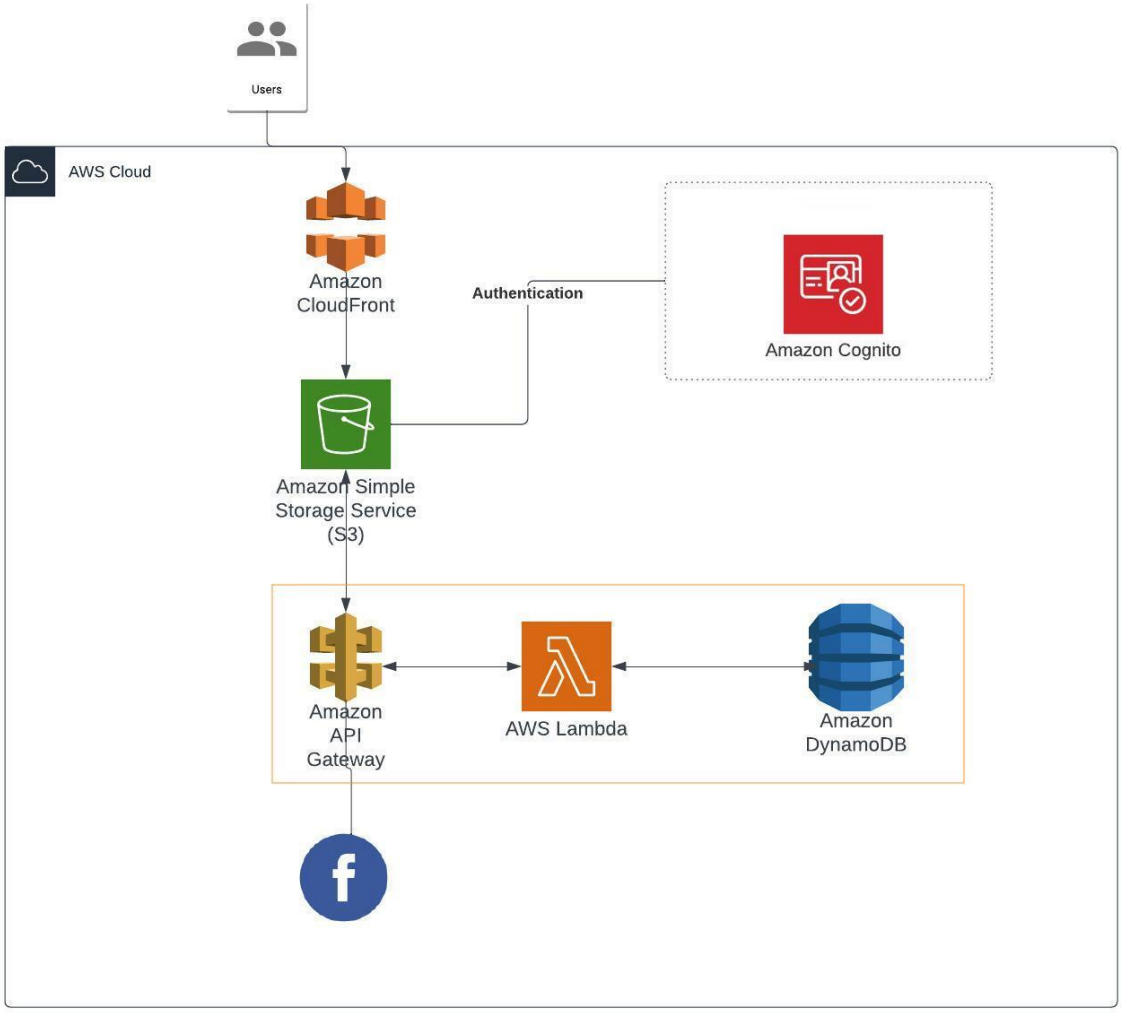
Assignment Full

Official System Architecture

This section describes the high level system design and the architecture of Coin 360. Coin 360 is a server-less web application deployed using Amazon webs services. Coin 360 is written in JavaScript using the react framework. Below is the system architecture of the application



Code Description

Dynamo DB:

The first dynamo db table is used to store a selected range of coins to be selected from when creating a new coin entry. The second table is used to store the record that is created by the user.

The id is the main component of the record

```
{
  "id": "Solana",
  "lastName": "",
  "firstName": "Solana"
}
```

Lambda:

Five lambda roles were created for the following uses:

- Getting all records for the user to choose a coin from

index.js Execution results

```
const AWS = require("aws-sdk");
const dynamodb = new AWS.DynamoDB({
  region: "us-east-1",
  apiVersion: "2012-08-10"
});

exports.handler = (event, context, callback) => {
  const params = {
   TableName: "authors"
  };
  dynamodb.scan(params, (err, data) => {
    if (err) {
      console.log(err);
      callback(err);
    } else {
       const authors = data.Items.map(item => {
         return { id: item.id.S, firstName: item.firstName.S, lastName: item.lastName.S };
       });
       callback(null, authors);
    }
  });
};
```

