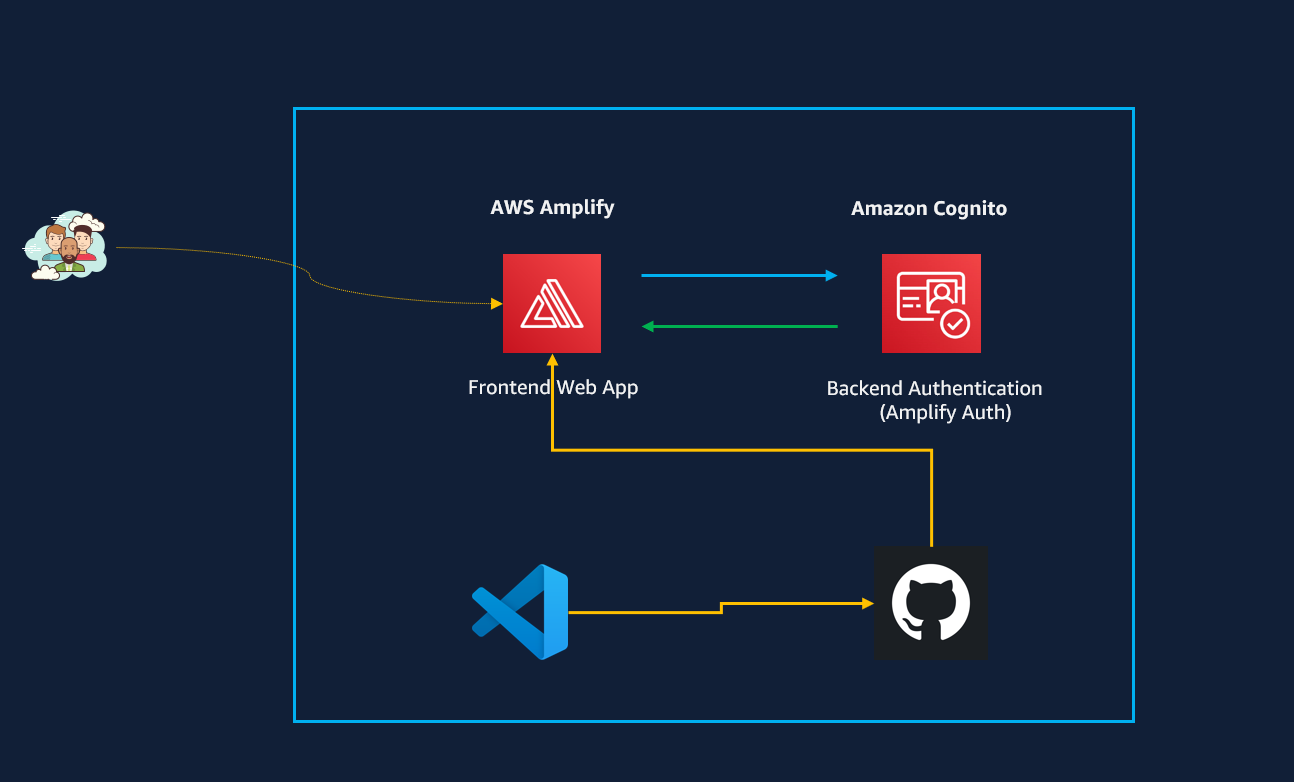
**Building a React App with Amplify (Gen 1), Cognito, and CI/CD with GitHub**



**prerequisites**

1 - AWS Account

2 - Code editor - IDE (example - VSCode)

3 - Installed on your environment: Nodejs and npm (https://nodejs.org/en/download/package-manager)

4 - Familiarity with AWS Management Console and IAM

5 – Github account

**Pricing**

Generally generous free tier on both Amazon Cognito and AWS Amplify (we will use both services)

See pricing below

AWS Amplify

https://aws.amazon.com/amplify/pricing/

A screenshot of a computer

Description automatically generated

Amazon Congnito (MAUs = Monthly Active Users)

https://aws.amazon.com/cognito/pricing/

A white rectangular object with black text

Description automatically generated

**1 – Set up environment**

* 1. – Install AWS Amplify CLI

On your Terminal issue the below command and press Enter

A black square with white text

Description automatically generated**npm install -g @aws-amplify/cli**

The AWS Amplify CLI is a command-line tool that enables you to set up and manage the backend resources for web and mobile applications directly from your local environment. With the CLI, you can easily configure and deploy various AWS services to support their application’s backend, without needing to go into the AWS Management Console

* 1. – Configure Amplify

1. On your Terminal issue the below command and press Enter

**amplify configure**

The amplify configure command in AWS Amplify CLI is used to set up your AWS credentials so that the Amplify CLI can access your AWS account to manage resources. It connects your local CLI environment to your AWS account.

