Application Front End

**SPL-BE-200-DVAPFE-1 - Version 1.0.2**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

In this lab, after deploying the serverless backend using AWS SAM and API Gateway, you use AWS Amplify to deploy the frontend of the grid-maker application and configure it to authenticate using Amazon Cognito.

AWS Amplify is a development platform that provides a set of tools and services for building scalable and secure mobile and web applications. In this lab, you will use Amplify to deploy the frontend of the grid-maker application, which is a web application that allows users to interact with the serverless backend. Amplify simplifies the deployment process by providing easy-to-use tools and services that automate tasks like building, testing, and deploying applications.

To authenticate users, the lab uses Amazon Cognito, a managed authentication service that allows you to easily add user sign-up, sign-in, and access control to your applications. Cognito provides several features, including user pools, identity pools, and multi-factor authentication, that allow you to secure your application and control access to resources.

During the lab, you configure Amplify to use Cognito for authentication and authorization. You create a user pool in Cognito, which stores user accounts and user attributes, and an identity pool, which provides temporary AWS credentials to users. You then integrate Amplify with Cognito and configure the frontend to use the Cognito user pool for authentication.

By the end of the lab, you have a complete serverless application that includes a frontend deployed using Amplify and a backend deployed using SAM and API Gateway. You gain an understanding of how to use Amplify to deploy frontend applications and how to use Cognito for authentication and authorization.

OBJECTIVES

By the end of this lab, you will be able to do the following:

* Install requirements to use AWS Amplify CLI.
* Initialize AWS Amplify and deploy a project.
* Configure Cognito to authenticate with the application.
* Run the grid-maker application using authentication to create the grid image.

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab:

* Familiarity with the basic navigation of the AWS Management Console.
* Versed in editing and running scripts using an AWS Cloud9 code editor and terminal.
* A basic understanding and familiarity with Amazon API Gateway, AWS Serverless Application Model (SAM), AWS Lambda, AWS CloudFormation, and AWS Amplify.
* Prior experience with AWS services including AWS Amplify and serverless computing will be helpful but is not necessarily required.

DURATION

This lab requires *60* minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A hint, tip, or important guidance.
* **Consider:** A moment to pause to consider how you might apply a concept in your own environment or to initiate a conversation about the topic at hand.
* **Task complete:** A conclusion or summary point in the lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**WARNING:** **Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Install and initialize AWS Amplify**

In this task, you connect to the AWS Cloud9 environment to install the Amplify CLI and its dependencies globally. Once installed you initialize Amplify. This will create all of the files and resources to deploy this project.

Amplify CLI is a command-line interface tool that you can use to configure and deploy your application. To install the Amplify CLI, you need to have *Node.js*, *npm* (Node.js Package Manager), and the Amplify dependencies installed. Node.js and npm are already installed.

You run the *npm install* from the ~/python-image-grid/react-amplified directory and npm dependencies are installed based on the *package.json* file.

TASK 1.1: CONNECT TO THE AWS CLOUD9 ENVIRONMENT

AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal. AWS Cloud9 comes prepackaged with essential tools for popular programming languages including JavaScript, Python, PHP, and more. You do not need to install files or configure your development machine to start new projects.

In this task, you connect to the AWS Cloud9 environment provisioned as part of this lab.

1. Copy the **Cloud9Environment** URL link from the **Lab Information** section to the left of these instructions and paste it into a new browser tab. The browser takes you to the AWS Cloud9 environment that you use during this lab.
2. You do not need the **Cloud9 Welcome screen** or any of the other default tabs that appear when you first launch **Cloud9**, so choose the **x** next to each tab to close them. This section is where you update various file throughout this lab.

**Consider:** Take a moment to familiarize yourself with the **AWS Cloud9** IDE interface.

* In the middle of the screen, a single terminal session is open in the editor. You can open multiple tabs in this window to edit files and run terminal commands.
* The file navigator appears on the left side of the screen. As you build out your CDK environment and application, additional directories and files appear here.
* A gear icon appears on the right side of the screen. Choosing this icon opens the AWS Cloud9 Settings panel.

Every *AWS Cloud9* workspace is automatically assigned *IAM* credentials that provide it with limited access to some AWS services in your account based on your federated role. We call these AWS managed temporary credentials.

