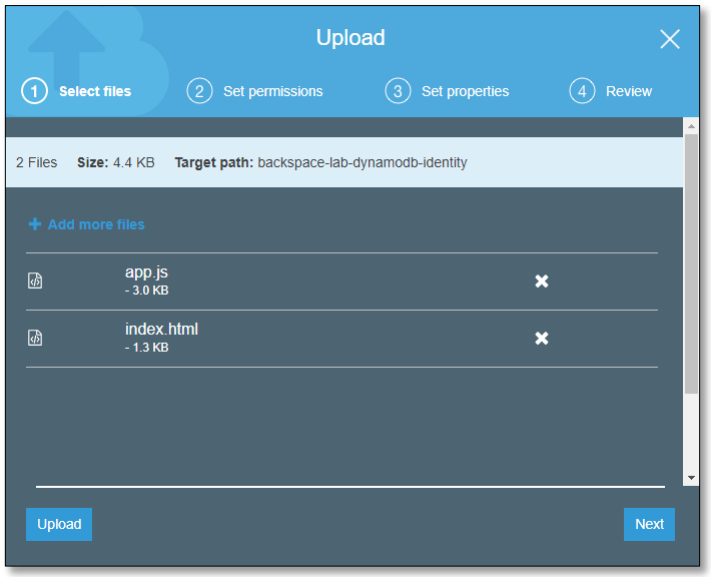
Click *Create bucket*

Download the following files:

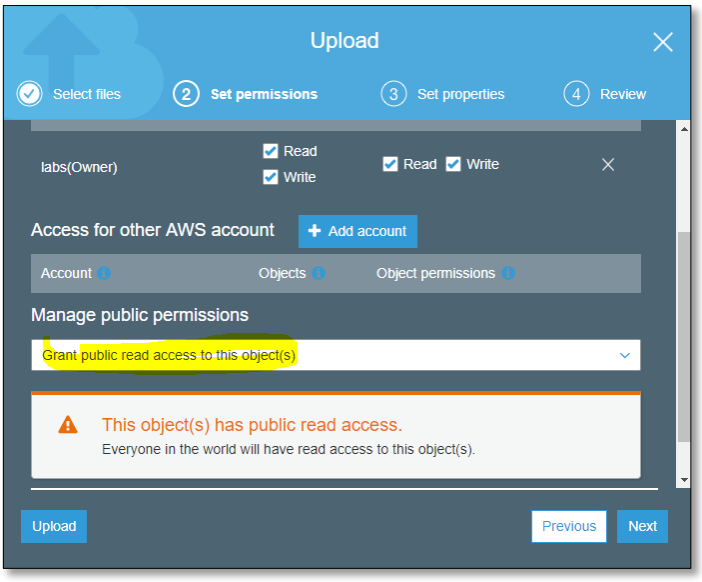
https://raw.githubusercontent.com/backspace-academy/aws-dynamodb-login-with-amazon/master/app.js

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)*



Creating a CloudFront Distribution

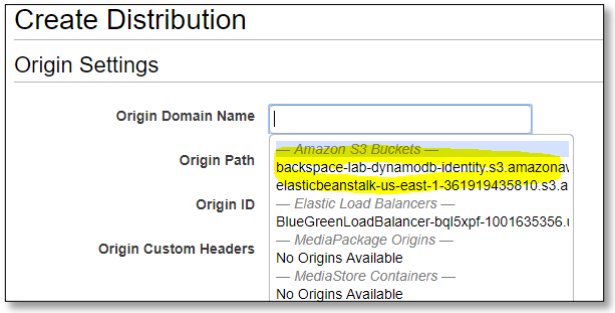
In this section we will use the Amazon CloudFront service to cache our website and enable https.

Go to the CloudFront console

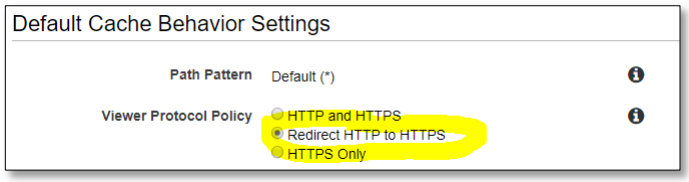
Click *Create Distribution*

Select *Web*

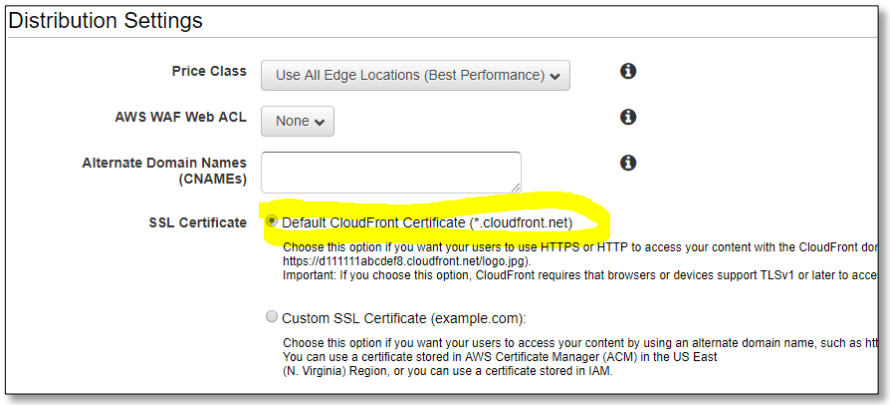
Select the bucket you created for *Origin Domain Name*