A screenshot of a computer program

Description automatically generated

This will launch a browser to the AWS Console. (Sign in to AWS Console with Administrator)

On your terminal press Enter and choose your AWS Region (up and down arrow keys).

An AWS Amplify documentation will open and follow the steps to create IAM user with programmatic Access (Access Key ID and Secrete Access Key)

<https://docs.amplify.aws/gen1/react/prev/start/project-setup/prerequisites/>

* Once you created the IAM User, return to your terminal and press Enter.

Enter Access Key ID, Secrete Access Key and provide a Profile Name for your local machine

A screenshot of a computer program

Description automatically generated

**2 – Create the React App**

On your Terminal issue the below command and press Enter

* **npx create-react-app <name of your app> (I am calling mine my-quiz-app)**

You will have your project directories and files on your IDE.

A screenshot of a computer

Description automatically generated

Change to the root director of your APP (in my case my-quiz-app)

**cd <your app name>**

Now let’s initialize the Amplify Project within the React App. (We will get a series of questions here)

Make sure you are in the correct directory and issue the following command on your Terminal

**amplify init**

Answer questions like name of your App in AWS cloud

Next press y to go with the default configuration

Press Enter for the next prompts

Enter “no” for “Help Improve Amplify…” to save time for this project

1. **– Add Authentication with Cognito**

Go the AWS Console and navigate to AWS Amplify. Choose the name of your app (Webapp in my case I went with the default suggestion) click on your app

A screenshot of a computer

Description automatically generated

An environment has been created for us (dev in my case)

A screenshot of a computer

Description automatically generated

Back to your terminal and issue the below command to add authentication

**amplify add auth**

You will get prompted for authentication and security configuration – choose “Default Configuration” and press Enter. This option will give us Cognito

A screen shot of a computer

Description automatically generated

Next, you will be prompted how you want your users to sign in – “Choose email” and press enter

Next choose “no I am done” and press Enter.

A screen shot of a computer

Description automatically generated

Next – Let’s push all the configurations to AWS by issuing the below command on your Terminal.

**amplify push**

type “y” to continue…

\*\*\* Wait while all the configurations are pushed to AWS and Cognito User Pool is being created for us\*\*\*

Go to the AWS console and confirm that a new Cognito User Pool has been created.

**NEXT**

Under the **src** directory, open the App.js file on your IDE and replace everything with the below code – This will enable us to be able to create user and login to the webapp. (We have commented out the quiz capability on line 20 and line 33).

import React from 'react';

import './App.css';

// Imports the Amplify library from 'aws-amplify' package. This is used to configure your app to interact with AWS services.

import {Amplify} from 'aws-amplify';

// Imports the Authenticator and withAuthenticator components from '@aws-amplify/ui-react'.

// Authenticator is a React component that provides a ready-to-use sign-in and sign-up UI.

// withAuthenticator is a higher-order component that wraps your app component to enforce authentication.

import { Authenticator, withAuthenticator } from '@aws-amplify/ui-react';

// Imports the default styles for the Amplify UI components. This line ensures that the authenticator looks nice out of the box.

import '@aws-amplify/ui-react/styles.css';

// Imports the awsExports configuration file generated by the Amplify CLI. This file contains the AWS service configurations (like Cognito, AppSync, etc.) specific to your project.

import awsExports from './aws-exports';

// Imports the Quiz component from Quiz.js for use in this file.

import Quiz from './Quiz';

// Configures the Amplify library with the settings from aws-exports.js, which includes all the AWS service configurations for this project.

Amplify.configure(awsExports);

function App() {

return (

<div className="App">

<Authenticator>

{({ signOut }) => (

<main>

<header className='App-header'>

{/\* Quiz Component \*/}

<Quiz />\*/

{/\* Sign Out Button \*/}

<button

onClick={signOut}

style={{

margin: '20px',

fontSize: '0.8rem',

padding: '5px 10px',

marginTop: '20px'

}}

>

Sign Out

</button>

</header>

</main>

)}

</Authenticator>

</div>

);

}

export default withAuthenticator(App);

Now let’s install the Amplify Libraries that will give us the **login interface**. Issue the following command to your terminal and press enter.