TASK 1.2: INSTALL NPM AND THE AMPLIFY CLI

1. **Command:** Use *npm* to install the *Amplify CLI dependencies* with the following commands to :

cd ~/environment/react-amplified/

npm install

**Expected output:** Output has been truncated. Your values may differ slightly.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

added 2799 packages, and audited 2800 packages in 47s

**Note:** All deprecation warning messages can be ignored.

1. **Command:** With the Amplify CLI dependencies installed, you can now install the Amplify CLI globally with the following command:

npm install -g @aws-amplify/cli

**Expected output:** Your values may differ slightly.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

added 26 packages in 12s

TASK 1.3: INITIALIZE AMPLIFY

Next, you initialize Amplify. During this step it creates all of the files and resources required to deploy this project.

1. **Command:** To initialize Amplify, run the following *amplify init* command:

**Caution:** The following commands create a variable for the Permissions Boundary ARN to help further secure the lab as Amplify has to create a few roles.

cd /home/ec2-user/environment/react-amplified

boundary\_policy=$(aws iam list-policies --query 'Policies[?PolicyName==`AmplifyPermissionsBoundary`].Arn' --output text)

amplify init --permissions-boundary $boundary\_policy

1. You will be asked a series of questions. Use the following answers:

* For **Enter a name for the project (reactamplified)**, press **Enter** to use the default value.
* For **Initialize the project with the above configuration?**, accept default value for

Yes

.

* For **Select the authentication method you want to use:**, choose

AWS profile

.

* For **Please choose the profile you want to use**, choose

default

.

**Note:** Should you setup Amplify in your own environment. Please know that Amplify needs something in the *~/.aws/config* file to be able to find and use the *default profile* in the *credentials* file that AWS Cloud9 configures with a temporary access key and secret access key. When bootstrapping the AWS Cloud9 environment, the *config* file was created to include the *output* format as *json*. Otherwise, it does not find it and asks if you want to create a user for this step.

It now initializes the Amplify project and create the required AWS resources.

* **Command:** For **Help improve Amplify CLI by sharing non sensitive configurations on failures (y/N)**, accept the default selection of **N**.

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Adding backend environment dev to AWS Amplify app: d3sogr4oscggoz

Deployment completed.

Deploying root stack reactamplified [ ==========------------------------------ ] 1/4

amplify-reactamplified-dev-18… AWS::CloudFormation::Stack CREATE\_IN\_PROGRESS Mon Mar 20 2023 18:32:26…

AuthRole AWS::IAM::Role CREATE\_COMPLETE Mon Mar 20 2023 18:32:46…

UnauthRole AWS::IAM::Role CREATE\_IN\_PROGRESS Mon Mar 20 2023 18:32:30…

DeploymentBucket AWS::S3::Bucket CREATE\_IN\_PROGRESS Mon Mar 20 2023 18:32:30…

✔ Help improve Amplify CLI by sharing non sensitive configurations on failures (y/N) · no

Deployment state saved successfully.

✔ Initialized provider successfully.

✅ Initialized your environment successfully.

Your project has been successfully initialized and connected to the cloud!

Some next steps:

"amplify status" will show you what you've added already and if it's locally configured or deployed

"amplify add <category>" will allow you to add features like user login or a backend API

"amplify push" will build all your local backend resources and provision it in the cloud

"amplify console" to open the Amplify Console and view your project status

"amplify publish" will build all your local backend and frontend resources (if you have hosting category added) and provision it in the cloud

Pro tip:

Try "amplify add api" to create a backend API and then "amplify push" to deploy everything

**Note:** If you get an error stating, *amplify-reactamplified-dev-######-deployment already exists.* Run the command again to allow a new bucket name to be created. The *6* digits are generated at random and it is possible to run into duplicates.

Once it finishes, it shows you a list of commands you can now perform.

**Task complete:** You have successfully installed npm, the Amplify CLI and initialized the Amplify project.

**Task 2: Add hosting and authorization to the Amplify project**

In this task, you add *hosting* to the Amplify project. The command *amplify add hosting* adds the hosting resources for the application backend. Once host has been added you publish the project and verify the application is reachable. Then you add authorization to the project using Amazon Cognito.

TASK 2.1: ADD HOSTING TO THE AMPLIFY PROJECT

Add *hosting* via the console so the application can be accessed using a URL.