Select *Redirect HTTP to HTTPS*

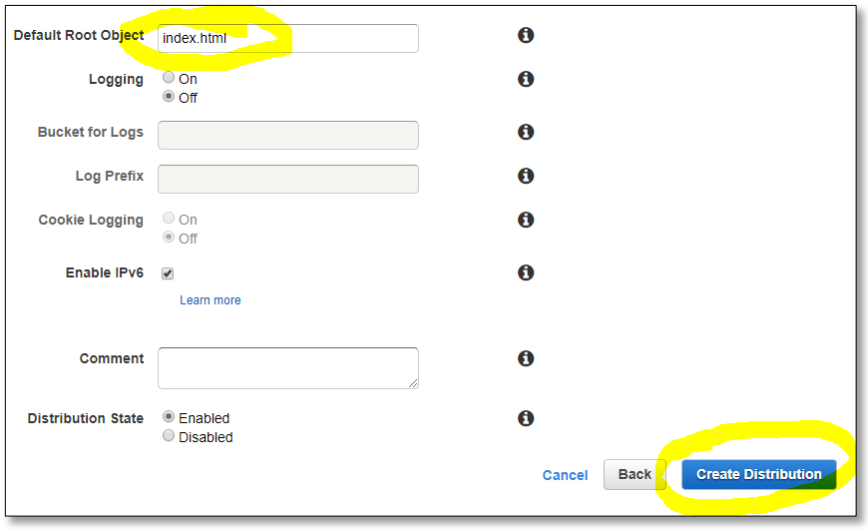


Select *Default CloudFront Certificate (*.cloudfront.net)*



Enter index.html for *Default Root Object*

Click *Create Distribution*

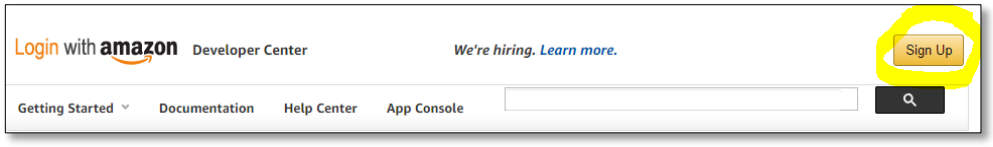


Registering an Amazon Developer App

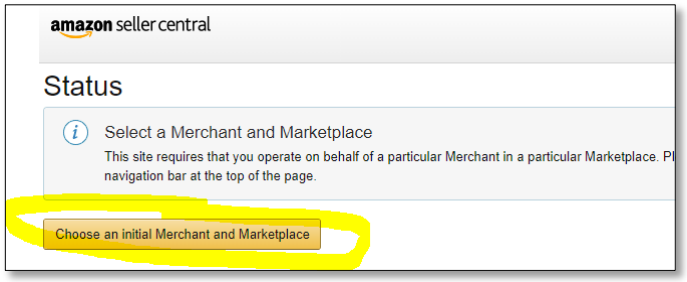
In this section we will use the Amazon Developer Portal to register a Login with Amazon app.

Go to https://login.amazon.com

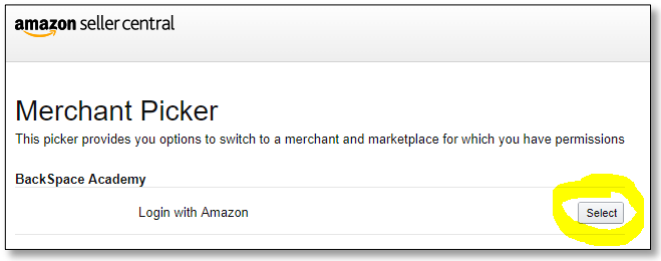
Use your Amazon login details to register as an Amazon developer



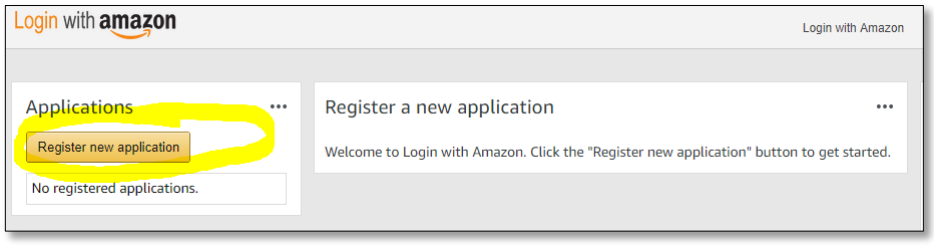
You will be redirected to Amazon Seller Central. Click *Choose an initial Merchant and Marketplace*



Click *Select*



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

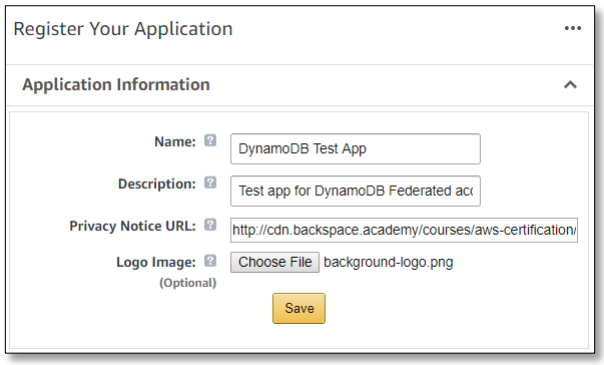


Give your app a name and description

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

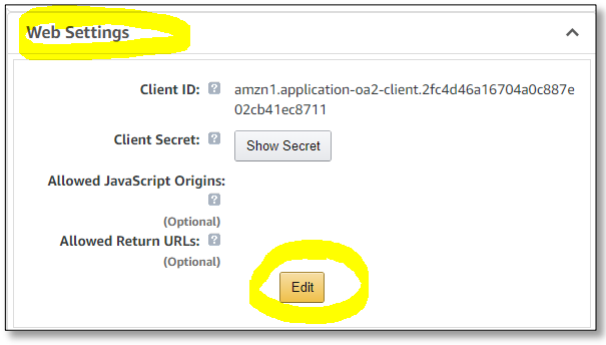
You can optionally upload an image if you wish

Click *Save*

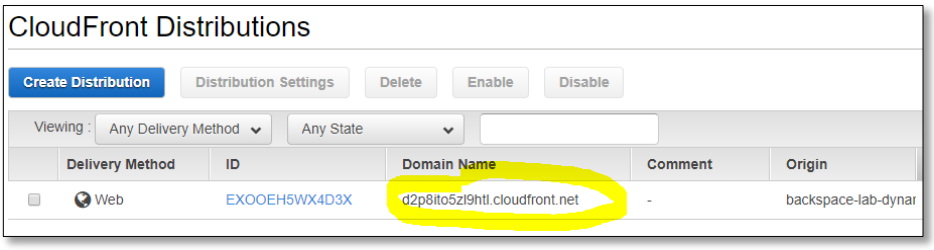


Select *Web Settings*

Click *Edit*

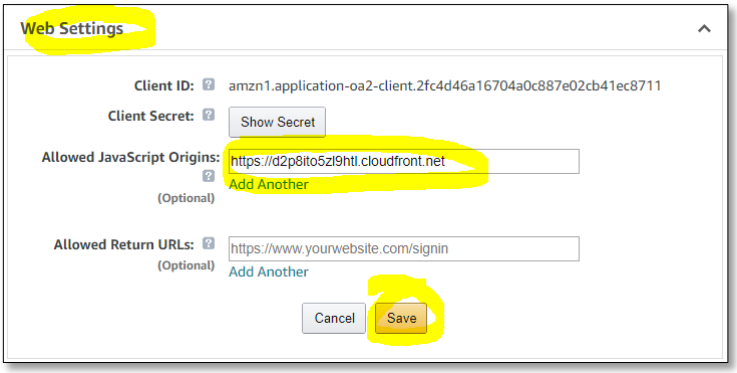


Go to the CloudFront management console and copy the domain of your distribution.