**npm install aws-amplify @aws-amplify/ui-react**

Now let’s start the app by typing the following command on the terminal

**npm start**

A sign-in page will be launched for us on local-host port 3000

A screenshot of a login form

Description automatically generated

Create an account and sign in. You will need to use a valid email address to get verification code.

Because we have commented out the Quiz capabilities we will only get a sign out button after logging in. This is what is expected.

In Cognito user-pool, you will be able to see that a user has been created for you.

1. **- Add functionality and styling for the Quiz**

Under the Src directory, create a new file and name it “Quiz.js”

Paste the below code in to the file you just created.

import React, { useState } from 'react';

import quizData from './quizData';

function Quiz() {

const [currentQuestion, setCurrentQuestion] = useState(0);

const [score, setScore] = useState(0);

const [showScore, setShowScore] = useState(false);

const [selectedAnswer, setSelectedAnswer] = useState("");

const [isCorrect, setIsCorrect] = useState(null);

const handleAnswerOptionClick = (option) => {

const correctAnswer = quizData[currentQuestion].answer;

setSelectedAnswer(option);

if (option === correctAnswer) {

setScore(score + 1);

setIsCorrect(true);

} else {

setIsCorrect(false);

}

// Delay moving to the next question to allow the user to see feedback

setTimeout(() => {

const nextQuestion = currentQuestion + 1;

if (nextQuestion < quizData.length) {

setCurrentQuestion(nextQuestion);

setIsCorrect(null); // Reset for the next question

setSelectedAnswer(""); // Reset selected answer

} else {

setShowScore(true);

}

}, 1000); // Adjust time as needed

};

return (

<div className='quiz'>

{showScore ? (

<div className='score-section'>

You scored {score} out of {quizData.length}

</div>

) : (

<>

<div className='question-section'>

<div className='question-count'>

<span>Question {currentQuestion + 1}</span>/{quizData.length}

</div>

<div className='question-text'>{quizData[currentQuestion].question}</div>

</div>

<div className='answer-section'>

{quizData[currentQuestion].options.map((option) => (

<button

onClick={() => handleAnswerOptionClick(option)}

key={option}

style={{ backgroundColor: selectedAnswer === option ? (isCorrect ? 'lightgreen' : 'pink') : '' }}

>

{option}

</button>

))}

</div>

{selectedAnswer && (

<div style={{ marginTop: '10px' }}>

{isCorrect ? 'Correct! 🎉' : 'Sorry, that’s not right. 😢'}

</div>

)}

</>

)}

</div>

);

}

export default Quiz;

Save the file.

Create a second file on Src directory and name it “quizdata.js”

Paste the below commands to the file (quizdata.js) and save the file.

const quizData = [

{

question: "What was the first video game ever made?",

options: ["Pong", "Spacewar!", "Tetris", "Computer Space"],

answer: "Spacewar!"

},

{

question: "Which company developed the first commercial antivirus software?",

options: ["Symantec", "McAfee", "Norton", "Kaspersky Lab"],

answer: "McAfee"

},

{

question: "Which animal is featured in the official PHP logo?",

options: ["Elephant", "Hippo", "Giraffe", "Lion"],

answer: "Elephant"

},

{

question: "What does 'HTTP' stand for?",

options: ["HyperText Transfer Protocol", "Hyperlink Transfer Technology Protocol", "Hyperlink Text Transfer Protocol", "HyperText Technology Protocol"],

answer: "HyperText Transfer Protocol"

},

{

question: "Which programming language is known as the backbone of the World Wide Web?",

options: ["Java", "C#", "Python", "HTML"],

answer: "HTML"

},

{

question: "What is the name of the world's first computer programmer?",

options: ["Charles Babbage", "Ada Lovelace", "Alan Turing", "Grace Hopper"],

answer: "Ada Lovelace"

},

{

question: "In what year was the iPhone first introduced?",

options: ["2005", "2007", "2009", "2011"],

answer: "2007"

},

{

question: "What was Google's original name?",

options: ["BackRub", "Googol", "SearchMaster", "WebSearch"],

answer: "BackRub"

},

{

question: "Which of these companies was not founded in a garage?",

options: ["Amazon", "Google", "Apple", "Microsoft"],

answer: "Amazon"

},

{

question: "What does 'GPU' stand for?",

options: ["Graphical Processing Unit", "Graphics Performance Unit", "Graphics Processing Unit", "Graphical Performance Unit"],

answer: "Graphics Processing Unit"

