Akilandeshwari Srinivasan(451036)

Lab 5 : Exploring CI/CD - Case Scenario for AWS Amplify Training: Introduction for Junior Cloud DevOps

CLCM3102 - Cloud Application Requirements and Specifications

Preparation:

Gone through the basic git commands and the documentation of AWS Amplify. In this lab we are going to deploy an React application to the cloud using AWS Amplify.

- Create a repository in github
- Create an react application.
- Configure AWS Amplify and deploy the react application.
- Do some changes in the code and check the output.

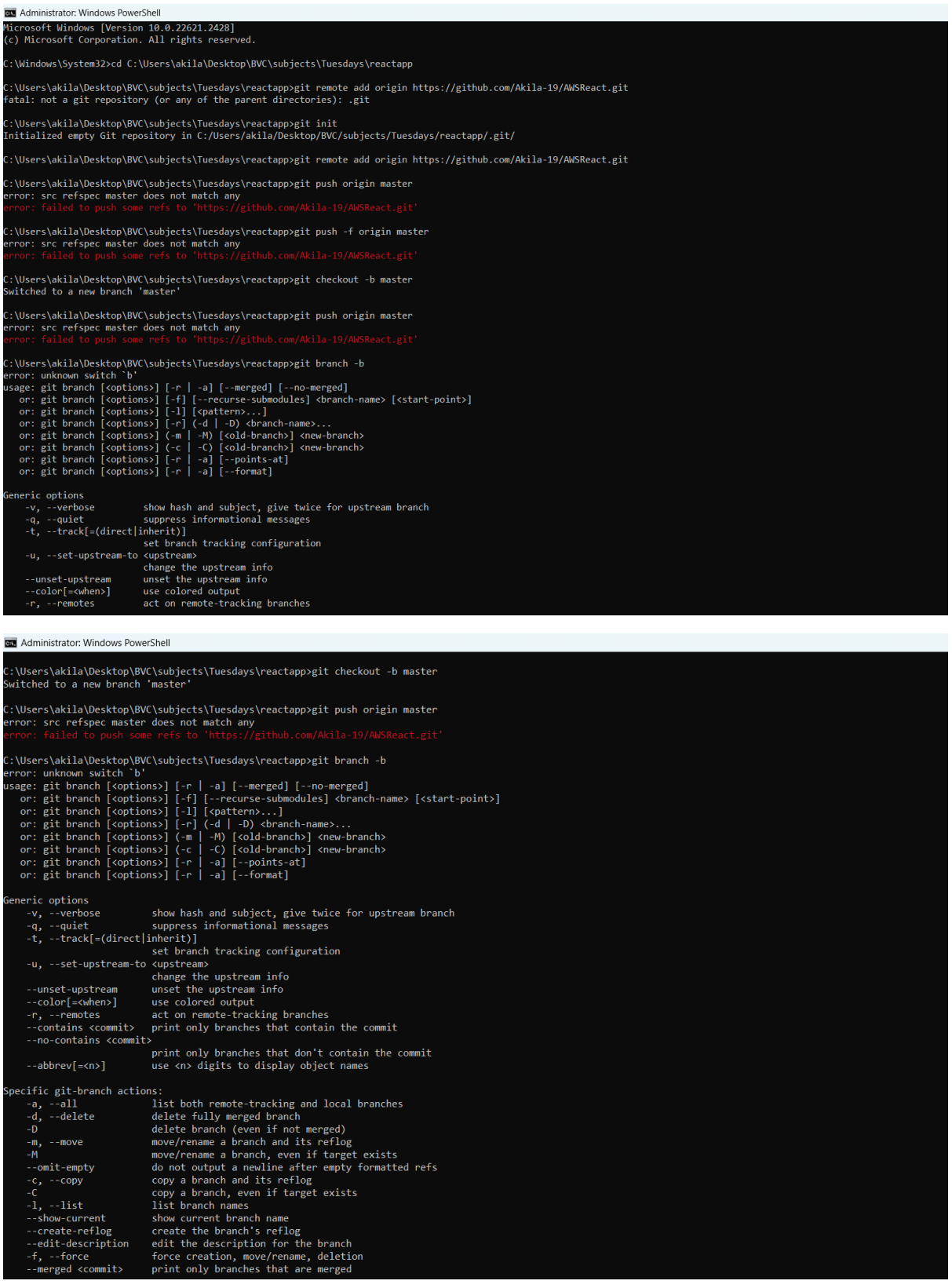
Step 1: Git Definition:

