BTI425/WEB422 - Web Programming for Apps and Services

Lecture Recap:

Week 11 – Deployment & Automated Testing

Agenda

- Continuous Integration
- Continuous Deployment

Continuous Integration

- CI/CD
  - stands for "Continuous Integration / Continuous Deployment"
  - is a vital technique in ensuring software quality and quick delivery.
- Continuous Integration:
  - the practice of automating the integration of code changes from multiple contributors into a single software project
  - a primary Agile and DevOps best practice of routinely integrating code changes into the main branch of a repository, and testing the changes, as early and often as possible.
- A source code version control system is the crux of the CI process.

Git / GitHub Review

Common "git" commands:

- git init
- git clone: get a local copy from remote repo

  >git clone https://github.com/wsong18/web322-assignments.git local-assignments

  Note:

  - to keep it simple, use the one with protocol **https,** e.g., https://github.com/wsong18/web322-assignments.git
  - recall: >git remote add origin https://github.com/wsong18/web322-assignments.git
- git add .: stages a change
- git commit: saves the snapshot to the project history …

  >git commit -m "initial commit"
- git status
- git remote: manage the set of "remote" repo whose branch you track

  >git remote -v

Git / GitHub Review

- git checkout: switch branches or restore working tree files (the -b flag creates a new branch before switching to it)

  >git checkout -b new-branch

- git branch: shows the local branches
- git merge: used to combine changes made on two distinct branches

  >git checkout master

  >git merge new-branch

- git pull: updates the local line of development with updates from its remote counterpart

  >git checkout -b new-branch

  >git pull origin master

- git push: updates the remote repository with any commits made locally to a branch

  >git push origin new-branch

  >git push origin master

  Note: see pull requests

Automating Tasks

- Hosting Your Code
  - Create a GitHub Repository,
  - Prepare Our Local Git Repo,
    - npx create-next-app my-app --use-npm
    - git init
    - git add .
    - git commit -m "Initial commit"
  - Connect the Local Git Repository to GitHub
    - git remote add origin URL
    - git push origin main // git push origin master
- GitHub Actions
  - *GitHub Actions* is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline.
  - *GitHub Actions* goes beyond just DevOps and lets you run workflows when other events happen in your repository.
  - GitHub provides Linux, Windows, and macOS virtual machines to run your workflows.

Automating Tasks

- Getting started using GitHub actions
  - Creating a "workflow" that responds to an event.
    - Creating the following starter file: .github/workflows/ci.yaml:

```
name: CI
on: [push]
jobs:
run-tests:
name: Lint and Test
runs-on: ubuntu-latest

steps:
- name: Checkout code
uses: actions/checkout@v3

- name: Install packages
run: npm ci

- name: Run ESlint
run: npm run lint

- name: Run Jest Tests
run: npm run ci:test

- name: Run Cypress Tests
uses: cypress-io/github-action@v4
with:
build: npm run build
start: npm start
```

- Note: need to add script entry in package.json:
  "ci:test": "jest",

Automating Tasks

- Merging Code from Other Branches
  - Create and switch to a new "branch": >git checkout -b fix/logo
  - Check current branch: >git branch
  - In "pages/index.js", change "By{' '}" to "By:{' ' }" to generate an "Error"; changie the "vercel.com" link to "abc.com"
  - On gitHub.com "Pull requests" page, press "Merge pull request" button if no conflict and all checks passed.

Continuous Deployment

- CD
  - stands for "Continuous Delivery" or "Continuous Deployment"
- Continuous delivery
  - is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage
    - This means that on top of automated testing, you have an automated release process and you can deploy your application any time by clicking a button
- Continuous Deployment
  - Continuous deployment goes one step further than continuous delivery.
  - With this practice, every change that passes all stages of your production pipeline is released to your customers.
  - There's no human intervention, and only a failed test will prevent a new change to be deployed to production.

Introduction to Vercel

- Introduction to Vercel
  - https://vercel.com
  - "Import Git Repository"
    - "Only select repositories"

- **Updating the Production Site**

The End