1. **Command:** To enable hosting, run the following command:

amplify add hosting

1. It asks you which hosting option to use. Choose **Hosting with Amplify Console (Managed hosting with custom domains, Continuous deployment)**.
2. For **Choose a type**, select **Manual deployment**.

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

✔ Select the plugin module to execute · Hosting with Amplify Console (Managed hosting with custom domains, Continuous deployment)

? Choose a type Manual deployment

You can now publish your app using the following command:

Command: amplify publish

It now indicates you are able to publish your project.

TASK 2.2: PUBLISH THE AMPLIFY PROJECT

The command *amplify publish* is designed to build and publish both the backend and frontend of the project. After running *amplify publish --yes*, your app is available online and your terminal displays your app URL hosted on an *amplifyapp.com* domain. Whenever you have additional changes to publish, just re-run the amplify publish command.

1. **Command:** To publish the project, run the following **amplify** command:

amplify publish --yes

**Note:** It does take a few minutes for the publishing process to finish.

While it deploys the resources in the AWS environment, open the Amplify console to see what is taking place.

1. At the top of the AWS Management Console, in the search bar, search for and choose

AWS Amplify

. Open it in a new browser tab.

1. Once you open the Amplify console, choose the **reactamplified** app link.
2. Under the **Hosting environments** tab you should see a status message reading, “**Waiting to deploy** Your deployment is being queued…”

**Note:** There is a *Backend environments* feature, but you are not using it in this lab. In this lab, you are focusing on using Amplify for *frontend hosting* and then Amplify for setting up *Cognito for authentication*.

1. Switch back to the **AWS Cloud9** browser tab and wait for the publishing to finish.

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

✔ Select the plugin module to execute · Hosting with Amplify Console (Managed hosting with custom domains, Continuous deployment)

? Choose a type Manual deployment

You can now publish your app using the following command:

Command: amplify publish

LabUser:~/environment/react-amplified $ amplify publish

✔ Successfully pulled backend environment dev from the cloud.

Current Environment: dev

┌──────────┬────────────────┬───────────┬───────────────────┐

│ Category │ Resource name │ Operation │ Provider plugin │

├──────────┼────────────────┼───────────┼───────────────────┤

│ Hosting │ amplifyhosting │ Create │ awscloudformation │

└──────────┴────────────────┴───────────┴───────────────────┘

✔ Are you sure you want to continue? (Y/n) · yes

Deployment completed.

Deployed root stack reactamplified [ ======================================== ] 2/2

amplify-reactamplified-dev-18… AWS::CloudFormation::Stack UPDATE\_COMPLETE Mon Mar 20 2023 19:59:16…

hostingamplifyhosting AWS::CloudFormation::Stack CREATE\_COMPLETE Mon Mar 20 2023 19:59:13…

Deployed hosting amplifyhosting [ ======================================== ] 1/1

Deployment state saved successfully.

Publish started for amplifyhosting

> react-amplified@0.1.0 build

> react-scripts build

Creating an optimized production build...

Compiled with warnings.

[eslint]

src/App.js

Line 19:20: 'setUniqueId' is assigned a value but never used no-unused-vars

Search for the keywords to learn more about each warning.

To ignore, add // eslint-disable-next-line to the line before.

File sizes after gzip:

251 kB build/static/js/main.dca75ce3.js

33.44 kB build/static/css/main.a3e3b07c.css

1.79 kB build/static/js/787.cbb326c1.chunk.js

The project was built assuming it is hosted at /.

You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.

You may serve it with a static server:

npm install -g serve

serve -s build

Find out more about deployment here:

https://cra.link/deployment

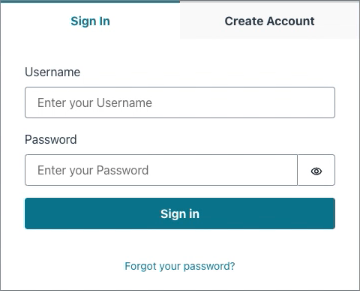
✔ Zipping artifacts completed.

✔ Deployment complete!

https://dev.d3sogr4oscggoz.amplifyapp.com

1. Once it finishes, you can switch back to the **Amplify** browser tab and see that it now shows that the **Deployment successfully completed.**. You can see the **Domain URL** is listed and that it matches the URL provided in the output in the **Cloud9** console.
2. Open the **URL** in a new browser tab.

The browser shows the application login screen. This uses Cognito to create an account with or to log in with a Cognito username if an account is already created. Cognito has not been configured yet, you configure Cognito in the next step.



