Deploy a Web App on AWS Amplify

Step 1: - Check Environment

In a new terminal window or command line, run the following commands to verify that you are running at least Node.js version 18.16.0 and npm version 6.14.4 or greater.

- If you are not running these versions, visit the nodejs and npm website for more information.

Note: Your output may differ based on the version installed.

node -v

npm -v

Step 2: - Create a react Application

1. In a new terminal or command line window, run the following command to use Vite to create a React application:

   npm create vite@latest expensetracker -- --template react

   cd expensetracker

   npm install

   npm run dev

2. In the terminal window, select and open the Local link to view the Vite + React application.
3. Open a new terminal window, navigate to your projects root folder (expensetracker), and run the following command:
   npm create amplify@latest -y
4. Running the previous command will scaffold a lightweight Amplify project in the app's directory where you installed the packages.

Step 3: - Setup Amplify Auth

1. On your local machine, navigate to the *amplify/auth/resource.ts* file, and update it with the following code to customize the verification email. Then, save the file.

```
import { defineAuth } from "@aws-amplify/backend";

export const auth = defineAuth({
 loginWith: {
  email: {
   verificationEmailStyle: "CODE",
   verificationEmailSubject: "Welcome to the ExpenseTracker!",
   verificationEmailBody: (createCode) =>
    `Use this code to confirm your account: ${createCode()}`,
  },
 },
});
```

Step 4: - Setup Amplify Data

1. On your local machine, navigate to the *amplify/data/resource.ts* file, and update the following information to define the schema. Then, save the file.

```
import { type ClientSchema, a, defineData } from '@aws-amplify/backend';

const schema = a.schema({

    Expense: a

    .model({

    name:a.string(),

    amount: a.float(),

    })

    .authorization((allow) => [allow.owner()]),

    });

    export type Schema = ClientSchema<typeof schema>;

    export const data = defineData({

    schema,

    authorizationModes: {

    defaultAuthorizationMode: 'userPool',

    },

    });
```

Step 5: - Deploy Amplify Cloud Snadbox

1. Open a new terminal window, navigate to your app's root folder (*expensetracker*), and run the following command to deploy cloud resources into an isolated development space so you can iterate fast.
   npx ampx sandbox

2. After the cloud sandbox has been fully deployed, your terminal will display a confirmation message and the *amplify_outputs.json* file will be generated and added to your project. This deployment will take several minutes to complete.

Step 6: - Install the Amplify Libraries

1. Open a new terminal window, navigate to your projects root folder (*expensetracker*), and run the following command to install these libraries in the root of the project.
   npm install aws-amplify @aws-amplify/ui-react

Step 7: - Style the app UI

1. On your local machine, navigate to the *expensetracker/src/index.css* file, and update it with the following code to set the style of the UI. Then, save the file.

```
:root {
  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;

  color: rgba(255, 255, 255, 0.87);

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;

  max-width: 1280px;
  margin: 0 auto;
  padding: 2rem;

}

.card {
  padding: 2em;
}

.read-the-docs {
  color: #888;
}

.box:nth-child(3n + 1) {
  grid-column: 1;
}
.box:nth-child(3n + 2) {
  grid-column: 2;
}
.box:nth-child(3n + 3) {
  grid-column: 3;
}
```

Step 8: - Implement the UI Flow for Expenses feature

1. In this step, you will update the src/App.jsx file to configure the Amplify library with the client configuration file (amplify_outputs.json). Then, it will generate a data client using the generateClient() function.

   The code uses the Amplify Authenticator component to scaffold out an entire user authentication flow allowing users to sign up, sign in, reset their password, and confirm sign-in for multifactor authentication (MFA).

   Additionally, the code contains the following:

   - A code to use a real-time observeQuery to subscribe to a live feed of the user's expenses data.
   - createExpense - Get the data from the form and use the data client to create a new expense.
   - deleteExpense - Use the data client to delete the selected expense.