- Get all records that are saved and made by the user

```javascript
const AWS = require("aws-sdk");
const dynamodb = new AWS.DynamoDB({
  region: "us-east-1",
  apiVersion: "2012-08-10"
});

exports.handler = (event, context, callback) => {
  const params = {
    TableName: "courses"
  };
  dynamodb.scan(params, (err, data) => {
    if (err) {
      console.log(err);
      callback(err);
    } else {
      const courses = data.Items.map(item => {
        return {
          id: item.id.S,
          title: item.title.S,
          watchHref: item.watchHref.S,
          authorId: item.authorId.S,
          length: item.length.S,
          category: item.category.S
        };
      });
      callback(null, courses);
    }
  });
};
```

- Get a single record that is saved and made by the user

```javascript
const AWS = require("aws-sdk");
const dynamodb = new AWS.DynamoDB({
  region: "us-east-1",
  apiVersion: "2012-08-10"
});

exports.handler = (event, context, callback) => {
  const params = {
    Key: {
      id: {
        S: event.id
      }
    },
    TableName: "courses"
  };
  dynamodb.getItem(params, (err, data) => {
    if (err) {
      console.log(err);
      callback(err);
    } else {
      callback(null, {
        id: data.Item.id.S,
        title: data.Item.title.S,
        watchHref: data.Item.watchHref.S,
        authorId: data.Item.authorId.S,
        length: data.Item.length.S,
        category: data.Item.category.S
      });
    }
  });
};
```

- Update(Edit) a chosen record

```javascript
const AWS = require("aws-sdk");
const dynamodb = new AWS.DynamoDB({
  region: "us-east-1",
  apiVersion: "2012-08-10"
});

exports.handler = (event, context, callback) => {
  const params = {
    Item: {
      id: {
        S: event.id
      },
      title: {
        S: event.title
      },
      watchHref: {
        S: event.watchHref
      },
      authorId: {
        S: event.authorId
      },
      length: {
        S: event.length
      },
      category: {
        S: event.category
      }
    },
    TableName: "courses"
  };
  dynamodb.putItem(params, (err, data) => {
    if (err) {
      console.log(err);
      callback(err);
    } else {
      callback(null, {
        id: params.Item.id.S,
        title: params.Item.title.S,
        watchHref: params.Item.watchHref.S,
        authorId: params.Item.authorId.S,
        length: params.Item.length.S,
        category: params.Item.category.S
      });
    }
  });
};
```

- Delete a chosen record

```javascript
const AWS = require("aws-sdk");
const dynamodb = new AWS.DynamoDB({
  region: "us-east-1",
  apiVersion: "2012-08-10"
});

exports.handler = (event, context, callback) => {
  const params = {
    Key: {
      id: {
        S: event.id
      }
    },
    TableName: "courses"
  };
  dynamodb.deleteItem(params, (err, data) => {
    if (err) {
      console.log(err);
      callback(err);
    } else {
      callback(null, data);
    }
  });
};
```

API Gateway:

The Rest API is created and managed by AWS and is used for the front end services of the application the GET POST and DELETE methods are created to manage the database interactions using the Lambda function.

Developer/Use Manual

The application was originally developed using a local server but now is uploaded to AWS S3 to hold the files and also to be accessed by the public.

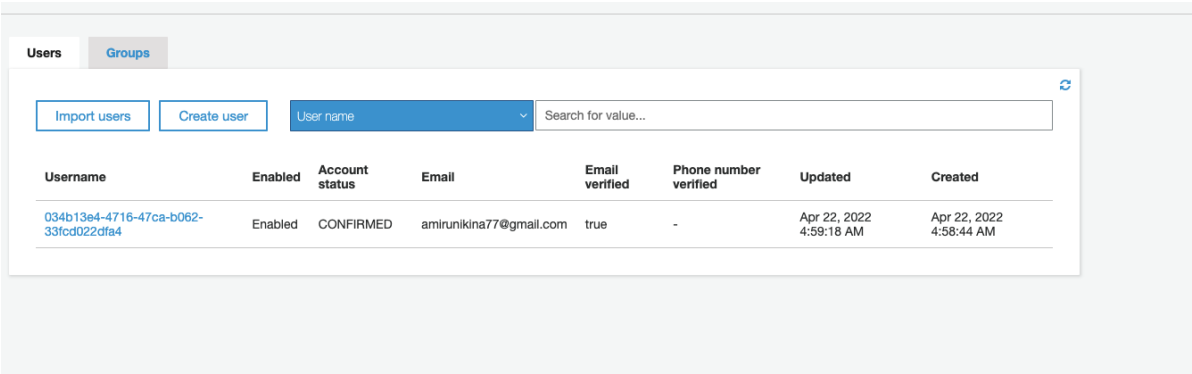
https://d1asnh0towkht9.cloudfront.net/

However the code can also be used locally using the following steps located in the ReadMe text file.

```
//To load the app on the local machine
Step 1: Download the files to chosen location(Ensure javascript is loaded)
Step 2: Open Command Terminal and navigate to the file
Step 3: First Run: "npm install" to download all node modules needed to run the application or else there will be an error
Step 4: Then Run: "npm start" to begin the server and launch the application locally
```

