COSC2626 Cloud Computer 2019 – Assignment 2

RMIT University

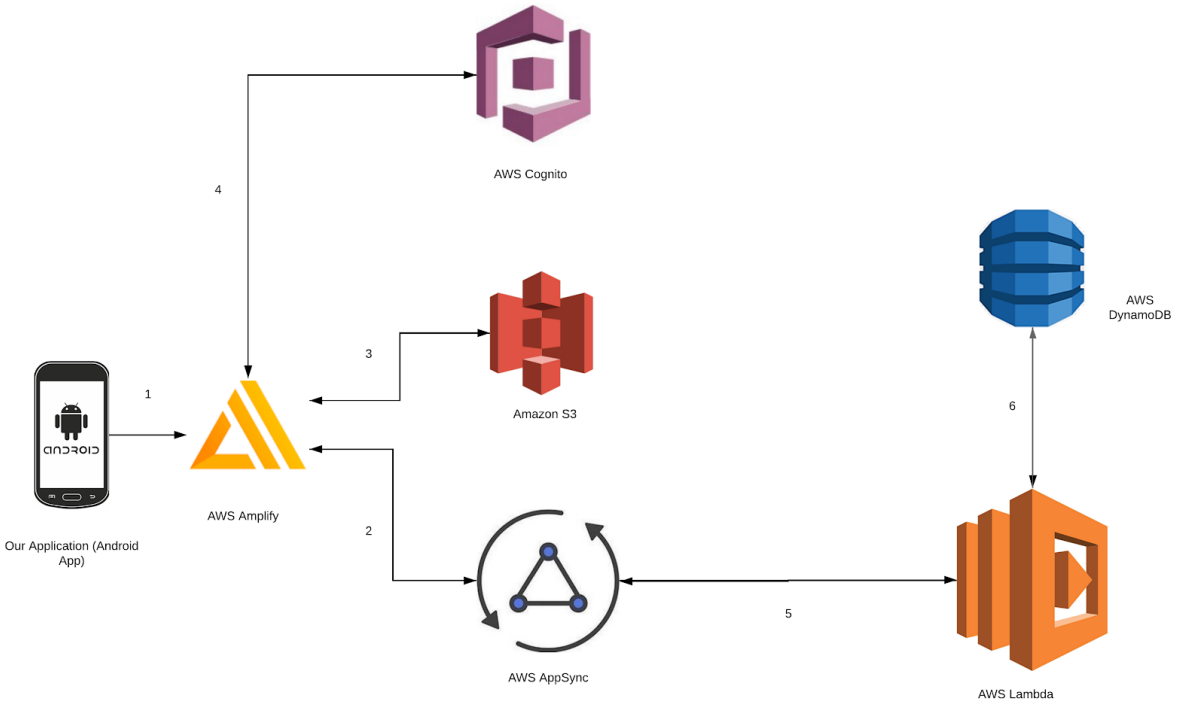
# Employee Tracker

Sara Moghaddasian s667123  Orion Selvon s3504057

## Introduction

We were looking to do a program that could efficiently etilize multiple Cloud based services and different API's while also keeping the app design relatively simple while being easy to use and understand for all types of users.

"Employee Tracker" is an application that allows employers to keep track of their employee information and data in an AWS cloud system. It can be tailored to whatever certain companies need. For example if a new manager is working at a company, he can use the app to quickly learn employee's names and faces while also being able to find out relevant information about them like their position in a fast, efficient manner.



# Summary

The app will utilize multiple Cloud based applications to deliver its services in a concise manner. Current the app using the following services

**AWS Amplify**

AWS Amplify helps with app development that can easily be integrated with AWS services. It is a set of libraries, components and CLI's that the majority of our application backend is built on..

**AWS Cognito**

Cognito is used on the backend for user authentication. All basic user information is stored on it and it allows users to log in with alternative platforms (e.g. Google). It allows us to add user sign up, sign ins and access control to our app easily

**AWS AppSync**

AppSync is a service that allows the creation of flexible API to securely access and combine data from multiple data sources. It helped us build a scalable application and assists with functionality with AWS Lambda

**AWS Lambda**

Lambda will be used to help run code from the cloud on the backend of the application. It will be used to communicate and interact with the other AWS services and connect to the API.

**AWS DynamoDB**

DynamoDB is a solid document database that delivers incredible performance at any scale. It works together with Lambda to retrieve information and then returns the data back to the user.

**AWS S3**

Amazon Simple storage service is a scalable object storage system. It lets us manage objects at scale through API requests and store & retrieve objects for the application

**Android Sdk**

Used in development of our mobile app which is written in java.