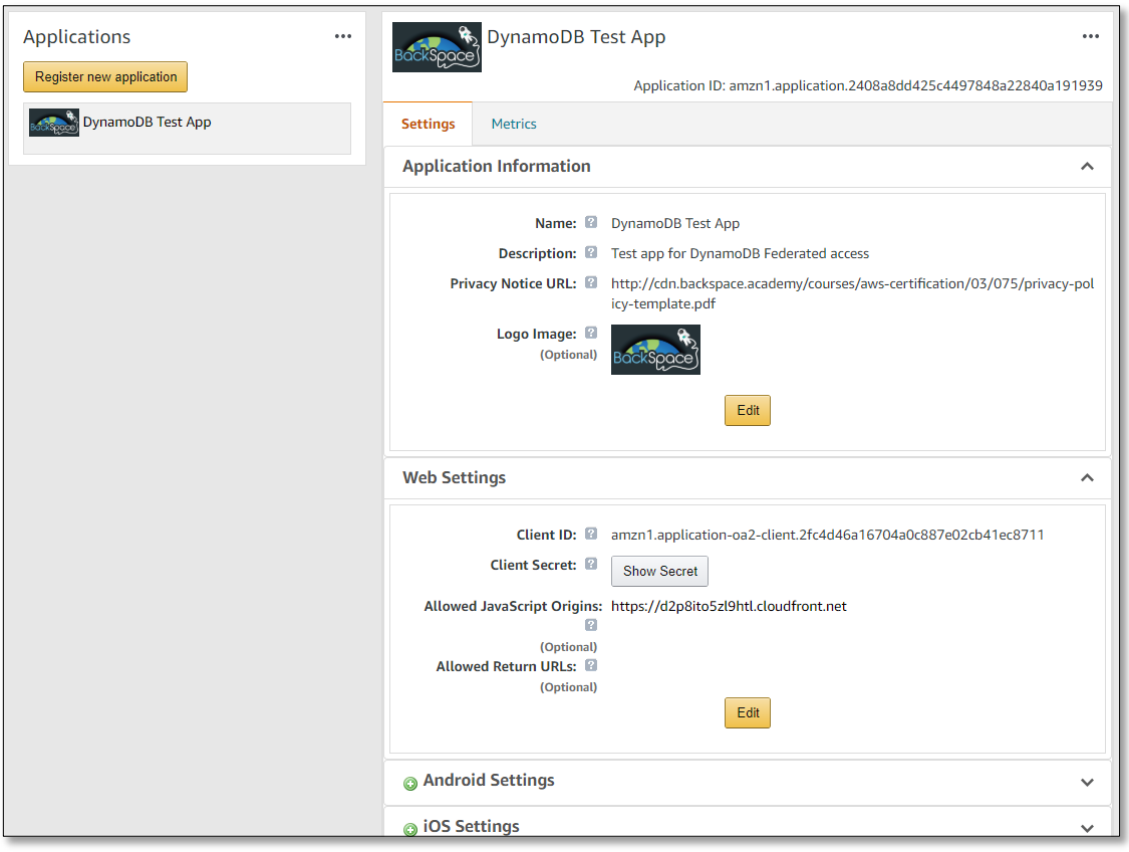


Add the https domain name **with https://** of your CloudFront distribution for *Allowed JavaScript Origins*

Click *Save*



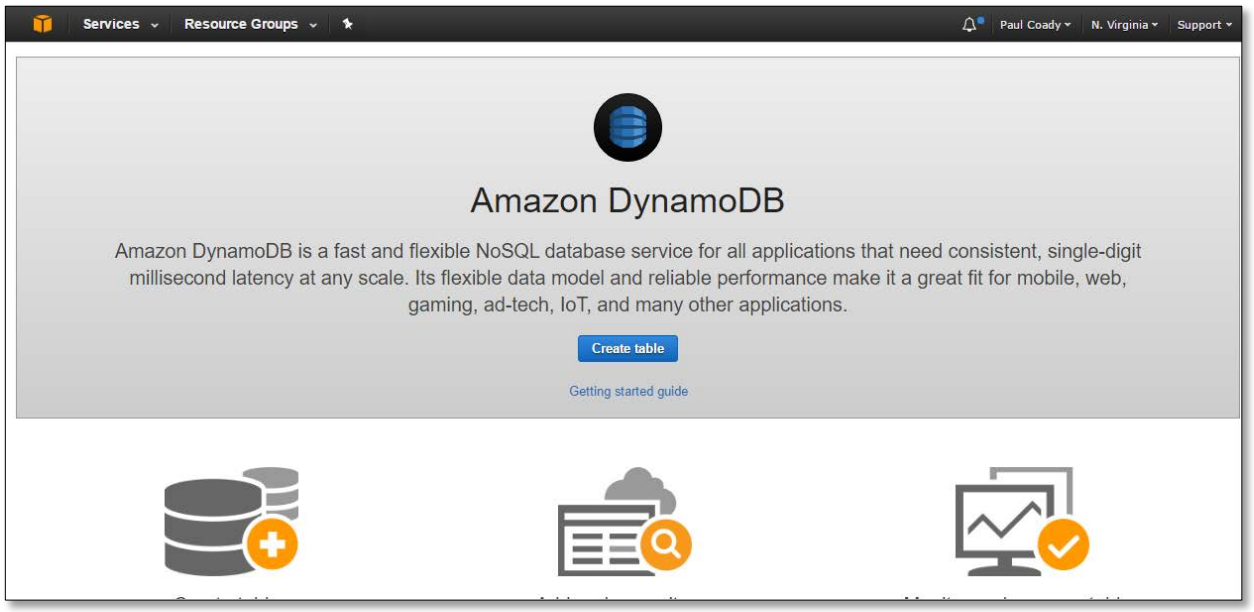
Your app is now registered



Creating a DynamoDB Table

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

Select the DynamoDB Console



Click “Create Table”