Git is an essential tool in software development. It plays an important role in version control, collaboration, and source code management. With Git, we can keep track of changes made to our codebase, work effectively with a team of developers, and ensure that our project's history is well-documented. we need to create a Git repository for our project. We can do this by using the following commands:

git init → initializes a new Git repository

git add . → stages all the files in the current directory for the initial commit

git commit -m "Initial commit"→ commit the changes to the local with a description message



Administrator: Windows PowerShell

```
C:\Users\akila\Desktop\BVC\subjects\Tuesdays\reactapp>git --list
unknown option: --list
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--config-env=<name>=<envvar>] <command> [<args>]

C:\Users\akila\Desktop\BVC\subjects\Tuesdays\reactapp>git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package-lock.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'package.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'public/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'public/manifest.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'public/robots.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/App.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/App.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/App.test.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/index.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/index.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/reportWebVitals.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/setupTests.js', LF will be replaced by CRLF the next time Git touches it

C:\Users\akila\Desktop\BVC\subjects\Tuesdays\reactapp>git commit -m "new changes"
[master (root-commit) 127fd3b] new changes
 18 files changed, 18186 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 README.md
 create mode 100644 package-lock.json
 create mode 100644 package.json
 create mode 100644 public/favicon.ico
 create mode 100644 public/index.html
 create mode 100644 public/logo192.png
 create mode 100644 public/logo512.png
 create mode 100644 public/manifest.json
 create mode 100644 public/robots.txt
 create mode 100644 src/App.css
 create mode 100644 src/App.js
 create mode 100644 src/App.test.js
 create mode 100644 src/index.css
 create mode 100644 src/index.js
 create mode 100644 src/logo.svg
 create mode 100644 src/reportWebVitals.js
 create mode 100644 src/setupTests.js

C:\Users\akila\Desktop\BVC\subjects\Tuesdays\reactapp>git push origin master
```

Administrator: Windows PowerShell

```
C:\Users\akila\Desktop\BVC\subjects\Tuesdays\reactapp>git push origin master
info: please complete authentication in your browser...
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 12 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (22/22), 176.58 KiB | 8.03 MiB/s, done.
Total 22 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/Akila-19/AWSReact/pull/new/master
remote:
To https://github.com/Akila-19/AWSReact.git
 * [new branch]      master -> master

C:\Users\akila\Desktop\BVC\subjects\Tuesdays\reactapp>npm start

> reactapp@0.1.0 start
> react-scripts start

(node:2100) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewar
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:2100) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlew
Starting the development server...
Compiled successfully!

You can now view reactapp in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.25.149.141:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Step 2: Simple React App Creation

To create a react app we have to open a command prompt and install Node Package Manager(npm) and npx. Then do the following commands.

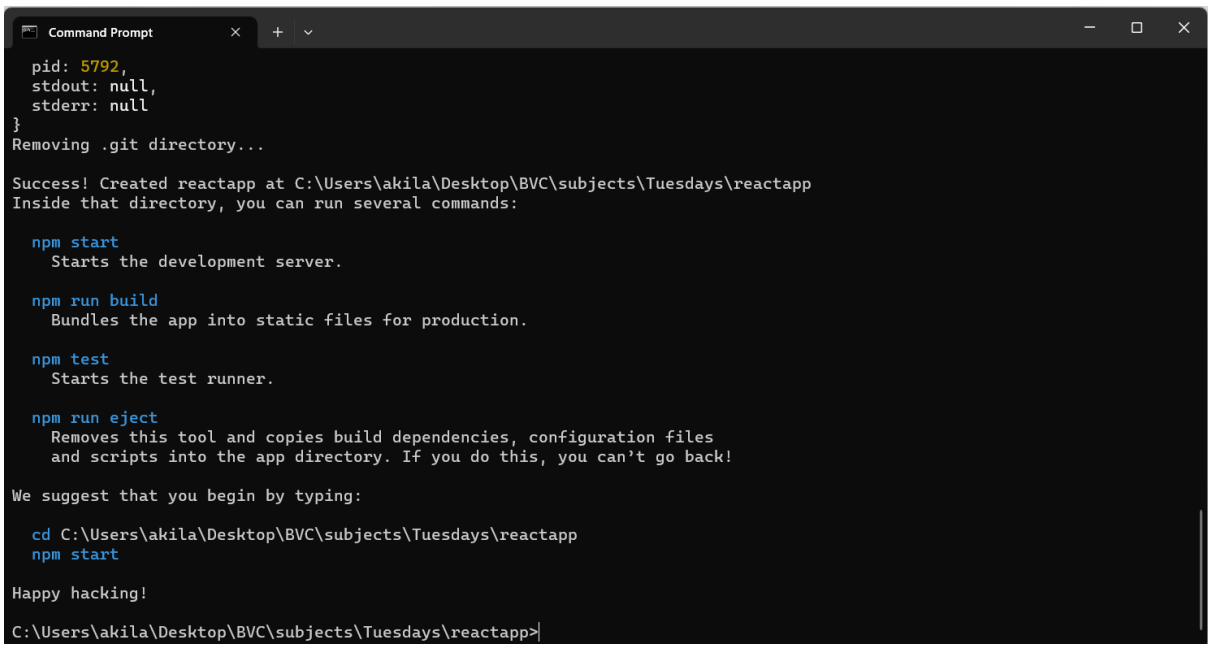
npx create-react-app my-react-app →→→ creates a new React application named "my-react-app."

cd my-react-app-→→→ changes your working directory to the newly created app's folder

npm install react react-dom n