# App Interactions

1.AWS Cognito is used to authenticate users via a predefined policy which is assigned to Authenticated users, so in this way only authenticated users will be able to access(read/write/delete) data available on our application.

2.  AWS Amplify integrates all the AWS clients into the app

3. AWS S3 is used to store large files like images for our application.

4. Appsync helps Amplify connect to Lambda and DynamoDB

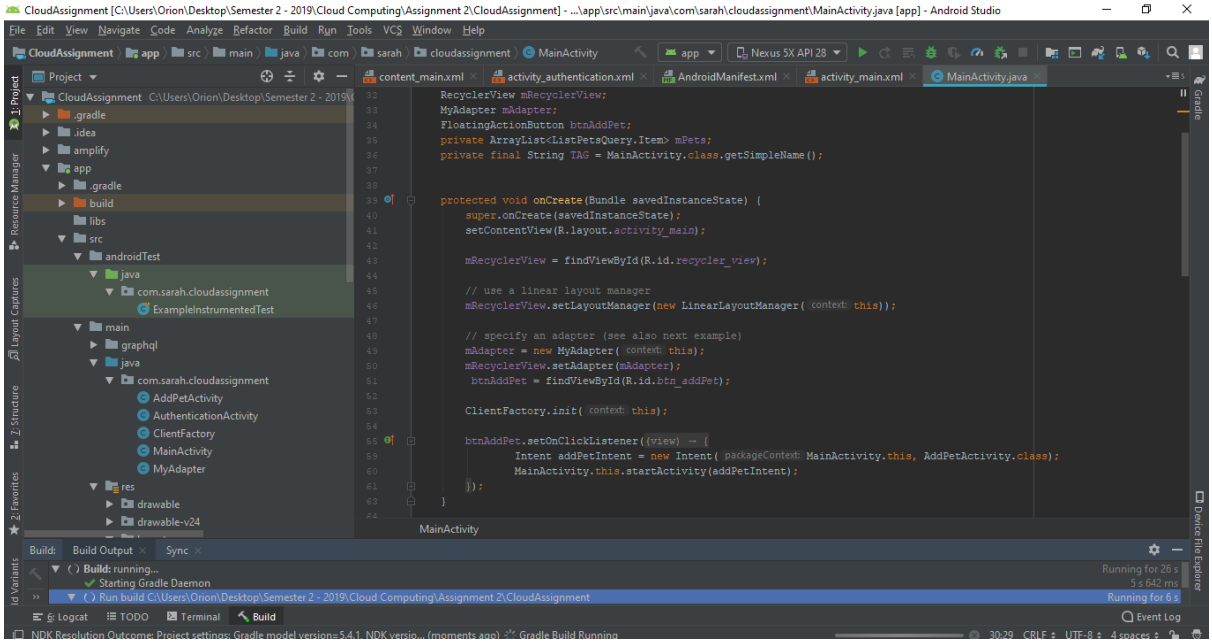
5. Lambda helps us run code for virtually any type of application/backend service with zero administration. We can set up our code to automatically trigger from other AWS services or call them directly from our mobile app.

6.DynamoDB is used to keep track of each employee data such as name,description, profile photo. All these data are stored in DynamoDB, but the images are stored in s3 so DynamoDB just keeps the name of each image assigned to each employee.