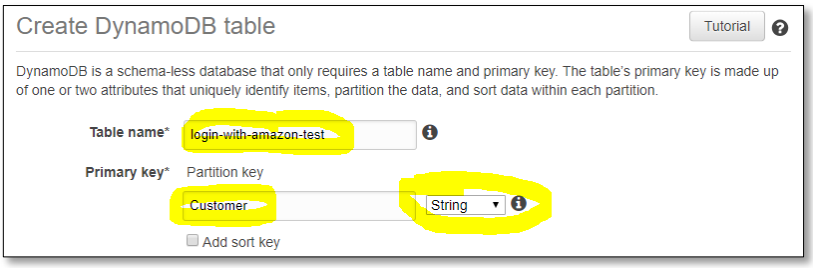
Enter the following details (enter exactly with correct case)

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

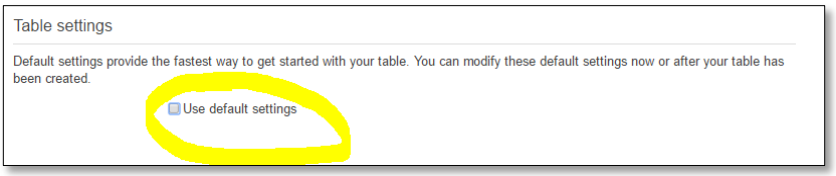
Table Name: login-with-amazon-test

Primary key Partition key: Customer (case sensitive)

Attribute Type: String

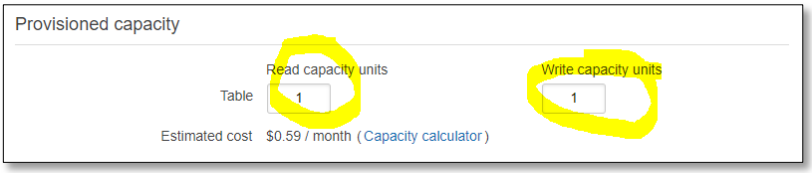


Uncheck *Use Default Settings*



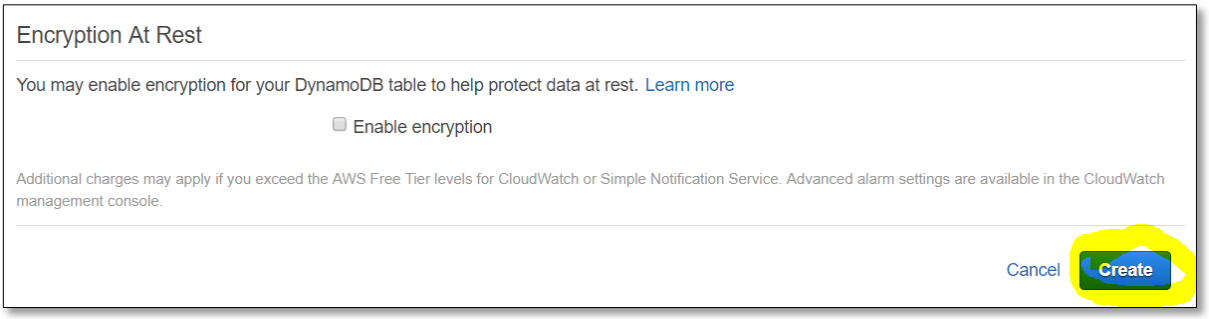
Disable auto scaling

Change provisioned capacity to 1.



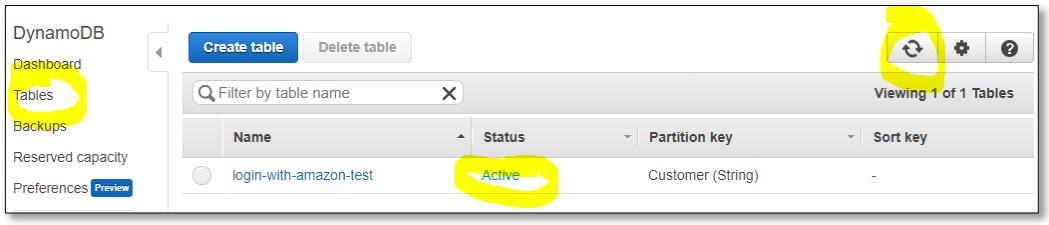
Leave encryption disabled.

Click Create.



Go to the *Tables* tab

Press refresh until table status is listed as active.

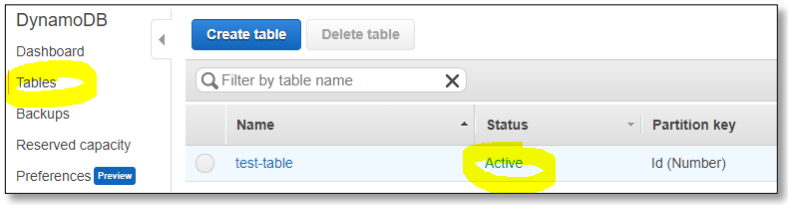


Creating an IAM Role for Federated Identity

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.

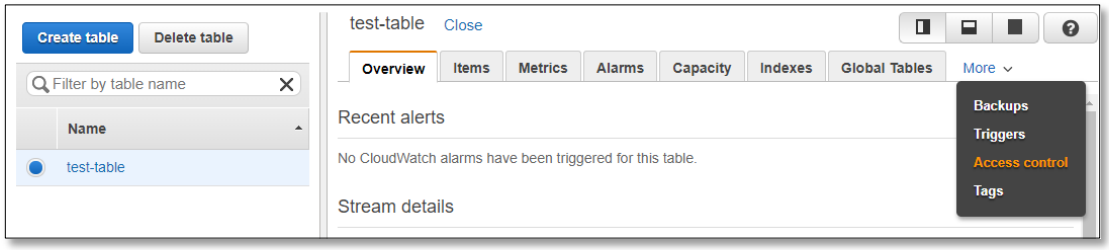
From the DynamoDB console select *Tables*