*Image depicts the application login screen. It has a link to Sign In or Create Account. There is a field for a Username and Password followed by a Sign in button. It also provides a link if you forgot your password, to have it reset.*

TASK 2.3: ADD AUTHENTICATION TO THE AMPLIFY PROJECT

The *amplify add auth* command is used to add authentication to your app. It creates an Amazon Cognito user pool and identity pool for your app, and configures your app to use them. It offers authentication/authorization services and you use the options to sign-up and sign-in. This application does not use multi-factor authentication (MFA) but it is an option should you need that as a requirement in the future. You configure Cognito to require an email and password to sign up. It sends a confirmation email with a verification code to activate your login credentials.

1. Switch back to the **AWS Cloud9** browser tab.
2. **Command:** Add Cognito authentication by running the **amplify add auth** option for this Amplify project with the following command:

amplify add auth

You are asked a series of questions:

* For the question **Do you want to use the default authentication security configuration?**, choose **Manual configuration**.
* For **User Sign-Up, Sign-In**, chose **user Sign-Up & Sign-In only (Best used with a cloud API only)**.
* For **Provide a friendly name for your resource that will be used to label this category in the project:**, use the **default** value for friendly name for your resource.
* For **Provide a name for your user pool:**, use the **default** value.
* For **How do you want users to sign in**, choose **email**.
* For **Do you want to add User Pool Groups?**, choose **No**.

**Note:** We do not have different levels of access to the application. Everyone gets the same access.

* For **Do you want to add admin queries API?**, choose **No**.
* For **Multifactor authentication (MFA) user login options**, choose **OFF**.
* For E**mail based user registration/forgot password**, choose **Enabled (Requires per-user email entry at registration)**.
* For **Specify an email verification subject: (Your verification code)**, use the **default** value.
* For **Specify and email verification message: Your verification code is {####}**, use the **default** value.
* For **Do you want to override the default password policy for this User Pool?**, choose **No**.
* For **What attributes are required for signing up?**, choose **Email**.
* For **Specify the app’s refresh token expiration period (in days):**, use the default value of **30**.
* For **Do you want to specify the user attributes this app can read and write?**, use default value of **No**.
* For **Do you want to enable any of the following capabilities?**, press **Enter** to use the default value of **No**.
* For **Do you want to use an OAuth flow?**, choose **No**.
* For **Do you want to configure Lambda triggers?**, type

n

.

Now it has everything it needs to push the changes.

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

✅ Successfully added auth resource reactamplified4c64a58c4c64a58c locally

✅ Some next steps:

"amplify push" will build all your local backend resources and provision it in the cloud

"amplify publish" will build all your local backend and frontend resources (if you have hosting category added) and provision it in the cloud

1. **Command:** Push the changes to the project with the following command:

amplify push --yes

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

✅ Successfully added auth resource reactamplified4c64a58c4c64a58c locally

✅ Some next steps:

"amplify push" will build all your local backend resources and provision it in the cloud

"amplify publish" will build all your local backend and frontend resources (if you have hosting category added) and provision it in the cloud

LabUser:~/environment/react-amplified $ amplify push

✔ Successfully pulled backend environment dev from the cloud.

Current Environment: dev

┌──────────┬────────────────────────────────┬───────────┬───────────────────┐

│ Category │ Resource name │ Operation │ Provider plugin │

├──────────┼────────────────────────────────┼───────────┼───────────────────┤

│ Auth │ reactamplified4c64a58c4c64a58c │ Create │ awscloudformation │

├──────────┼────────────────────────────────┼───────────┼───────────────────┤

│ Hosting │ amplifyhosting │ No Change │ awscloudformation │

└──────────┴────────────────────────────────┴───────────┴───────────────────┘

✔ Are you sure you want to continue? (Y/n)

**Note:** This output shows that the project is going to create the Auth option and that there is no change to the **Hosting** option.

The deployment with the authentication updates start.

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Deployment completed.