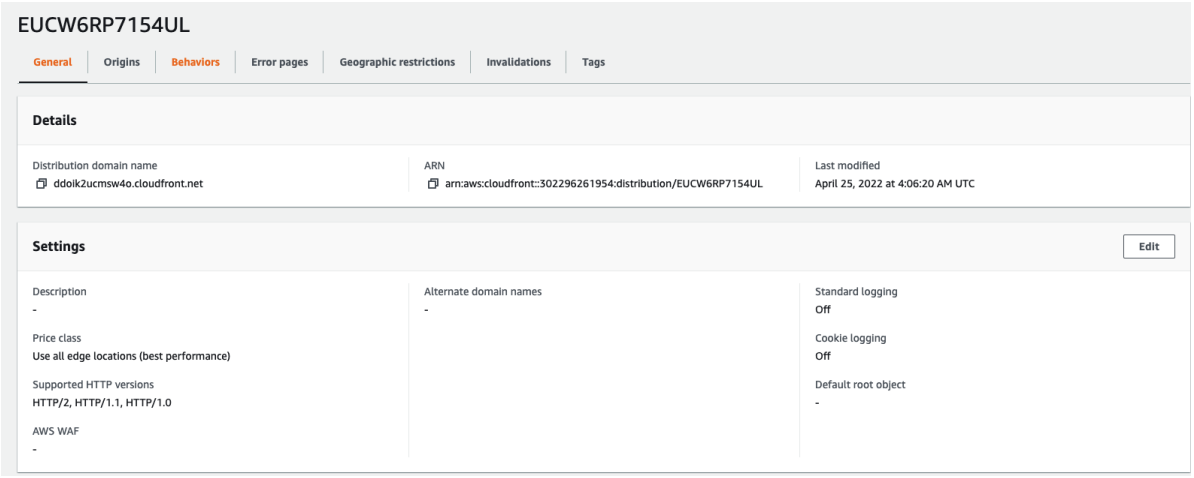
AWS Cognito:

AWS cognito is used to manage access to the web application, this also allows for log in and registration functionality.



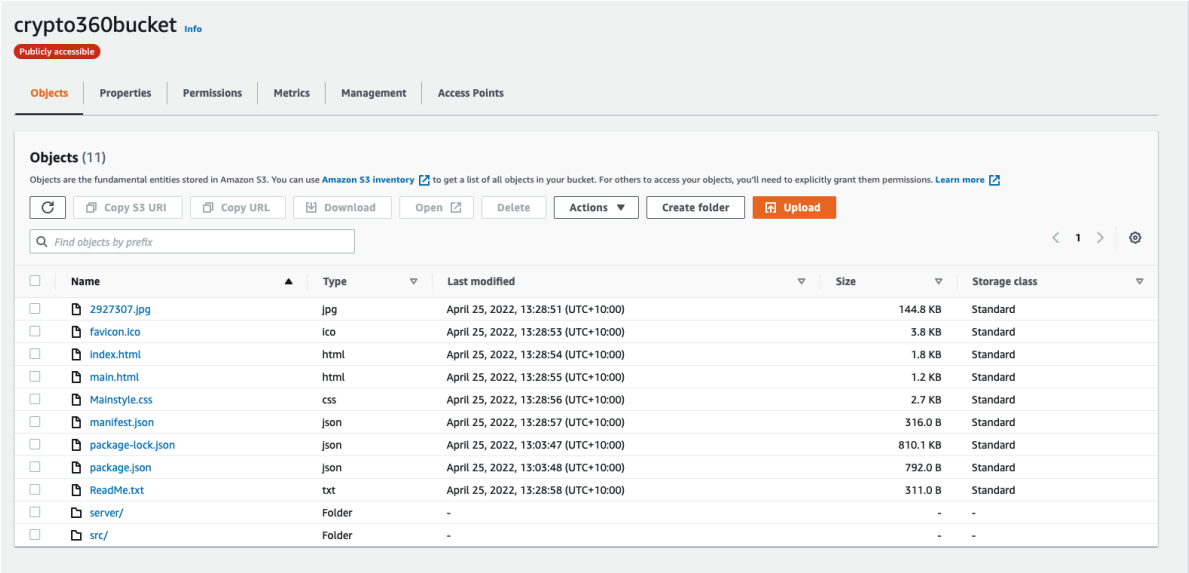
AWS CloudFront:

AWS Cloud front is used to allow the application to load faster



AWS S3:

Stores the files for the application including the media



References:

[1]"Facebook for Developers", *Developers.facebook.com*
, 2022. [Online]. Available: https://developers.facebook.com. [Accessed: 25- Apr- 2022].

[2]"Module Five", *Amazon Web Services, Inc.*
, 2022. [Online]. Available: https://aws.amazon.com/getting-started/hands-on/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/module-5/. [Accessed: 25- Apr- 2022].

[3]E. Herrera and B. Services, "Build a Serverless Web App on AWS Services | Pluralsight | Pluralsight", *Pluralsight.com*
, 2022. [Online]. Available: https://www.pluralsight.com/guides/building-a-serverless-web-app-on-aws-services. [Accessed: 25- Apr- 2022].