Wait until your table status is active



Select the table you created

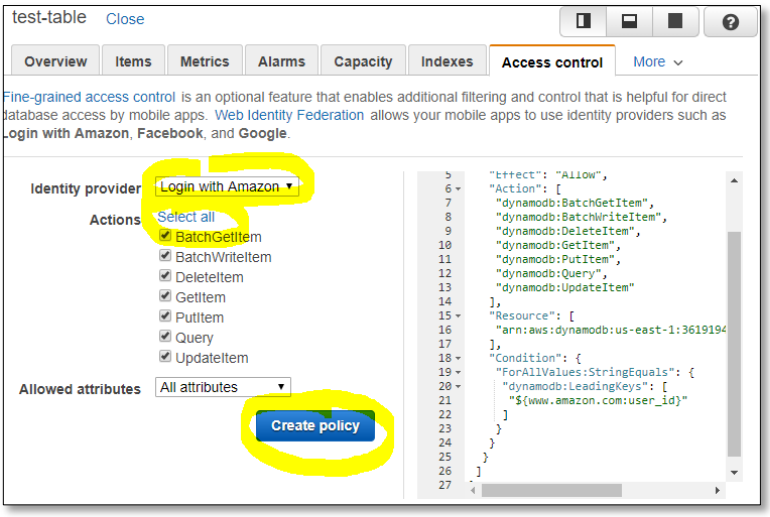
Select *Access control*



Select Login with Amazon

Select All Actions

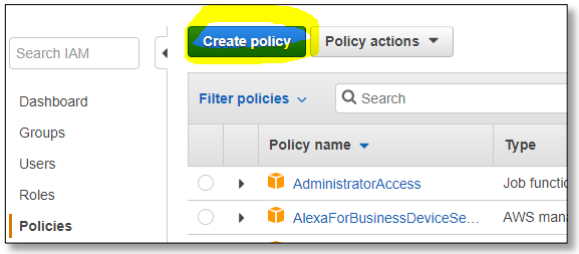
Click *Create policy*



Open the IAM console in another browser tab

Select *Policies*

Click *Create policy*

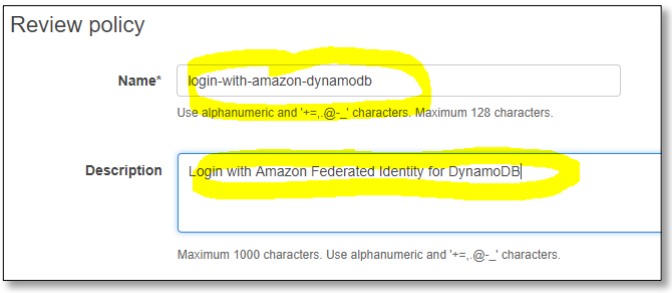


Select the *JSON* tab

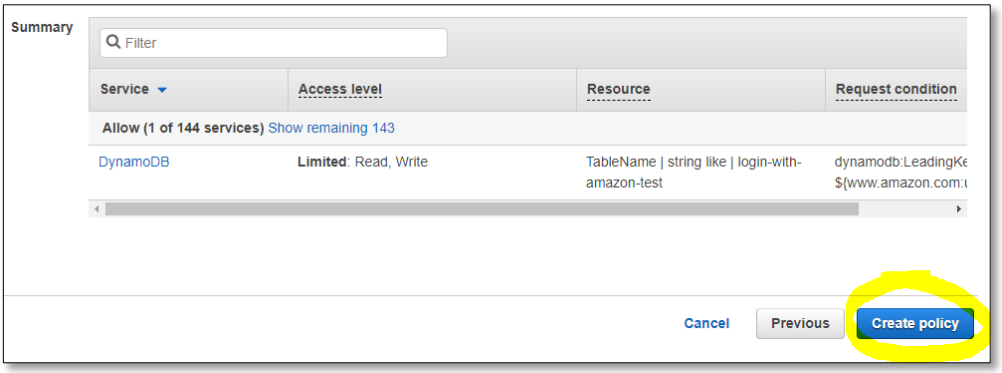
Copy and paste the policy you created previously in the DynamoDB console