Deploying root stack reactamplified [ ===========================------------- ] 2/3

amplify-reactamplified-dev-18… AWS::CloudFormation::Stack UPDATE\_IN\_PROGRESS Mon Mar 20 2023 20:30:12…

authreactamplified4c64a58c4c6… AWS::CloudFormation::Stack CREATE\_COMPLETE Mon Mar 20 2023 20:31:52…

hostingamplifyhosting AWS::CloudFormation::Stack UPDATE\_COMPLETE Mon Mar 20 2023 20:30:18…

Deployed auth reactamplified4c64a58c4c64a58c [ ======================================== ] 8/8

UserPool AWS::Cognito::UserPool CREATE\_COMPLETE Mon Mar 20 2023 20:30:26…

UserPoolClientWeb AWS::Cognito::UserPoolClient CREATE\_COMPLETE Mon Mar 20 2023 20:30:31…

UserPoolClient AWS::Cognito::UserPoolClient CREATE\_COMPLETE Mon Mar 20 2023 20:30:31…

UserPoolClientRole AWS::IAM::Role CREATE\_COMPLETE Mon Mar 20 2023 20:30:50…

UserPoolClientLambda AWS::Lambda::Function CREATE\_COMPLETE Mon Mar 20 2023 20:31:01…

UserPoolClientLambdaPolicy AWS::IAM::Policy CREATE\_COMPLETE Mon Mar 20 2023 20:31:20…

Deployment state saved successfully.

Once finished creating the *Cognito* resources and configuring the application, you need to update the *API* so it requires *authentication*.

**Task complete:** You have added hosting and authorization to the Amplify project.

**Task 3: Configure the application to require authentication**

With the Cognito resources in-place, you now configure the application to use Cognito to authenticate existing users and allow new users to sign up. Then you use the SAM CLI to re-deploy the application with the authentication capabilities.

TASK 3.1: UPDATE THE APPLICATION TO ENABLE AUTHENTICATION

The *amplify-meta.json* file is generated automatically by the Amplify CLI when you run *amplify init* command. This file contains metadata about your Amplify project and is used by the Amplify library to connect to your cloud resources.

The *amplify-meta.json* file is used by the Amplify CLI to keep track of your project’s backend environment. It contains information about your project’s backend environment such as the *name* of your environment, the *AWS region* where your resources are deployed, and more.

In this task, you obtain the *UserPoolId* value and the *AppClientIDWeb* value to update the SAM template for the API Gateway resource to be able to communicate with Cognito once it is created.

1. **Command:** To view the contents of the **amplify/backend/amplify-meta.json** file, run the following command:

cat ~/environment/react-amplified/amplify/backend/amplify-meta.json

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"providers": {

"awscloudformation": {

"AuthRoleName": "amplify-reactamplified-dev-183225-authRole",

"UnauthRoleArn": "arn:aws:iam::111111111111:role/amplify-reactamplified-dev-183225-unauthRole",

"AuthRoleArn": "arn:aws:iam::111111111111:role/amplify-reactamplified-dev-183225-authRole",

"Region": "us-west-2",

"DeploymentBucketName": "amplify-reactamplified-dev-183225-deployment",

"UnauthRoleName": "amplify-reactamplified-dev-183225-unauthRole",

"StackName": "amplify-reactamplified-dev-183225",

"StackId": "arn:aws:cloudformation:us-west-2:111111111111:stack/amplify-reactamplified-dev-183225/90229c70-c74d-11ed-a568-063961e3d8f3",

"AmplifyAppId": "d3sogr4oscggoz"

}

},

"hosting": {

"amplifyhosting": {

"service": "amplifyhosting",

"providerPlugin": "awscloudformation",

"type": "manual",

"providerMetadata": {

"s3TemplateURL": "https://s3.amazonaws.com/amplify-reactamplified-dev-183225-deployment/amplify-cfn-templates/hosting/amplifyhosting-template.json",

"logicalId": "hostingamplifyhosting"

},

"lastPushTimeStamp": "2023-03-20T20:32:13.552Z",

"output": {},

"lastPushDirHash": "xi5ZgYwfYeFOHtERuead8NN4zwo="

}

},

"auth": {

"reactamplified4c64a58c4c64a58c": {

"service": "Cognito",

"providerPlugin": "awscloudformation",

"dependsOn": [],

"customAuth": false,

"frontendAuthConfig": {

"socialProviders": [],

"usernameAttributes": [

"EMAIL"

],

"signupAttributes": [

"EMAIL"

],

"passwordProtectionSettings": {

"passwordPolicyMinLength": 8,

"passwordPolicyCharacters": []

},

"mfaConfiguration": "OFF",

"mfaTypes": [

"SMS"