# Implementation

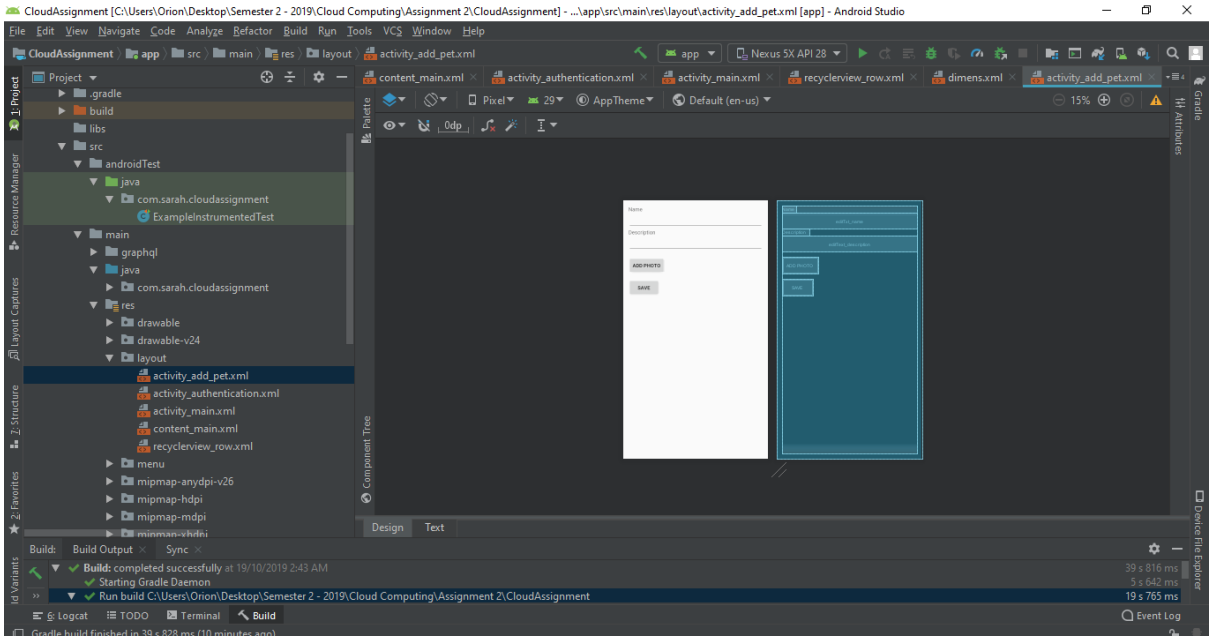
## Android Studio Code

-The majority of the app was coded in Android Studio using Java, It the application to use for android programming




## Android Studio Display

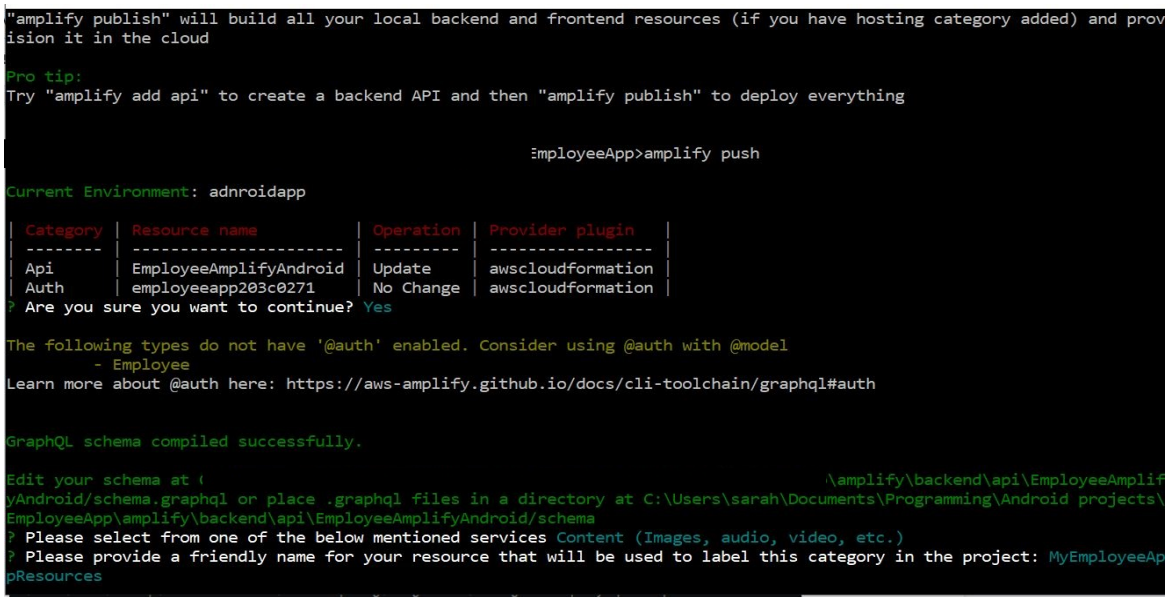
-Android studio shows users a preview of what the app will look like in its layout xml files