```
npm ERR! 404 This package name is not valid, because
npm ERR! 404  1. name can no longer contain capital letters
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.

npm ERR! A complete log of this run can be found in:
npm ERR!     C:\Users\akila\AppData\Local\npm-cache\_logs\2023-10-17T17_42_30_963Z-debug-0.log

C:\Users\akila\Desktop\BVC\subjects\Tuesdays\AWSAmplifyReactApp>npx create-react-app .
Cannot create a project named "AWSAmplifyReactApp" because of npm naming restrictions:

  * name can no longer contain capital letters

Please choose a different project name.

C:\Users\akila\Desktop\BVC\subjects\Tuesdays\AWSAmplifyReactApp>cd..

C:\Users\akila\Desktop\BVC\subjects\Tuesdays>mkdir reactapp

C:\Users\akila\Desktop\BVC\subjects\Tuesdays>cd reactapp

C:\Users\akila\Desktop\BVC\subjects\Tuesdays\reactapp>npx create-react-app .

Creating a new React app in C:\Users\akila\Desktop\BVC\subjects\Tuesdays\reactapp.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

[                    ] | idealTree:reactapp: timing idealTree:#root Completed in 2209ms
```

Activity Details

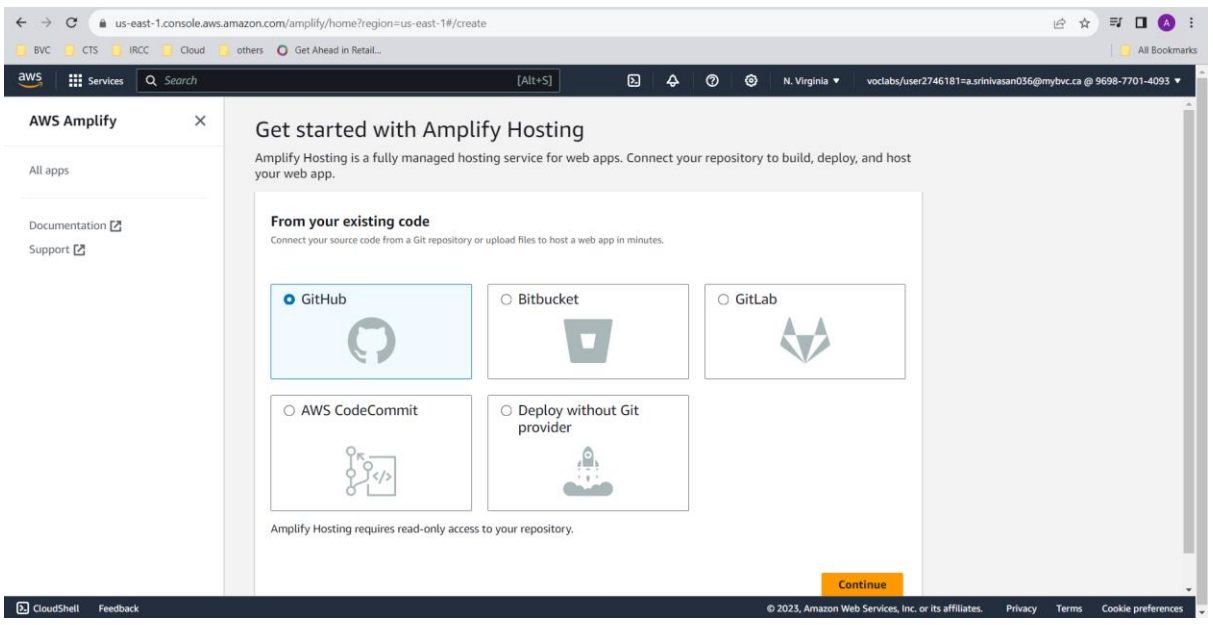


Step 3: AWS Amplify

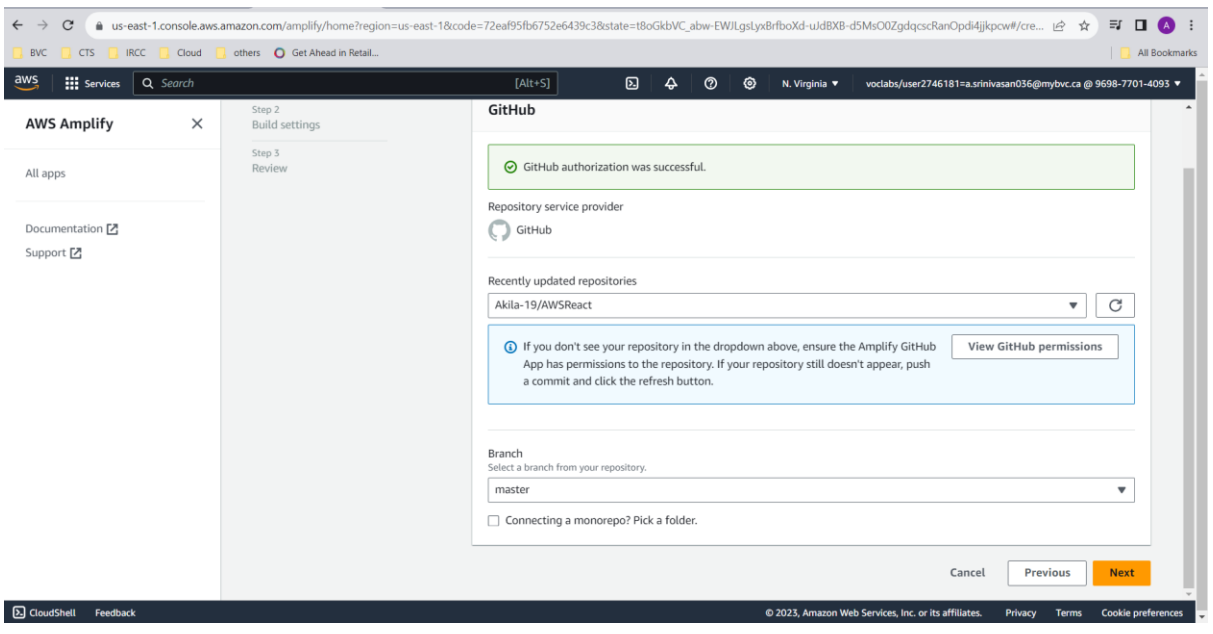