],

"verificationMechanisms": [

"EMAIL"

]

},

"providerMetadata": {

"s3TemplateURL": "https://s3.amazonaws.com/amplify-reactamplified-dev-183225-deployment/amplify-cfn-templates/auth/reactamplified4c64a58c4c64a58c-cloudformation-template.json",

"logicalId": "authreactamplified4c64a58c4c64a58c"

},

"lastPushTimeStamp": "2023-03-20T20:32:13.667Z",

"output": {

"UserPoolId": "us-west-2\_axVszKHhS",

"AppClientIDWeb": "5nr9pittlv7o2pcj0eml8254qp",

"AppClientID": "3akhtr71tpdct78l0idnrhk81f",

"UserPoolArn": "arn:aws:cognito-idp:us-west-2:111111111111:userpool/us-west-2\_axVszKHhS",

"UserPoolName": "reactamplified4c64a58c\_userpool\_4c64a58c"

},

"lastPushDirHash": "+ELQ2dT1KrawBZ30JG8omXtMk/E="

}

}

}

1. Find the values shown in the JSON output for the **UserPoolId** and the **AppClientIDWeb**.

The *api-backend-sam/template.yaml* file has place-holders for the *AppClientIDWeb* and the *USERPoolID*. You need to update the values as shown from the output of the *amplify-meta.json* file. (Should be towards the end of the output.)

To accomplish this, you will use a

jq

 command.

**Note:**

jq

 stands for *JASON Query*. It is a lightweight, flexible, and powerful command-line JSON processor. It allows you to filter, transform, and manipulate JSON data in various ways directly from the command line or within shell scripts.

1. **Command:** To create variables to store the values that will be used to update the placeholders for **UserPoolId** and **AppClientIDWeb** in the **api-backend-sam/template.yaml** file, run the following commands:

APP\_CLIENT\_ID\_WEB=$(jq -r '.auth | .[keys[0]] | .output.AppClientIDWeb' ~/environment/react-amplified/amplify/backend/amplify-meta.json)

USER\_POOL\_ID=$(jq -r '.auth | .[keys[0]] | .output.UserPoolId' ~/environment/react-amplified/amplify/backend/amplify-meta.json)

printf "\n\nUserPoolId placeholder value: $USER\_POOL\_ID\n\nAppClientIDWeb placeholder value: $APP\_CLIENT\_ID\_WEB\n\n"

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

UserPoolId placeholder value: us-west-2\_kEL35ZebC

AppClientIDWeb placeholder value: 7llgrsm9d9vc4p1kpcp93u7nhu

To help automate updating the *template.yaml* file with the variables just created, you use the

sed

 command.

**Note:**

sed

 stands for *stream editor*. It is a command line tool used to perform basic text transformations on an input stream. It is most commonly used to replace placeholders in a file with a new value.

1. **Command:** With the variables set, update the **api-backend-sam/template.yaml** file placeholders with the appropriate values using the following

sed

 commands:

clear

before=$(cat ~/environment/api-backend-sam/template.yaml | head -41 | tail -7)

update1=$(sed -i "s/REPLACE\_WITH\_USERPOOL\_ID/$USER\_POOL\_ID/g" ~/environment/api-backend-sam/template.yaml)

update2=$(sed -i "s/REPLACE\_WITH\_APP\_CLIENT\_ID\_WEB/$APP\_CLIENT\_ID\_WEB/g" ~/environment/api-backend-sam/template.yaml)

after=$(cat ~/environment/api-backend-sam/template.yaml | head -41 | tail -7)

printf "template.yaml --> code BEFORE update:\n\n$before; \n\ntemplate.yaml --> code AFTER update:\n\n $after\n\n"

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

template.yaml --> code BEFORE update:

Authorizers:

GeneralAuth:

IdentitySource: "$request.header.Authorization"

JwtConfiguration:

issuer: !Sub "https://cognito-idp.${AWS::Region}.amazonaws.com/REPLACE\_WITH\_USERPOOL\_ID"

audience:

- "REPLACE\_WITH\_APP\_CLIENT\_ID\_WEB";

template.yaml --> code AFTER update:

Authorizers:

GeneralAuth:

IdentitySource: "$request.header.Authorization"

JwtConfiguration:

issuer: !Sub "https://cognito-idp.${AWS::Region}.amazonaws.com/us-west-2\_g1J6kO5Iu"

audience:

- "7uqrunav741v4rja3e63vss28q"

**Note:** These updates are how the application authenticates users.

TASK 3.2: DEPLOY THE BACKEND RESOURCES

With the backend SAM template updated, it is time to create the deployment package and deploy the backend resources that will be used the the application.