Click *Review policy*

Give the policy a name and description



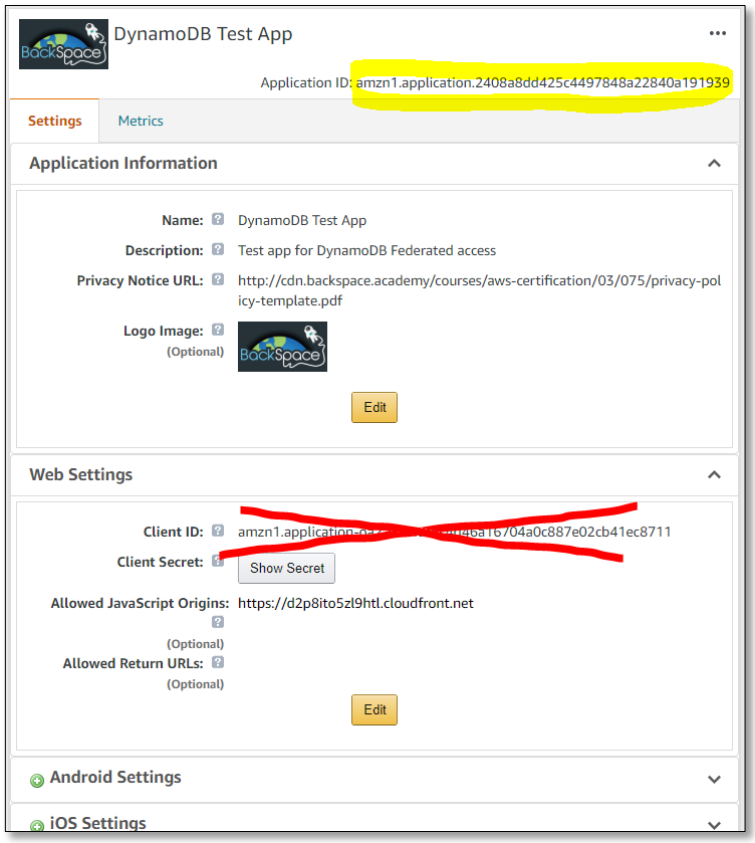
Click *Create policy*



Go to https://sellercentral.amazon.com

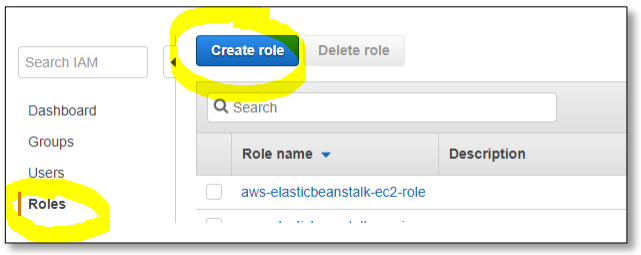
Copy the *Application ID* of your app

PLEASE NOTE COPY THE APPLICATION ID NOT THE CLIENT ID



Go to *Roles*

Click *Create role*

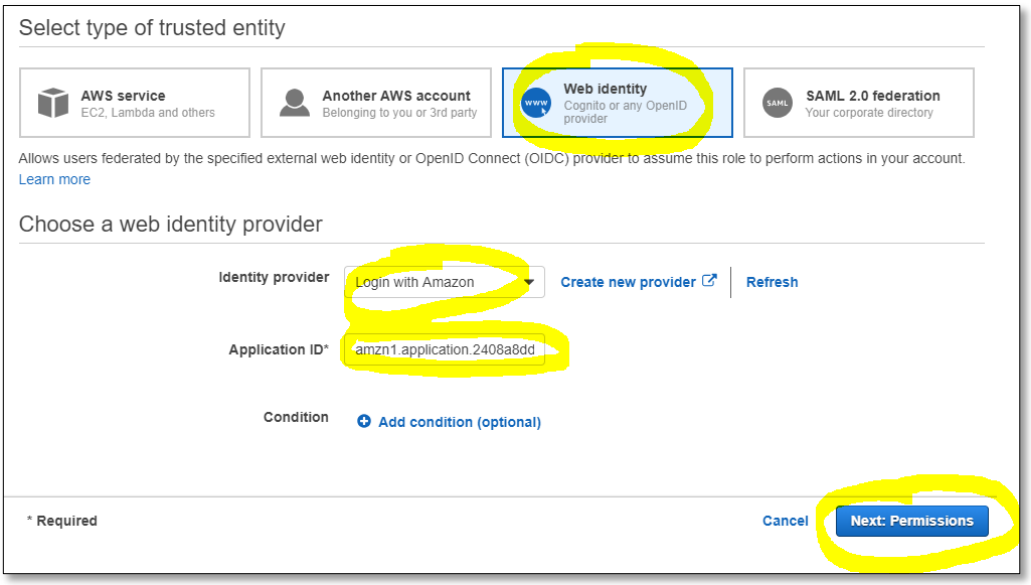


Select *Web identity*

Select *Login with Amazon*

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

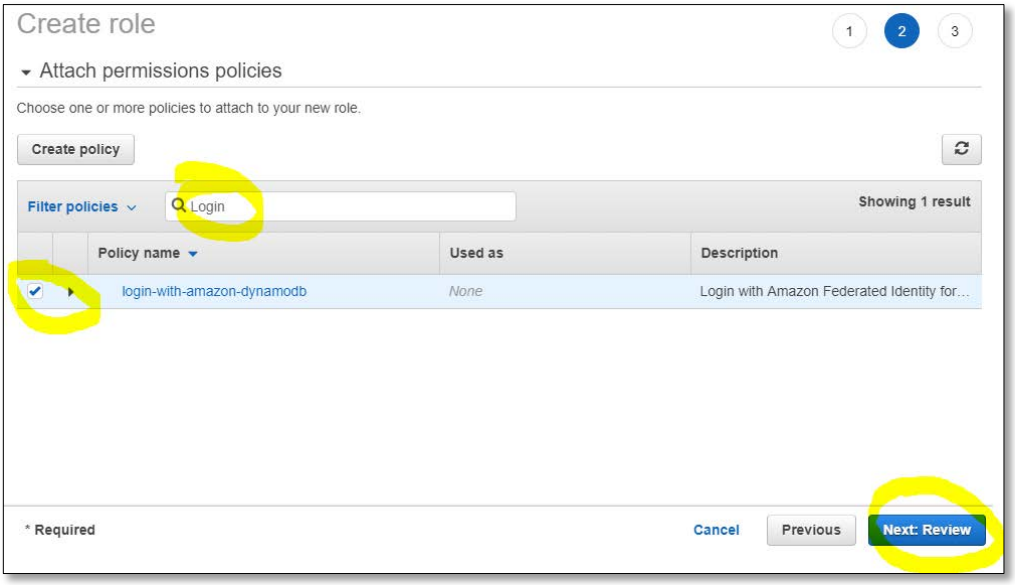