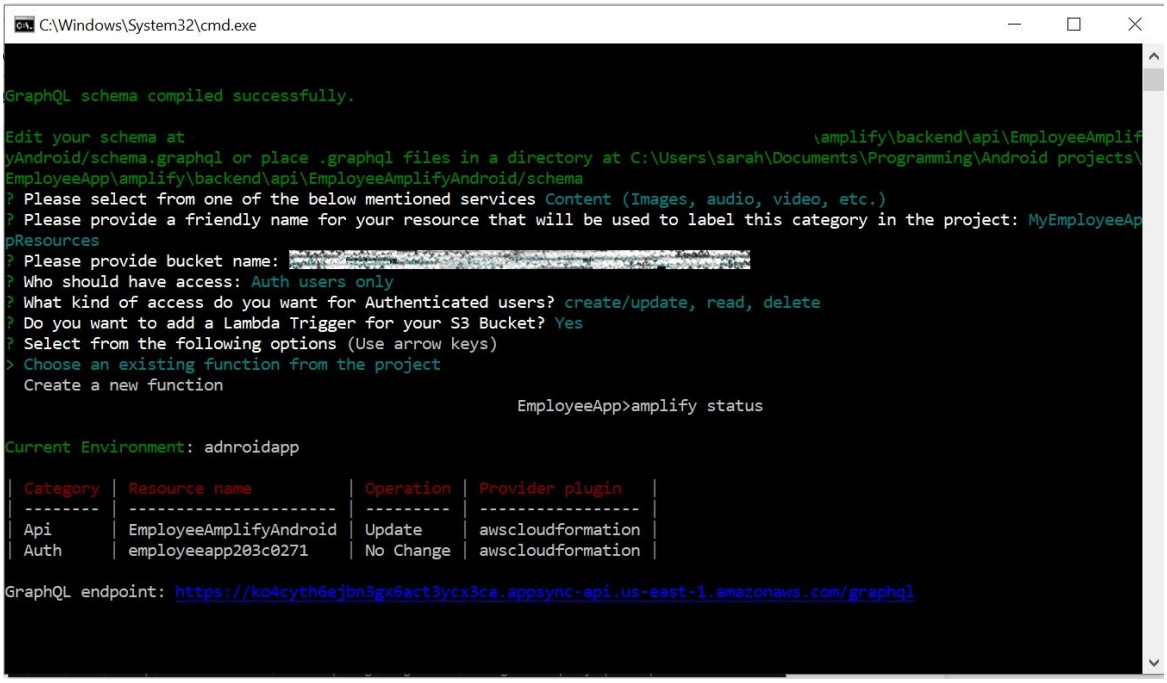
**Amplify CLI**

-Amplify is used through a command line interface and requires you to go to the root directory of your android app

-Amplify pushes data to the Cloud




-Viewing the status

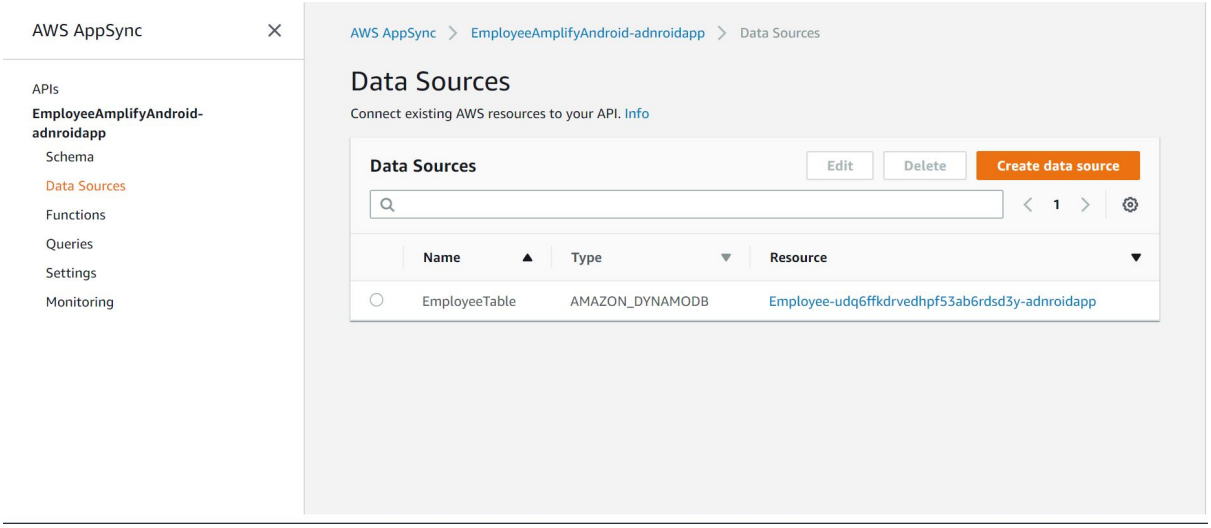


-Adding storage on the S3 bucket through Amplify

```
                                                \EmployeeApp>amplify add storage
? Please select from one of the below mentioned services Content (Images, audio, video, etc.)
? Please provide a friendly name for your resource that will be used to label this category in the project: MyEmployeeAp
pResources
? Please provide bucket name: employeeapp-adnroidapp-20191613151032-deployment
? Who should have access: Auth users only
? What kind of access do you want for Authenticated users? create/update, read, delete
? Do you want to add a Lambda Trigger for your S3 Bucket? No
Successfully added resource MyEmployeeAppResources locally

Some next steps:
"amplify push" builds all of your local backend resources and provisions them in the cloud
"amplify publish" builds all of your local backend and front-end resources (if you added hosting category) and provisio
s them in the cloud


                                                \EmployeeApp>amplify push

Current Environment: adnroidapp

| Category | Resource name          | Operation | Provider plugin   |
| -------- | ---------------------- | --------- | ----------------- |
| Storage  | MyEmployeeAppResources | Create    | awscloudformation |
| Api      | EmployeeAmplifyAndroid | Update    | awscloudformation |
| Auth     | employeeapp203c0271    | No Change | awscloudformation |
```