1. **Command:** Build the deployment artifacts for the API using SAM with the following commands:

cd ~/environment/api-backend-sam

sam build

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Building codeuri: /home/ec2-user/environment/api-backend-sam/add\_image runtime: python3.9 metadata: {} architecture: x86\_64 functions: AddImageFunction

requirements.txt file not found. Continuing the build without dependencies.

Running PythonPipBuilder:CopySource

Building codeuri: /home/ec2-user/environment/api-backend-sam/generate\_grid runtime: python3.9 metadata: {} architecture: x86\_64 functions: GenerateGridFunction

Running PythonPipBuilder:ResolveDependencies

Running PythonPipBuilder:CopySource

Build Succeeded

Built Artifacts : .aws-sam/build

Built Template : .aws-sam/build/template.yaml

Commands you can use next

=========================

[\*] Validate SAM template: sam validate

[\*] Invoke Function: sam local invoke

[\*] Test Function in the Cloud: sam sync --stack-name {stack-name} --watch

[\*] Deploy: sam deploy --guided

1. **Command:** Now that the deployment package has been created, you deploy the changes to upgrade the API to require authentication with the following command:

sam deploy

**Expected output:** Output has been truncated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Uploading to first-sam/b9f19ea8c53248d010c396d15857686e 675 / 675 (100.00%)

Uploading to first-sam/4381ef5609eb70894de1fe227c1cdce8 14396311 / 14396311 (100.00%)

Deploying with following values

===============================

Stack name : first-sam

Region : us-west-2

Confirm changeset : False

Disable rollback : False

Deployment s3 bucket : sam-app-images-us-west-2-0936467

Capabilities : ["CAPABILITY\_IAM"]

Parameter overrides : {}

Signing Profiles : {}

Initiating deployment

=====================

Uploading to first-sam/eb390d8c82cdff1605b91e0ff0f72d90.template 3325 / 3325 (100.00%)

Waiting for changeset to be created..

CloudFormation stack changeset

----------------------------------------------

CloudFormation outputs from deployed stack

----------------------------------------------

Outputs

----------------------------------------------

Key Api

Description API Gateway endpoint URL

Value https://aoabc0cq3i.execute-api.us-west-2.amazonaws.com/Prod/

----------------------------------------------

Successfully created/updated stack - sam-lab in us-west-2

1. Review the **API Gateway endpoint URL** value. You use this value less the ending **/** (forward slash) to update the **baseUrl** placeholder in the **react-amplified/src/app.js** file in the next task.

TASK 3.3: UPDATE THE REACT APP AND TEST FUNCTIONALITY

To update the React app to know what endpoint to communicate with, you query the **sam-lab** CloudFormation stack, using the

aws cloudformation

 CLI command. It queries the stack for the API Gateway endpoint URL value and stores it in a variable. That variable is then used to update a placeholder for the API endpoint value.

1. **Command:** To set a variable that is used to update the the **baseUrl** placeholder value in the **react-amplified/src/app.js** file, run the following commands:

API\_URL=$(aws cloudformation describe-stacks --stack-name sam-lab --query 'Stacks[0].Outputs[?OutputKey==`Api`].OutputValue' --output text | sed 's/\/$//'); echo ""; echo "URL = $API\_URL"; echo ""

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

URL = https://wugzgju4e5.execute-api.us-west-2.amazonaws.com/Prod

1. **Command:** To replace the **baseUrl** placeholder value with the **Api Gateway endpoint URL**, run the following commands:

clear

before1=$(cat ~/environment/react-amplified/src/App.js | head -15 | tail -2)

update3=$(sed -i "s|REPLACE\_ME\_WITH\_THE\_API\_URL|$API\_URL|g" /home/ec2-user/environment/react-amplified/src/App.js)

after1=$(cat ~/environment/react-amplified/src/App.js | head -15 | tail -2)

printf "App.js --> code BEFORE update:\n\n$before1\n\nApp.js --> code AFTER update:\n\n$after1\n\n"

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

App.js --> code BEFORE update:

const baseUrl = 'REPLACE\_ME\_WITH\_THE\_API\_URL'

App.js --> code AFTER update:

const baseUrl = 'https://vs01rshudi.execute-api.us-west-2.amazonaws.com/Prod'

For React to know what API endpoint to talk to, you had to update the **baseUrl** variable value in the place-holder for **REPLACE\_ME\_WITH\_THE\_API\_URL** section.