Click *Next: Permissions*



Search for the policy you created

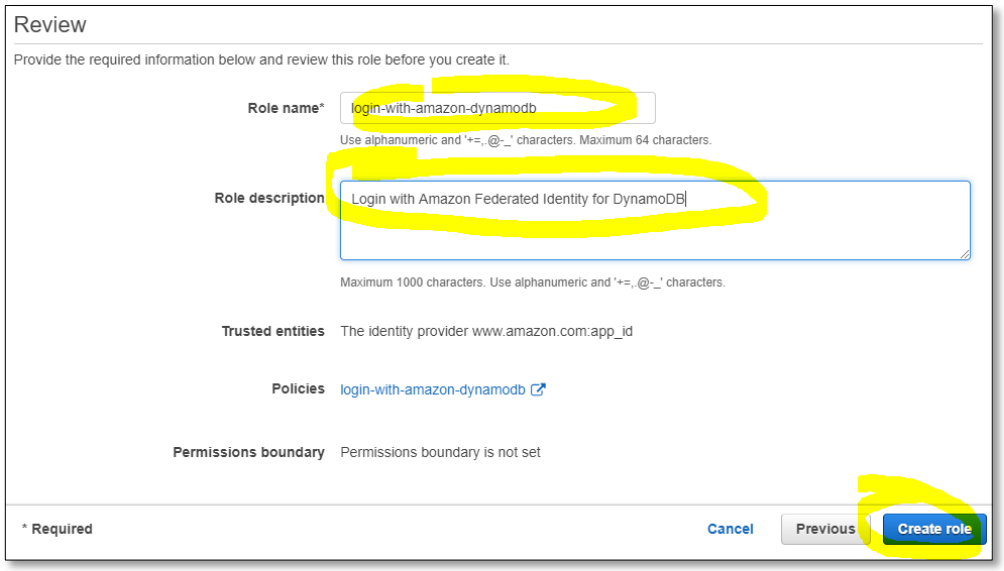
Select the policy

Click *Next: Review*

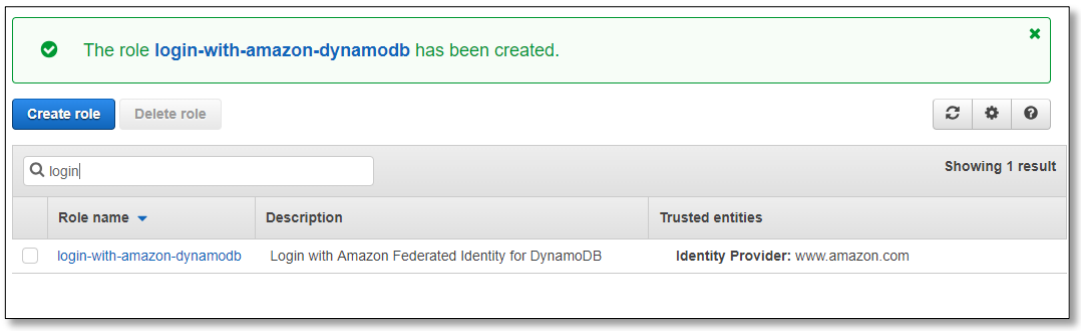


Give the role a name and description

Click *Create role*



Your role will now be created

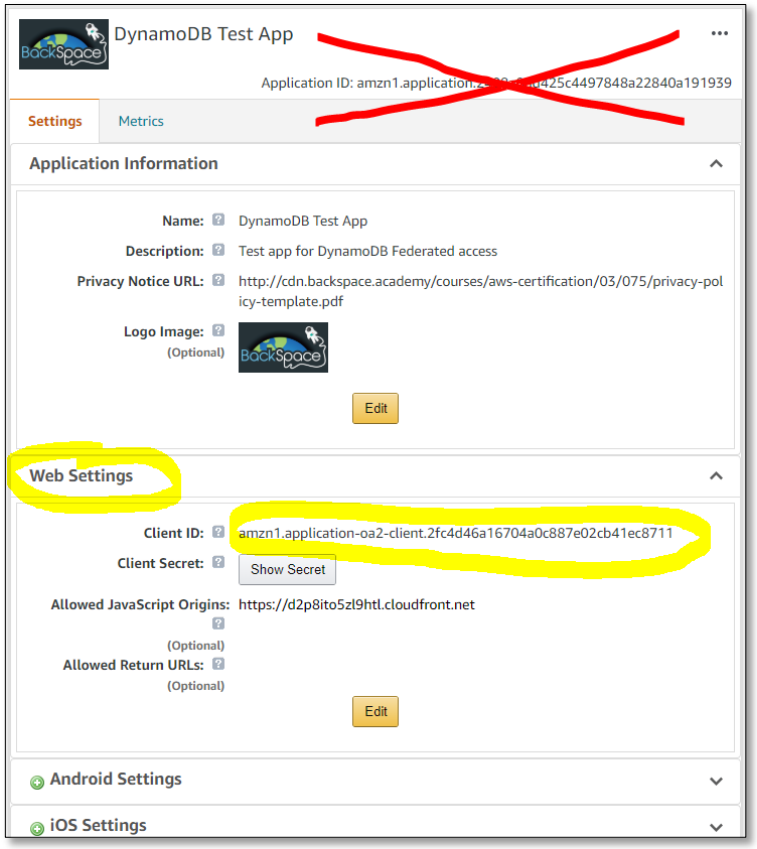


Creating a Federated Identity Application

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

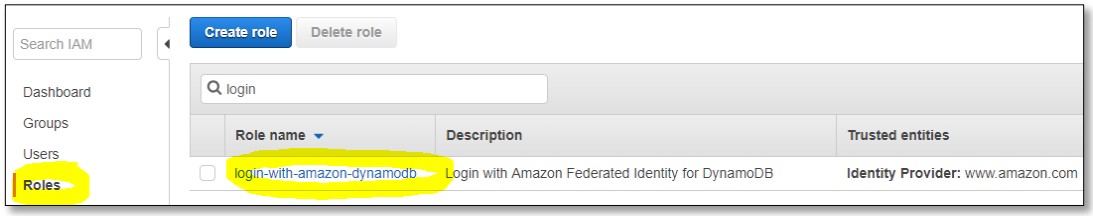
```
        <!-- App Code -->
        <script src="app_v1.js"></script>
```