   Code: -

```
import { useState, useEffect } from "react";
import {
 Authenticator,
 Button,
 Text,
 TextField,
 Heading,
 Flex,
 View,
 Grid,
 Divider,
} from "@aws-amplify/ui-react";
import { Amplify } from "aws-amplify";
import "@aws-amplify/ui-react/styles.css";
import { generateClient } from "aws-amplify/data";
import outputs from "../amplify_outputs.json";
/**
 * @type {import('aws-
amplify/data').Client<import('../amplify/data/resource').Schema>}
 */

Amplify.configure(outputs);
const client = generateClient({
 authMode: "userPool",
});

export default function App() {
 const [expenses, setExpenses] = useState([]);

 useEffect(() => {
```

```
  client.models.Expense.observeQuery().subscribe({
    next: (data) => setExpenses([...data.items]),
  });
}, []);

async function createExpense(event) {
  event.preventDefault();
  const form = new FormData(event.target);

  await client.models.Expense.create({
    name: form.get("name"),
    amount: form.get("amount"),
  });

  event.target.reset();
}

async function deleteExpense({ id }) {
  const toBeDeletedExpense = {
    id,
  };

  await client.models.Expense.delete(toBeDeletedExpense);
}

return (
  <Authenticator>
    {({ signOut }) => (
      <Flex
        className="App"
        justifyContent="center"
        alignItems="center"
        direction="column"
        width="70%"
        margin="0 auto"
      >
        <Heading level={1}>Expense Tracker</Heading>
        <View as="form" margin="3rem 0" onSubmit={createExpense}>
          <Flex
            direction="column"
            justifyContent="center"
            gap="2rem"
            padding="2rem"
          >
            <TextField
              name="name"
              placeholder="Expense Name"
              label="Expense Name"
              labelHidden
              variation="quiet"
```

```
    required
   />
   <TextField
    name="amount"
    placeholder="Expense Amount"
    label="Expense Amount"
    type="float"
    labelHidden
    variation="quiet"
    required
   />

   <Button type="submit" variation="primary">
    Create Expense
   </Button>
  </Flex>
 </View>
 <Divider />
 <Heading level={2}>Expenses</Heading>
 <Grid
  margin="3rem 0"
  autoFlow="column"
  justifyContent="center"
  gap="2rem"
  alignContent="center"
 >
  {expenses.map((expense) => (
   <Flex
    key={expense.id || expense.name}
    direction="column"
    justifyContent="center"
    alignItems="center"
    gap="2rem"
    border="1px solid #ccc"
    padding="2rem"
    borderRadius="5%"
    className="box"
   >
    <View>
     <Heading level="3">{expense.name}</Heading>
    </View>
    <Text fontStyle="italic">${expense.amount}</Text>

    <Button
     variation="destructive"
     onClick={() => deleteExpense(expense)}
    >
     Delete note
    </Button>
   </Flex>
```

```
        ))}
      </Grid>
      <Button onClick={signOut}>Sign Out</Button>
    </Flex>
  )}
 </Authenticator>
);
}
```

2. Open a new terminal window, navigate to your projects root folder (*expensetracker*), and run the following command to launch the app:
```
npm run dev
```

3. Select the Local host link to open the Vite + React application.
4. Choose the Create Account tab, and use the authentication flow to create a new user by entering your email address and a password. Then, choose Create Account.
5. You will get a verification code sent to your email. Enter the verification code to log into the app.
6. When signed in, you can start creating expenses and delete them.