AWS Amplify is a powerful service that simplifies web application development and deployment on Amazon Web Services (AWS). It provides a wide range of features to help streamline the development process. One of the most significant benefits of Amplify is its ability to automate the Continuous Integration and Continuous Deployment (CI/CD) process. This means that changes to your code can be automatically built, tested, and deployed to AWS services when pushed to the connected Git repository. Sign in to the AWS Console and search for this service.

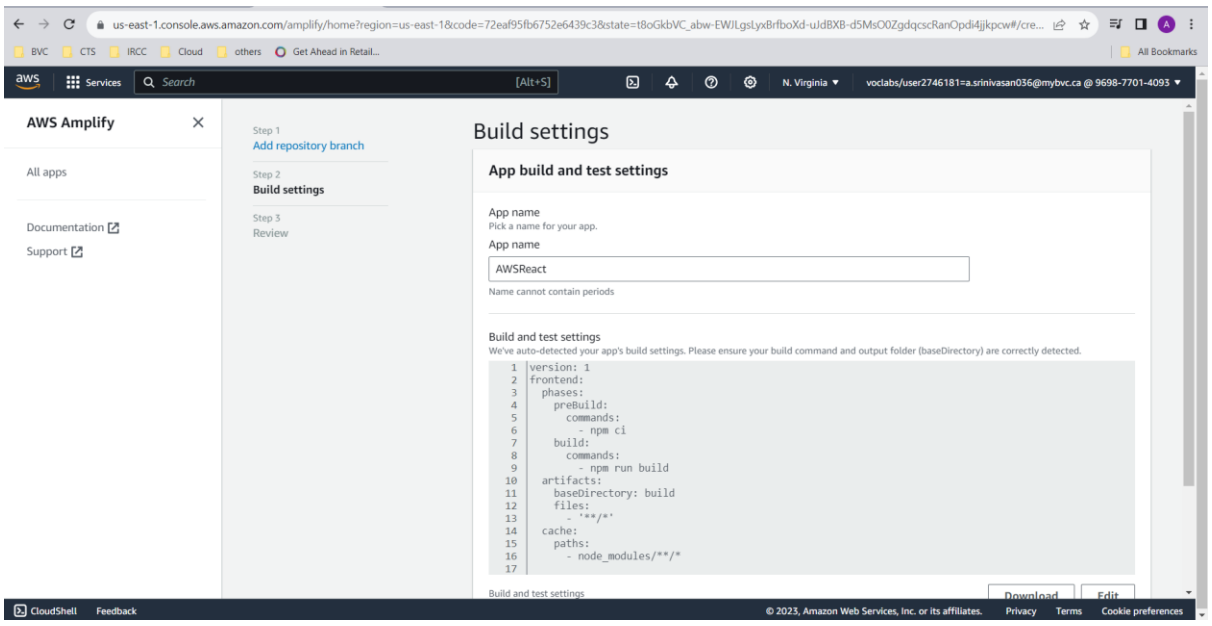
Step 4: Manual deployment to AWS Amplify