1. **Command:** Before updating the Amplify project with these changes, verify what happens if you attempt to **curl** the API with the following commands:

**Note:** As a step in this process, you set the baseUrl variable value to the API Gateway endpoint URL since it doesn’t currently exist. You set the variables and then print their values afterwards to make it cleaner to read. The “-s” in the curl command is used to hide the progress information from the command to clean up the output.

**Hint**

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

baseUrl: https://e1smslsuqd.execute-api.us-west-2.amazonaws.com/Prod

uniqueGridId: 1684769690

Curl command output: {"message":"Unauthorized"}

Now, you need to publish the changes Amplify project to consume these updates.

1. **Command:** To redeploy the Amplify project, run the following commands:

cd ~/environment/react-amplified

amplify publish --yes

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

✔ Successfully pulled backend environment dev from the cloud.

Current Environment: dev

┌──────────┬────────────────────────────────┬───────────┬───────────────────┐

│ Category │ Resource name │ Operation │ Provider plugin │

├──────────┼────────────────────────────────┼───────────┼───────────────────┤

│ Hosting │ amplifyhosting │ No Change │ awscloudformation │

├──────────┼────────────────────────────────┼───────────┼───────────────────┤

│ Auth │ reactamplified3a9d423b3a9d423b │ No Change │ awscloudformation │

└──────────┴────────────────────────────────┴───────────┴───────────────────┘

No changes detected

? Do you still want to publish the frontend? Yes

Publish started for amplifyhosting

> react-amplified@0.1.0 build

> react-scripts build

Creating an optimized production build...

Compiled with warnings.

[eslint]

src/App.js

Line 19:20: 'setUniqueId' is assigned a value but never used no-unused-vars

Search for the keywords to learn more about each warning.

To ignore, add // eslint-disable-next-line to the line before.

File sizes after gzip:

251.18 kB (+174 B) build/static/js/main.6887d31d.js

33.44 kB build/static/css/main.a3e3b07c.css

1.79 kB build/static/js/787.cbb326c1.chunk.js

The project was built assuming it is hosted at /.

You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.

You may serve it with a static server:

npm install -g serve

serve -s build

Find out more about deployment here:

https://cra.link/deployment

✔ Zipping artifacts completed.

✔ Deployment complete!

https://dev.d2mn8e6o5x1t2o.amplifyapp.com

**Note:** With this update, you are publishing the Cognito configuration, which is now part of the frontend application, and the update to the **baseUrl** variable value in the Amplify project so that it knows the API Gateway endpoint to communicate with.

1. Open the application by opening the **URL** from the output in a new browser tab.
2. Choose the **Create Account** link.
3. **Command:** Enter an **email address**, that you have online access to, as well as a **password** and choose **Create Account** .

**Expected output:**

**We Emailed You**

Your code is on the way. To log in, enter the code we emailed to ***@***. It may take a minute to arrive?

1. Enter the **confirmation code** sent to your email in the **Confirmation Code** input field and choose **Confirm** .
2. **Command:** Enter an **email address**, that you have online access to, as well as a **password** and choose .

The browser opens the application allowing you to upload images to generate a grid from.

**Note:** You can choose your own image files or download the following images to upload. The only constraint is that the images must be in any of the following formats:

* filename.jpg
* filename.jpeg

**Images you can download:**

1. From the **Drag and drop images below field** choose **Upload**.
2. Select as many images as you want.
3. Once the images show as being uploaded, choose **Generate Grid** and a grid image based on all of the images you selected with be shown.

**Task complete:** You have created a frontend for the application that requires authentication for all access to the frontend and to the APIs in the backend using Amazon Cognito.

**Conclusion**

You have completed the following:

* Installed requirements to use AWS Amplify CLI.
* Initialized AWS Amplify and deployed a project.
* Configured Cognito to authenticate with the application.
* Ran the grid-maker application using authentication to create the grid image.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* For more information about how to use AWS Lambda, see [AWS Lambda Documentation](https://docs.aws.amazon.com/lambda/index.html).

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).