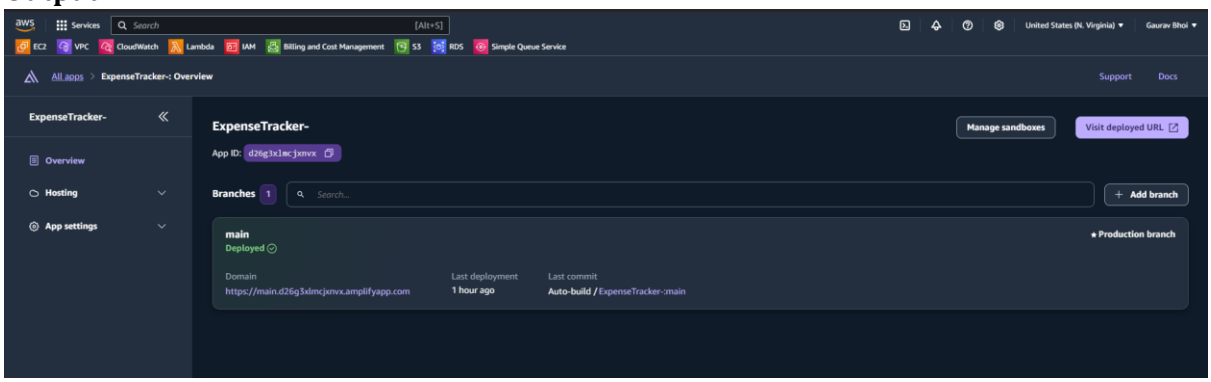
Step 9: - Initialize GitHub Repository

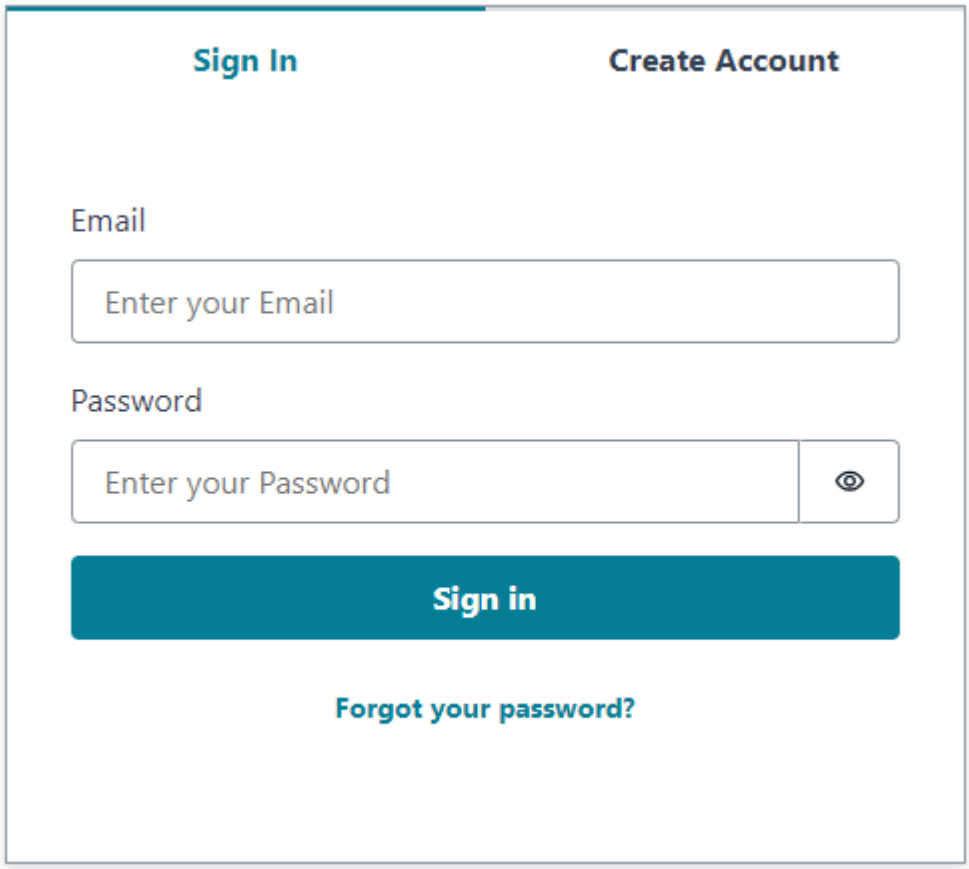
1. Sign in to GitHub at https://github.com/.
2. In the Start a new repository section, make the following selections:
   For Repository name, enter expensetracker, and choose the Public radio button.
   Then select, Create a new repository.
3. Open a new terminal window, navigate to your projects root folder (*expensetracker*) , and run the following commands to initialize a git and push of the application to the new GitHub repo:
   Notes:
   Replace the SSH GitHub URL in the command with your GitHub URL.
   git init
   git add .
   git commit -m "first commit"
   git remote add origin git@github.com:<your-username>/profilesapp.git
   git branch -M main
   git push -u origin main

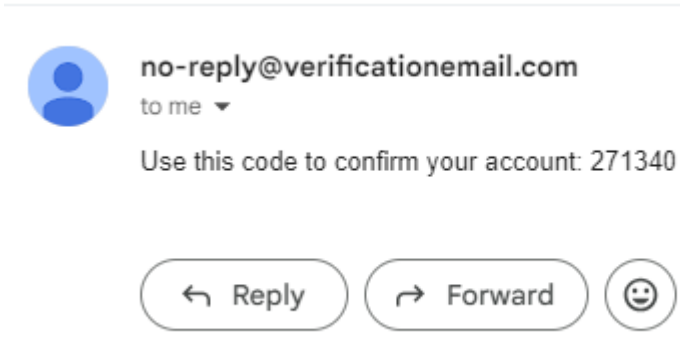
Step 10: -Create your app with aws amplify

1. Sign in to the AWS Management Console in a new browser window, and open the AWS Amplify console at https://console.aws.amazon.com/amplify/apps.
2. Choose Create new app.
3. On the Start building with Amplify page, for Deploy your app, select GitHub, and select Next.
4. When prompted, authenticate with GitHub. You will be automatically redirected back to the Amplify console. Choose the repository and main branch that you created earlier. Then, select Next.
5. Leave the default build setting, and select Next.
6. Review the inputs selected and choose Save and deploy.
7. Once the build completes, select the Visit deployed URL button to see your web app up and running live.

Output: -


aws
Services
Search
[Alt+S]
EC2
VPC
CloudWatch
Lambda
IAM
Billing and Cost Management
S3
RDS
Simple Queue Service
United States (N. Virginia)
Gaurav Bhoi
All apps
ExpenseTracker-: Overview
Support
Docs
ExpenseTracker-
Overview
Hosting
App settings
ExpenseTracker-
App ID: d26g3xlmcjxnvx
Manage sandboxes
Visit deployed URL
Branches 1
Search...
Add branch
main
Deployed
Production branch
Domain
https://main.d26g3xlmcjxnvx.amplifyapp.com
Last deployment
1 hour ago
Last commit
Auto-build / ExpenseTracker-:main


Sign In
Create Account
Email
Enter your Email
Password
Enter your Password
Sign in
Forgot your password?


no-reply@verificationemail.com
to me
Use this code to confirm your account: 271340
Reply
Forward

Expense Tracker

Expense Name

Expense Amount

Create Expense

Expenses

Gaurav Bhoi

$240

Delete note

Sign Out