Open the aws management console and search for the service. As we have the code in git repository choose “github” option and give continue.

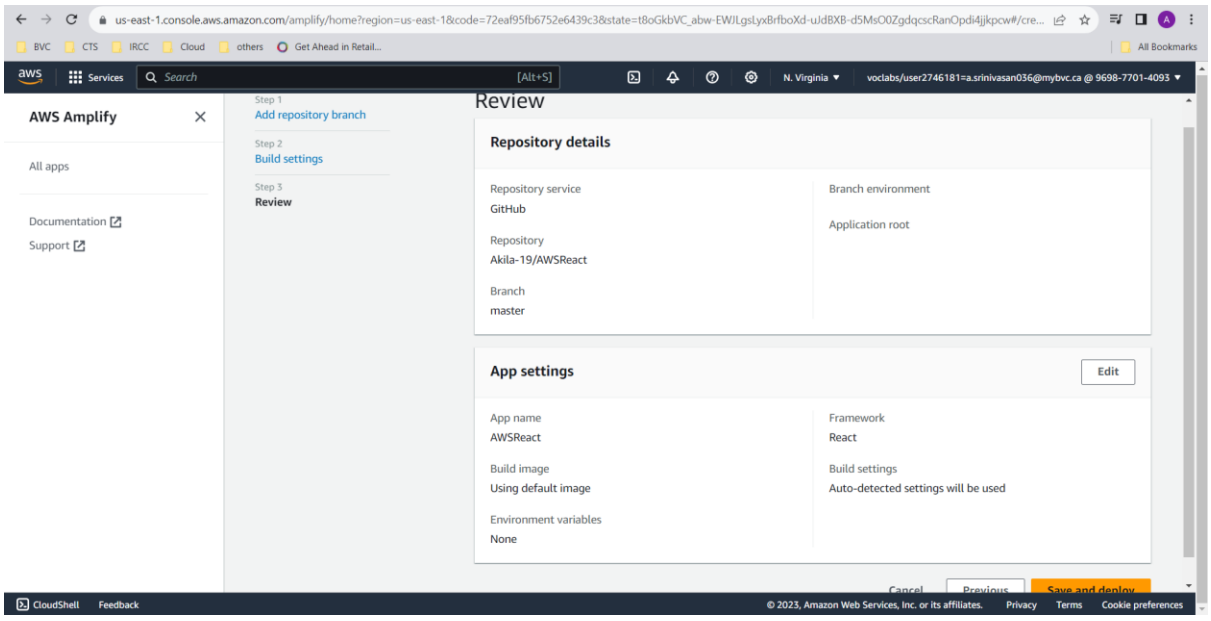


Now select your repository and the branch in the github where you have pushed the code.

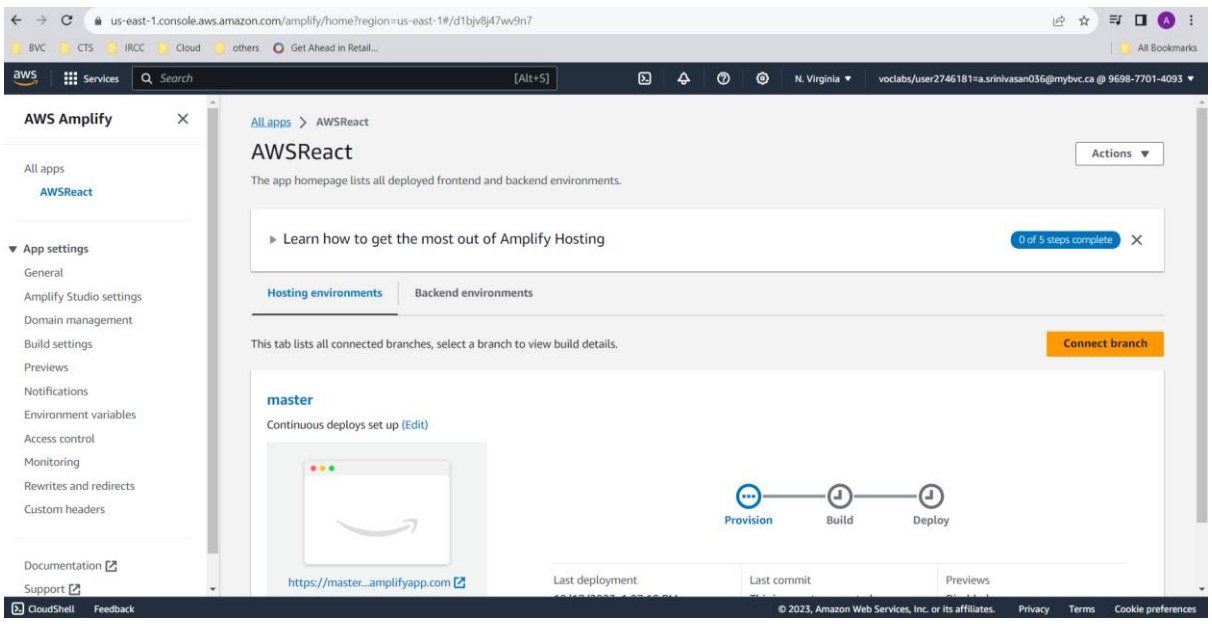




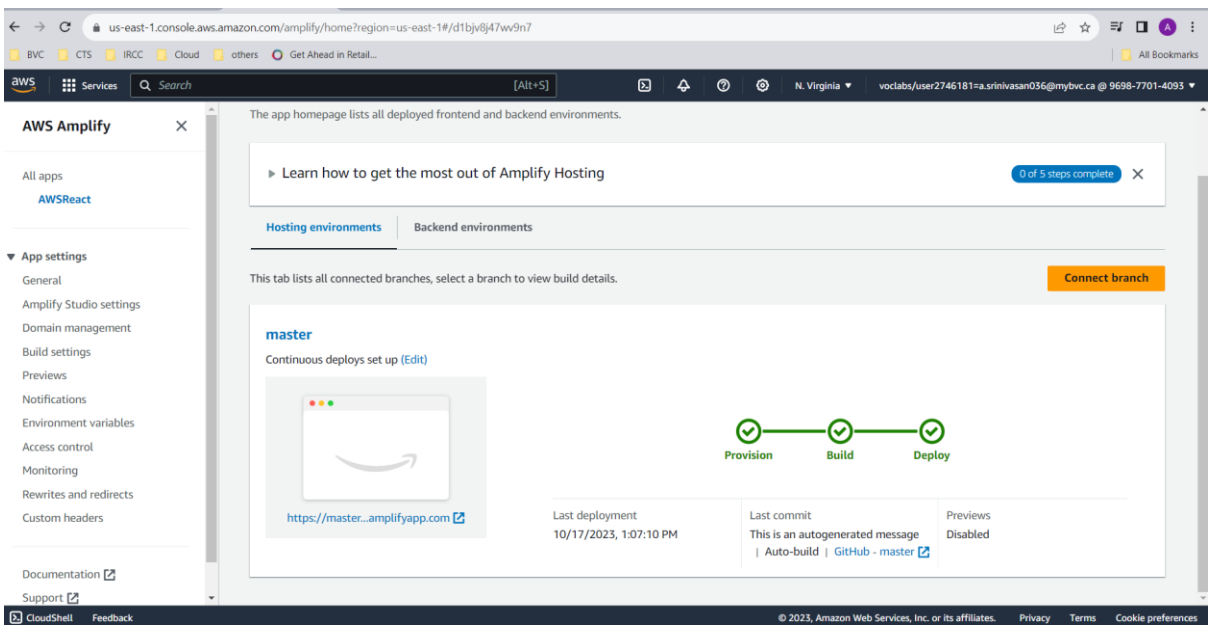
Once its selected give next and click save and deploy button to start the deployment.



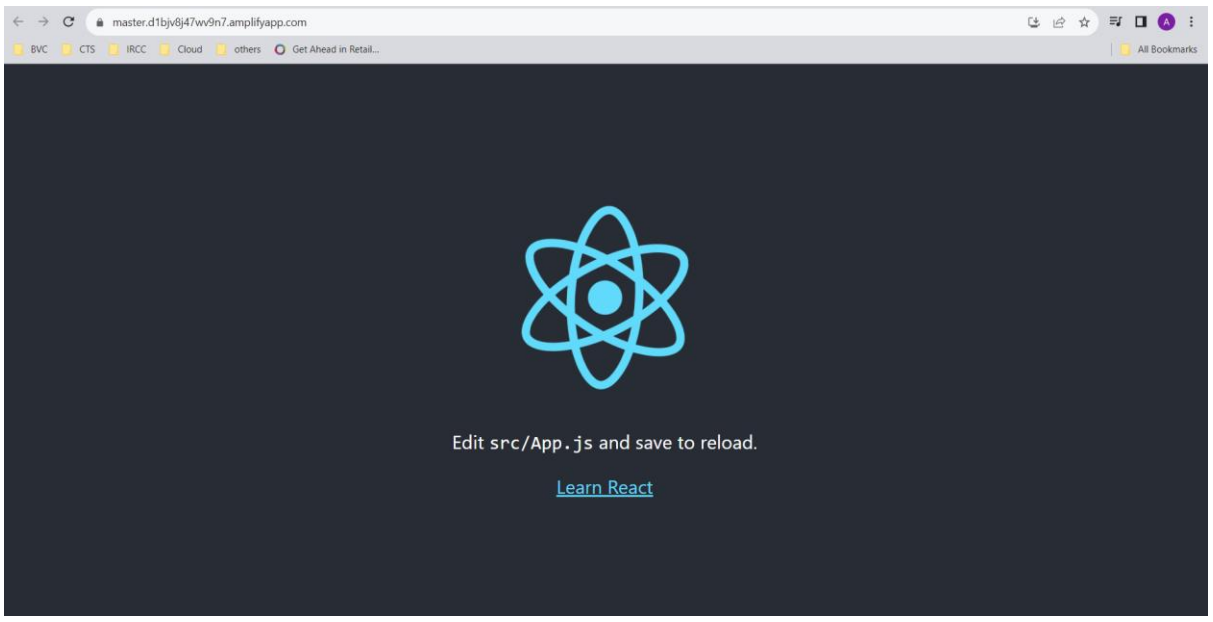
It takes some time to deploy. Once its done its shows green which indicates its completed.



