**Assignment 4: Continuous Integration and Deployment (CI/CD)**

**Set up a CI/CD pipeline for your MERN app.**

**Automatically deploy changes to a staging environment.**

**Setting up a Continuous Integration/Continuous Deployment (CI/CD) pipeline is crucial for automating the testing and deployment processes of your MERN (MongoDB, Express.js, React, Node.js) app. Below is a general guide using popular CI/CD tools like GitHub Actions and Heroku for deploying changes to a staging environment.**

**Step 1: Version Control and Repository**

**Ensure your MERN app is hosted in a version-controlled repository, such as GitHub. If not, initialize a Git repository and push your code to a GitHub repository.**

| **# Initialize a Git repository**  **git init**  **# Add and commit your code**  **git add .**  **git commit -m "Initial commit"**  **# Create a new GitHub repository and push your code**  **git remote add origin <repository\_url>**  **git push -u origin master** |
| --- |

**Step 2: Set Up GitHub Actions for CI**

**Create a .github/workflows/ci.yml file in your project to define the CI workflow. This workflow will run tests whenever changes are pushed to the repository.**

| **# .github/workflows/ci.yml**  **name: CI**  **on:**  **push:**  **branches:**  **- main**  **jobs:**  **build:**  **runs-on: ubuntu-latest**  **steps:**  **- name: Checkout code**  **uses: actions/checkout@v2**  **- name: Set up Node.js**  **uses: actions/setup-node@v2**  **with:**  **node-version: '14'**  **- name: Install dependencies**  **run: npm install**  **- name: Run tests**  **run: npm test** |
| --- |

**Step 3: Set Up Heroku for Deployment**

**Create a Heroku Account:**

**If you don't have a Heroku account, sign up here.**

**Install Heroku CLI:**

**Install the Heroku CLI by following the instructions here.**

**Login to Heroku:**

**Open a terminal and run:**

| **heroku login** |
| --- |

**Step 4: Set Up GitHub Actions for CD**

**Create a .github/workflows/cd.yml file to define the CD workflow. This workflow will deploy changes to Heroku whenever you push to the main branch.**

| **# .github/workflows/cd.yml**  **name: CD**  **on:**  **push:**  **branches:**  **- main**  **jobs:**  **deploy:**  **runs-on: ubuntu-latest**  **steps:**  **- name: Checkout code**  **uses: actions/checkout@v2**  **- name: Set up Node.js**  **uses: actions/setup-node@v2**  **with:**  **node-version: '14'**  **- name: Install Heroku CLI**  **run: curl https://cli-assets.heroku.com/install.sh | sh**  **- name: Login to Heroku**  **run: heroku login -i**  **- name: Deploy to Heroku**  **run: |**  **git remote add heroku https://git.heroku.com/<your-heroku-app-name>.git**  **git push heroku main** |
| --- |

**Step 5: Configure Heroku**

1. **Create a Heroku App:**

**Create a new Heroku app using the Heroku CLI:**

| **heroku create <your-heroku-app-name>** |
| --- |

1. **Configure Environment Variables:**

**Set any necessary environment variables in your Heroku app. For example, set the NODE\_ENV variable to "production".**

| **heroku config:set NODE\_ENV=production** |
| --- |

1. **Add MongoDB Atlas (Optional):**

**If you are using MongoDB, configure the MongoDB Atlas add-on or any other preferred database.**

**Step 6: Push Changes**

**Push your changes to the main branch to trigger the CI/CD pipeline:**

| **git add .**  **git commit -m "Setup CI/CD"**  **git push origin main** |
| --- |

**Step 7: Monitor Workflow Execution**

**Visit the "Actions" tab in your GitHub repository to monitor the execution of the CI/CD workflows.**

**Notes:**

**Replace <your-heroku-app-name> with the actual name of your Heroku app.**

**Customize the CI/CD workflows according to your project structure and requirements.**

**Ensure that your application has proper configuration for different environments (development, testing, production).**

**With these steps, you've set up a basic CI/CD pipeline for your MERN app using GitHub Actions and Heroku. Customize the pipeline to include additional steps such as database migrations, asset optimizations, or other deployment tasks specific to your application.**