CI/CD – continuous integration and continuous deployment

Github actions run test before deploying it. It make sure that each test passed, even if one or two test fails, it will not allow to deploy.

TEST DRIVEN DEVELOPMENT:

Write first test even before writing code. We write code so that it can passed the test. So test is our major prirority. First we learn how to write test, before jumping on to test driven development.

Test is used to find error in the code, so to make our code error free.

React by default provide test dependencies

For CI/CD, we use travis CI. Their we can run test

In react, App.test.js and files similar to that use to run the test

We are automating the test

To run the test in react:

Yarn test

or

npm test

To import test:

import { render, screen } from '@testing-library/react';

TEST:

test(‘test name', () => {

  const {getByText} = render(Component to used like <App/>);

  const linkElement = getByText(/What we want to test like Hello World/i);

  expect(linkElement).toBeInTheDocument();

});

function App() {

  return (

    <div className="App">

      Hello World

      Hello Muneeb

    </div>

  );

}

export default App;

test('renders learn react link', () => {

  const {getByText} = render(<App />);

  const linkElement = getByText(/Hello World/i);

  expect(linkElement).toBeInTheDocument();

});

test('test case', () => {

  const {getByText} = render(<App />);

  const linkElement = getByText(/Hello Muneeb/i);

  expect(linkElement).toBeInTheDocument();

});

Now this, time we need to check again again whether all test has been passed or not, to avoid this we use CI/CD technique, it will automatically run the test cases and keep checked them.

For this, we use travis ci and it give option to run test cases.

First we place test cases in travis ci, then it will run test and after that we can deploy it on surge or anywhere else. Incase of surge, github actions check whether these test cases passed or not, after that it will deploy on surge

1. CREATE TRAVIS CI ACCOUNT.
2. Then create .travis.yml file outside src folder.

In travis.yml:

script:

  - npm install

  - npm test

  - npm run build

Travis clone repository from github, then use npm install to install dependencies, then npm test to run the test, then build and if all successful then deploy

Deploy can be github pages(gh.pages) or surge

deploy:

  provider: surge

  skip\_cleanup: true

  domain:

  project: ./build/

on:

  branch: master

Now we are left with domain, for domain run this command

npm run build && cd build && surge

It will create build file, then go into that file and then deploy it on surge and give us the domain

After doing that, I got this command

tasteless-dad.surge.sh

happy-eyes.surge.sh

In client side, we automate ui testing.