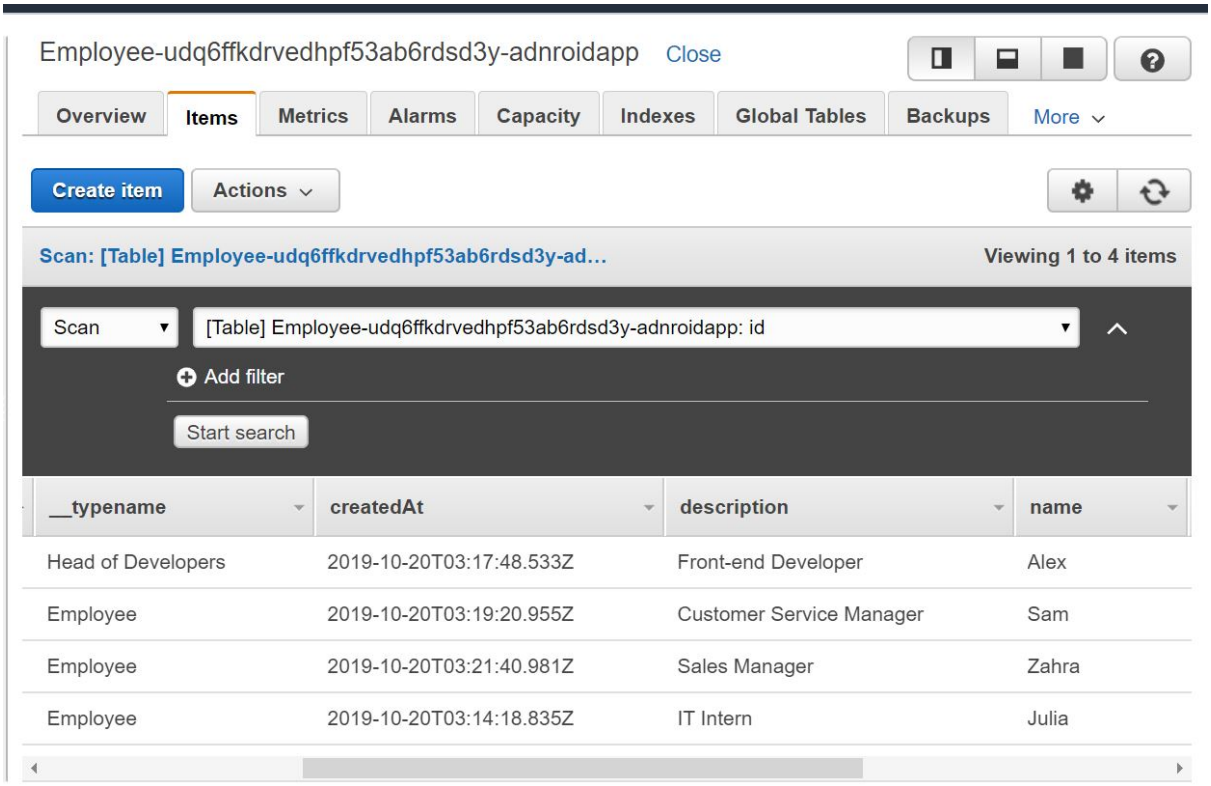
**AppSync:**

-Acknowledgement of the data source in AppSync



**DynamoDB:**

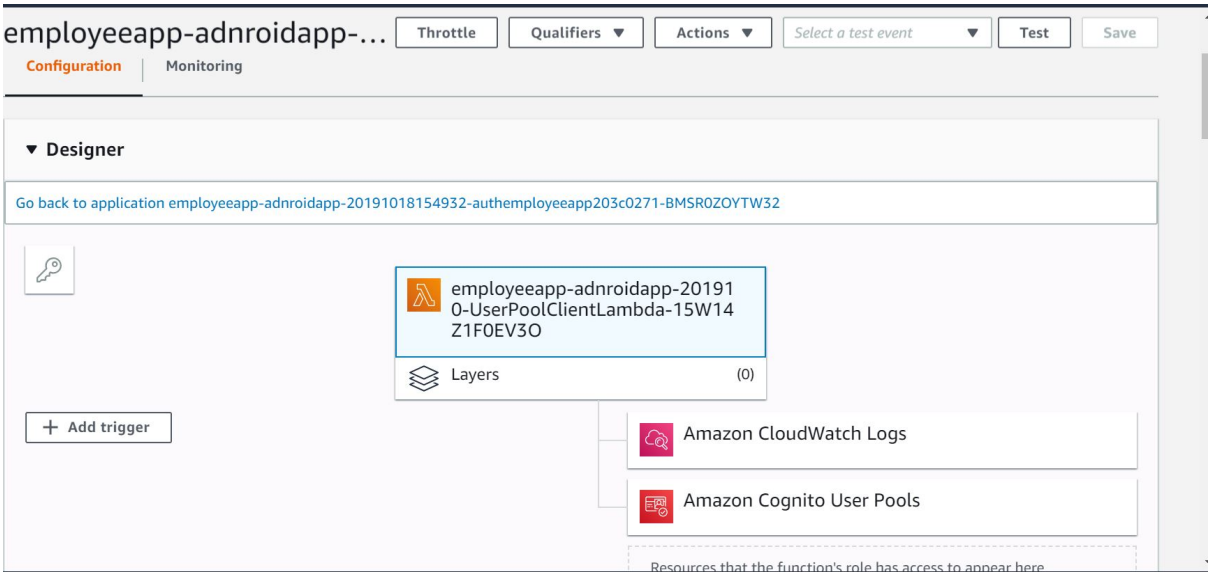
-Table of database in DynamoDB




| __typename | createdAt | description | name |
|---|---|---|---|
| Head of Developers | 2019-10-20T03:17:48.533Z | Front-end Developer | Alex |
| Employee | 2019-10-20T03:19:20.955Z | Customer Service Manager | Sam |
| Employee | 2019-10-20T03:21:40.981Z | Sales Manager | Zahra |
| Employee | 2019-10-20T03:14:18.835Z | IT Intern | Julia |