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

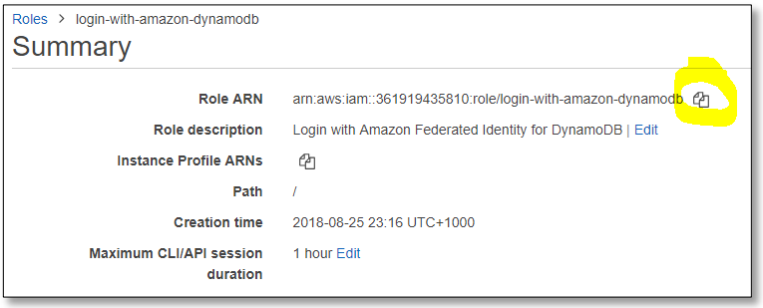
Go to the IAM management console

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

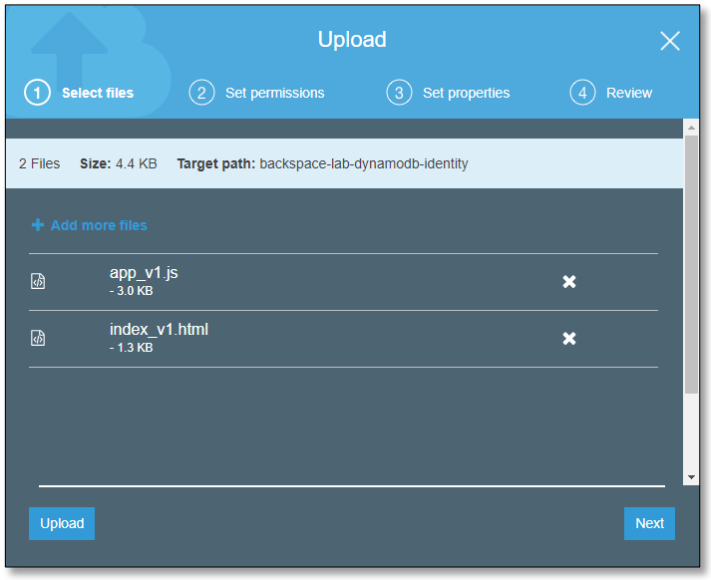
```
// Self-invoking anonymous function
(function($) {
        'use strict';

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

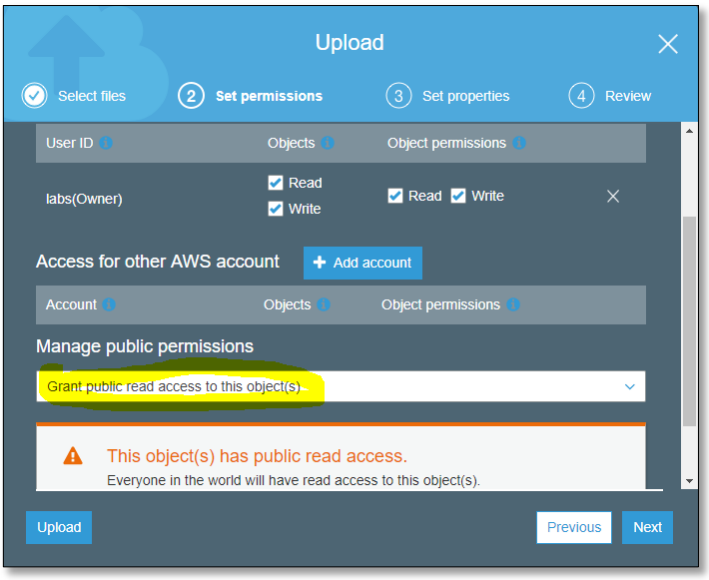
Save the file as *app_v1.js* (so that you don't have to invalidate the CloudFront distribution).

Go to the S3 management console

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

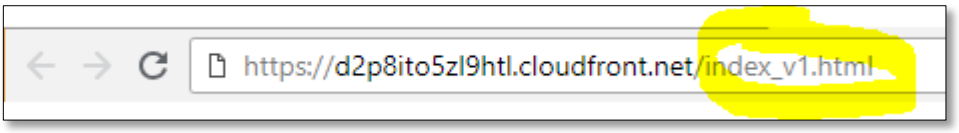


Make sure permissions set to public read.



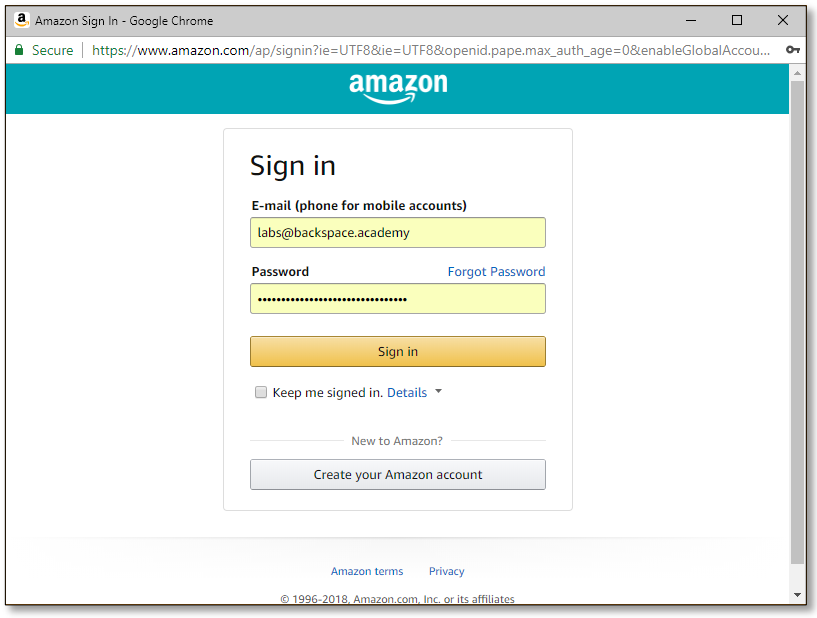
Open your browser to the URL of your CloudFront distribution

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)

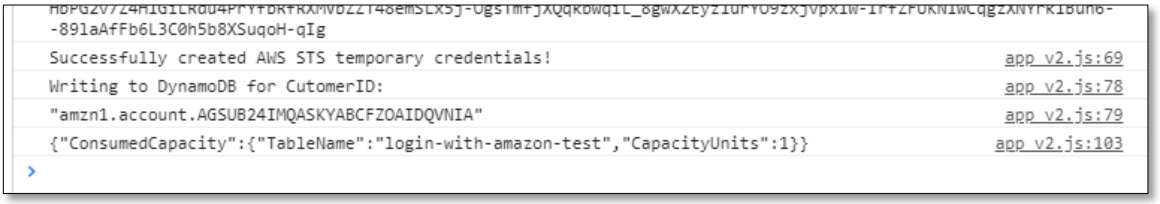


You should get an alert saying, “You are now signed in”.

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

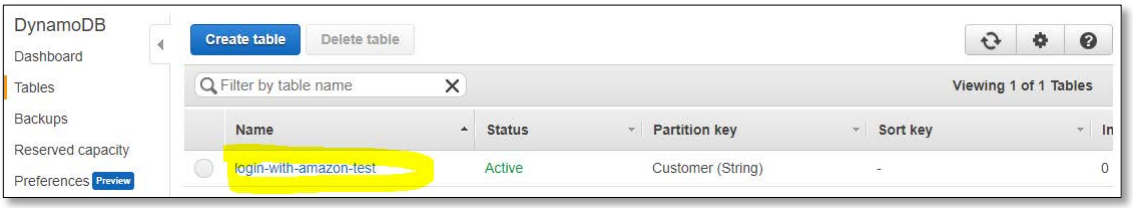
Now Click “Write to DynamoDB

You should get response *ConsumedCapacity*



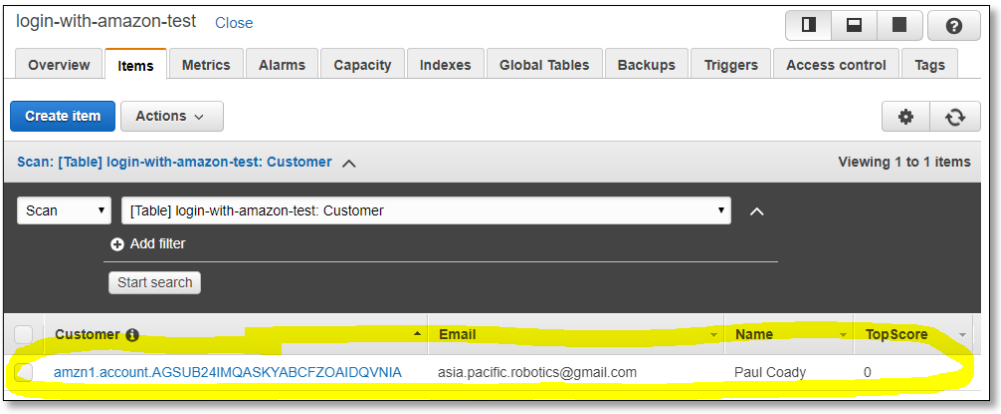
Go back to the DynamoDB management console

Select the table



Go to the Items tab

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



Clean Up

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

Go to the DynamoDB management console and delete the table