Once its completed you can click the link and see the output in the browser.

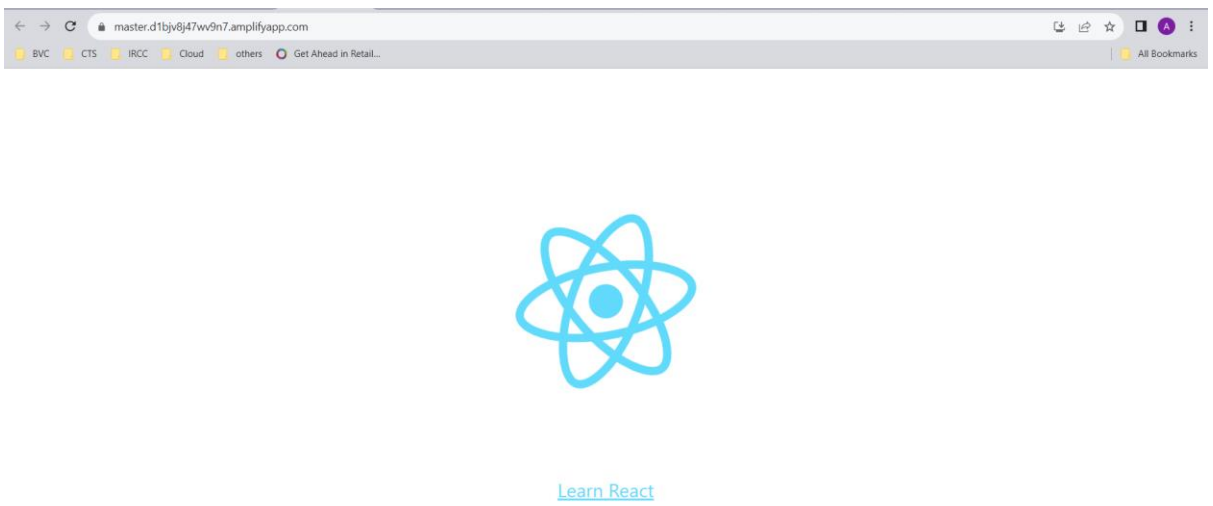


This is the output of the react code after deployed in the AWS Amplify. The URL itself indicates it as amplify.



Now do some changes in the code like I have changed the background colour of the react website and again commit the changes to the git.

Now without doing anything manually the changes have reflected in the browser.



Step :5 Continuous Deployment:

Continuous Deployment is a important concept in modern software development. It means that any changes pushed to your connected Git repository will automatically trigger the build and deployment processes. AWS Amplify is set up to handle this for you, ensuring that your application stays up-to-date and reflects the latest code changes.

In summary, Git is the foundation for version control, Create React App simplifies React development, and AWS Amplify streamlines web app development and deployment with CI/CD automation. With this combination, you can efficiently manage your code, build and deploy your React apps, and enjoy the benefits of continuous deployment with AWS Amplify.

References: www.google.com , AWS Official Documentation