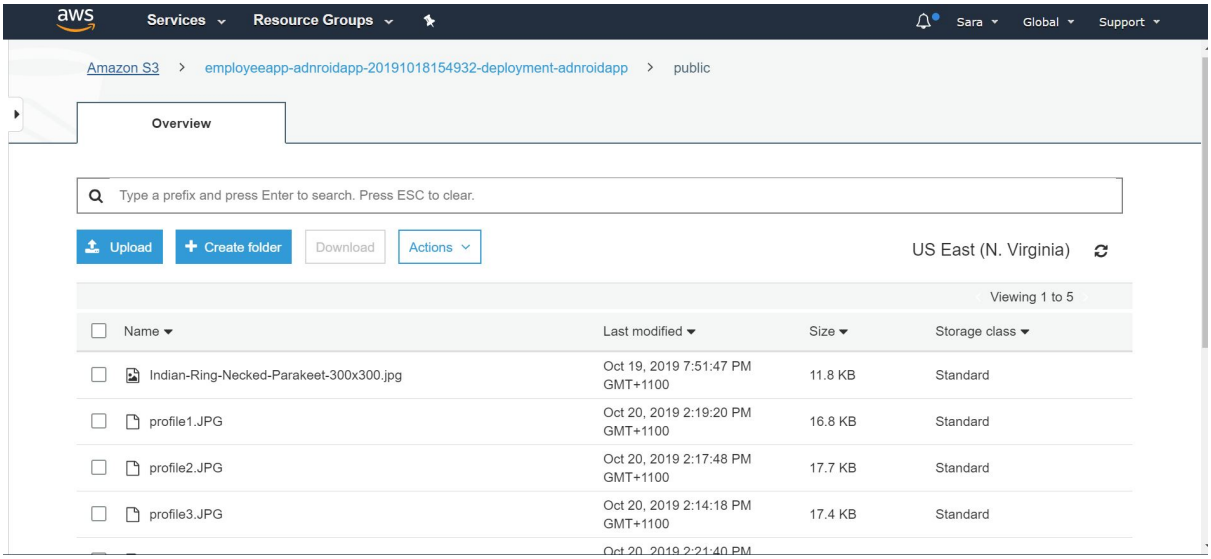
**Lambda:**

-User pool code executing in Lambda



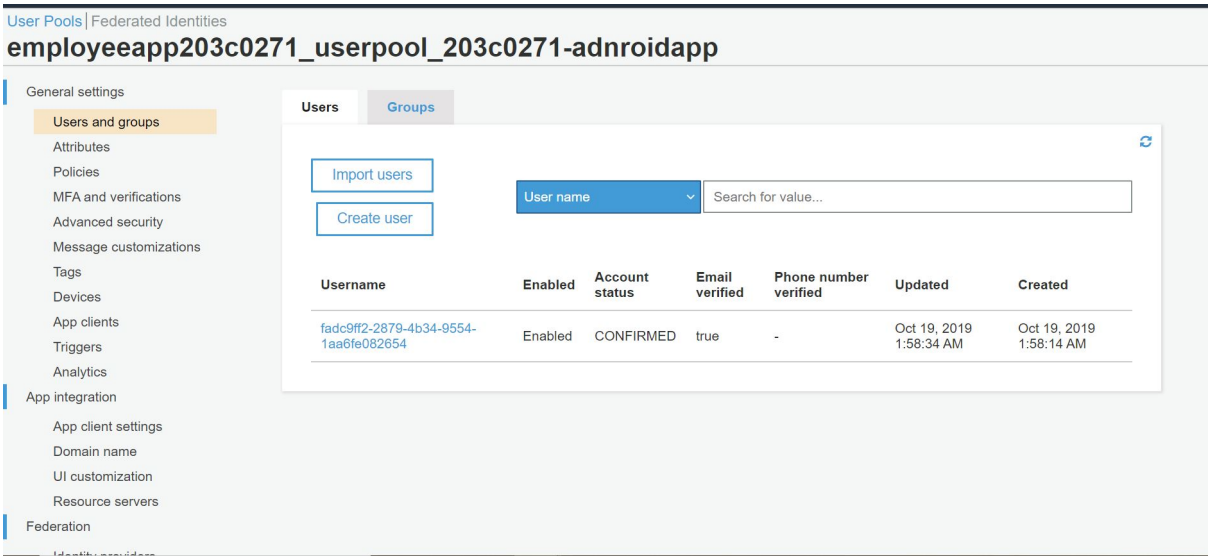
**S3:**

-Images from the upload stored in S3 bucket



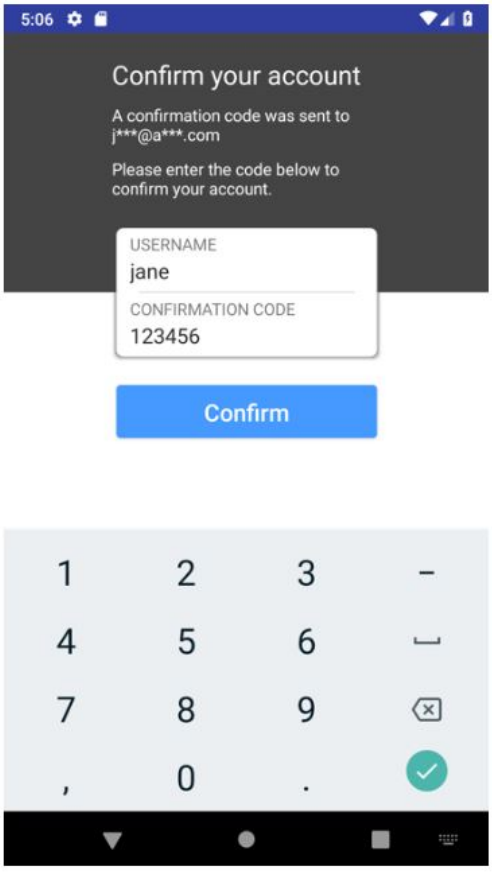

**Cognito:**

**-**User pool information generated by Cognito. Only authenticated users can edit and read the data in our mobile app.

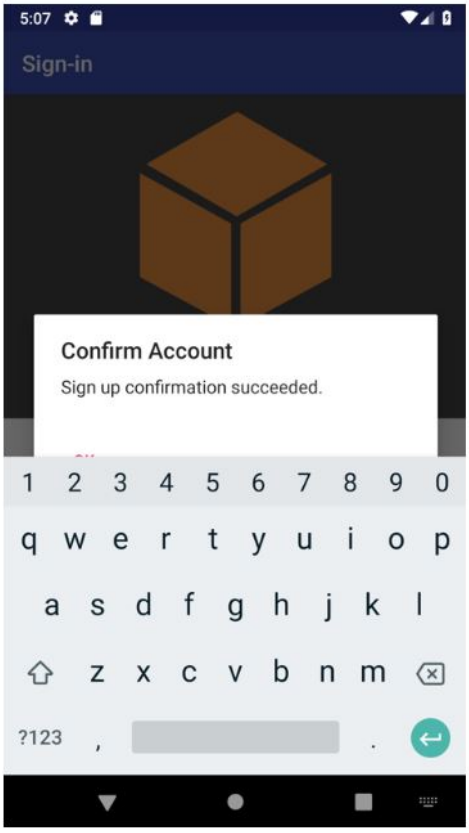


1. **Sign in page**

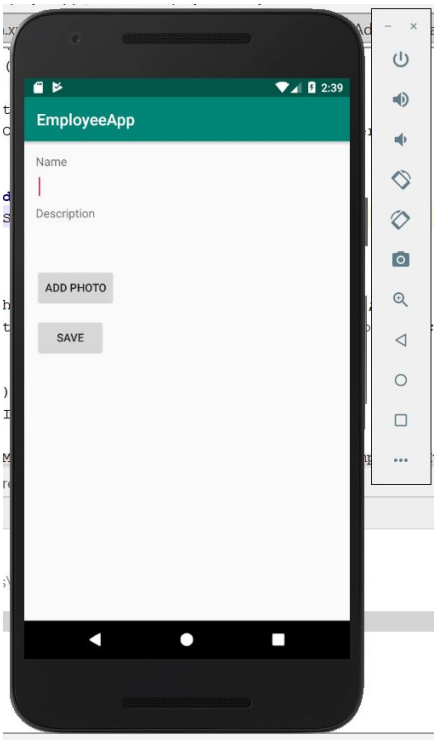

2. **Confirmation page**



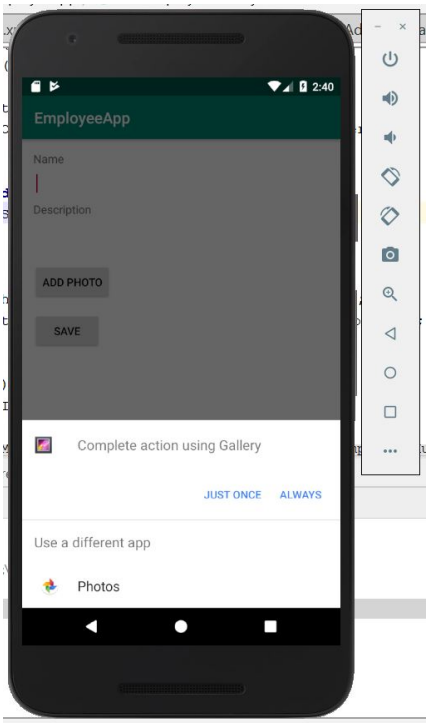
### 3. Confirm box



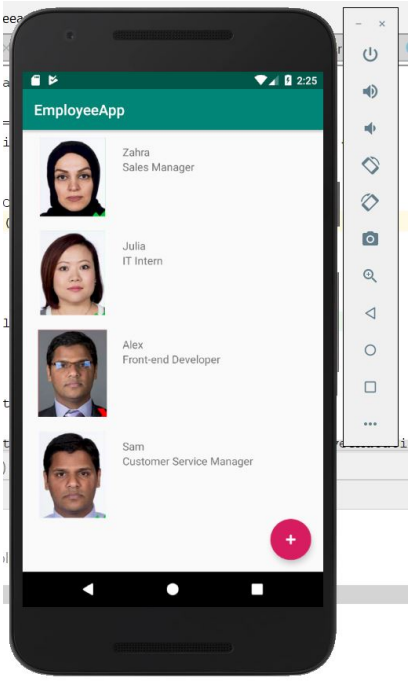

### 4. Input employee information



5. **Show prompt to add profile image**