},

{

question: "What is the capital of France?",

options: ["New York", "London", "Paris", "Dublin"],

answer: "Paris"

},

{

question: "Who painted the Mona Lisa?",

options: ["Vincent Van Gogh", "Leonardo da Vinci", "Pablo Picasso", "Claude Monet"],

answer: "Leonardo da Vinci"

},

{

question: "What is the largest planet in our solar system?",

options: ["Earth", "Jupiter", "Saturn", "Mars"],

answer: "Jupiter"

}

];

export default quizData;

Below is the code from App.js file where I have uncommented lines 20 and 33. Either uncomment those lines or copy the entire code below and replace the App.js

import React from 'react';

import './App.css';

// Imports the Amplify library from 'aws-amplify' package. This is used to configure your app to interact with AWS services.

import {Amplify} from 'aws-amplify';

// Imports the Authenticator and withAuthenticator components from '@aws-amplify/ui-react'.

// Authenticator is a React component that provides a ready-to-use sign-in and sign-up UI.

// withAuthenticator is a higher-order component that wraps your app component to enforce authentication.

import { Authenticator, withAuthenticator } from '@aws-amplify/ui-react';

// Imports the default styles for the Amplify UI components. This line ensures that the authenticator looks nice out of the box.

import '@aws-amplify/ui-react/styles.css';

// Imports the awsExports configuration file generated by the Amplify CLI. This file contains the AWS service configurations (like Cognito, AppSync, etc.) specific to your project.

import awsExports from './aws-exports';

// Imports the Quiz component from Quiz.js for use in this file.

import Quiz from './Quiz';

// Configures the Amplify library with the settings from aws-exports.js, which includes all the AWS service configurations for this project.

Amplify.configure(awsExports);

function App() {

return (

<div className="App">

<Authenticator>

{({ signOut }) => (

<main>

<header className='App-header'>

{/\* Quiz Component \*/}

<Quiz />\*/

{/\* Sign Out Button \*/}

<button

onClick={signOut}

style={{

margin: '20px',

fontSize: '0.8rem',

padding: '5px 10px',

marginTop: '20px'

}}

>

Sign Out

</button>

</header>

</main>

)}

</Authenticator>

</div>

);

}

export default withAuthenticator(App);

Save App.js file.

Note – Up to this point we are running our frontend from local host. Now it’s time to push all the code to Github and create a CI/CD.

1. **– Push Local Code to Github**

Create a repository in your Github – name it whatever you want.

On your terminal issue the following commands to push your code to Github

**git init**

**git add .**

**git commit -m “First commit”**

**git branch -M main**

git remote add origin [https://github.com/<YOUR](https://github.com/%3cYOUR) GITHUB ACCOUNT NAME>/<YOUR REPO>

\*\*\*When you create your repository, it will display the required commands\*\*\*

**git push -u origin main**

Now your local code must be on Github repository you created.

1. **– Host the Frontend in Amplify Via GitHub (CI/CD)**

Navigate to AWS Amplify from the AWS Console. On the Branches tab choose “Get Started”

A screenshot of a chat

Description automatically generated

Choose GitHub and click “Next”

Click on the search bar and find the repository you created for this project. If you are unable to find it, “Update GitHub permissions”. A new GitHub browser will popup and add the repository.

Click Next.

Scroll down and find “**Select a backend environment to use with this branch”**

Select the backend App Name and choose the correct Environment.

**A screenshot of a computer

Description automatically generated**

Make sure “**Full-stack deploys**” is enabled.

Amplify will need a Service Role to host our App. Create a new role if you don’t already have one.

(AWS Service Role for Amplify) and choose that role. (amplifyconsole-backend-role).

Click “Next” and “Save and Deploy”.

Wait for your App to be deployed on Amplify.

Once done, click the URL Domain to visit your App.

A screenshot of a computer

Description automatically generated

To confirm CICD is working, go to your GitHub repo and update quizData.js (edit in place) under the Src directory by removing the first question, commit changes and check your App.

A screenshot of a computer

Description automatically generated

Amplify will detect the changes and moves to building and deploying.

1. **– Clean up**

\*\*\*\*\*Follow Instructions to cleanup everything\*\*\*\*\*

A screenshot of a computer

Description automatically generated

Confirm by typing “delete” on the dialog box.

Once you delete the Amplify backend, Cognito should be deleted as well.

You can delete your IAM user as well.

-------------------------------------------END----------------------------------------