6. **Selecting an image from local galley which followed by getting uploaded into s3 automatically right after the user clicks on save button. The picture name and its url in s3 will be saved in DynamoDB as well as user information.**



## References used


-Amazon Web Services, Inc.. 2019. AWS Amplify. [ONLINE] Available at: https://aws.amazon.com/amplify/. [Accessed 2 October 2019].

-Amazon Web Services, Inc.. 2019. AWS AppSync - Build data driven apps with real time and offline capabilities based on GraphQL. [ONLINE] Available at: https://aws.amazon.com/appsync/. [Accessed 2 October 2019].

-Amazon Web Services, Inc.. 2019. Amazon DynamoDB - Overview. [ONLINE] Available at: https://aws.amazon.com/dynamodb/. [Accessed 10 October 2019].

-Amazon Web Services, Inc.. 2019. Cloud Object Storage | Store & Retrieve Data Anywhere | Amazon Simple Storage Service. [ONLINE] Available at: https://aws.amazon.com/s3/. [Accessed 3 October 2019].

-Amazon Web Services, Inc.. 2019. AWS Lambda – Serverless Compute - Amazon Web Services. [ONLINE] Available at: https://aws.amazon.com/lambda/. [Accessed 7October 2019].

-Amazon Web Services, Inc.. 2019. Amazon Cognito - Simple and Secure User Sign Up & Sign In | Amazon Web Services (AWS). [ONLINE] Available at: https://aws.amazon.com/cognito/. [Accessed 7 October 2019].