What is Github?

GitHub is a platform for hosting, sharing, and collaborating on code repositories. It provides a place where developers can store and manage their code, track changes, and work with other people. GitHub is built on top of Git, a version control system that allows you to keep a history of your codebase, revert to previous versions, and collaborate with others more effectively.

CI/CD stands for Continuous Integration and Continuous Deployment (or Continuous Delivery), and it refers to a set of practices used in software development to ensure code changes are tested, integrated, and deployed quickly and reliably.

- **Continuous Integration (CI)**: Developers regularly merge code changes into a shared repository. Each change is automatically tested to catch issues early, making it easier to address bugs and ensure the codebase remains stable.
- **Continuous Deployment (CD)**: After successful testing, code is automatically deployed to production (Continuous Deployment) or staged for manual review and approval (Continuous Delivery). This makes the release process faster and allows for more frequent updates.

CI/CD pipelines are often automated using tools like GitHub Actions, Jenkins, or CircleCI, which integrate with GitHub and other code repositories to streamline the entire process.

1. Deploying to Netlify (Static Sites or Frontend Projects)

Netlify is ideal for static sites or frontend frameworks like React, Vue, or simple HTML/CSS/JS projects.

Steps:

1. **Push Your Code to GitHub**:
   - Ensure your code is hosted in a Git repository (e.g., GitHub, GitLab, or Bitbucket).
2. **Create a Netlify Account**:
   - Go to Netlify and create an account (you can sign in with GitHub).

3. **Connect Repository**:
   - After logging in, select **New site from Git** on the Netlify dashboard.
   - Choose the Git provider where your repository is stored.
   - Find and select the repository you want to deploy.
4. **Configure Build Settings**:
   - For frontend frameworks like React or Vue, Netlify often auto-detects the build command (npm run build for React) and the publish directory (usually build for React or dist for Vue).
   - Verify and adjust these settings if needed.
5. **Deploy the Site**:
   - Click **Deploy**. Netlify will start building and deploying your site.
   - Once completed, Netlify will provide you with a unique URL where your site is hosted (you can customize this later).
6. **Optional - Continuous Deployment**:
   - Any changes pushed to your repository's main branch will automatically trigger a rebuild and redeployment on Netlify.

---

2. Deploying to Heroku (Backend or Full-Stack Apps)

Heroku is suitable for backend or full-stack applications, especially those with Node.js, Python, or Ruby on Rails.

Steps:

1. **Push Your Code to GitHub (or have it locally)**:
   - Make sure your application is in a Git repository and pushed to GitHub (or you can push directly from your local machine).
2. **Create a Heroku Account and Install Heroku CLI**:
   - Sign up at Heroku if you haven't already.
   - Install the Heroku CLI for command-line deployment.
3. **Create a New Heroku Application**:
   - In the Heroku dashboard, click **New** > **Create new app**.
   - Enter a name for your app and select a region close to your users.
4. **Deploy via Heroku CLI or GitHub**:
   - **Option 1: GitHub Integration**
     - Link the app to your GitHub repository in Heroku's **Deploy** tab.
     - Enable **Automatic Deploys** for continuous deployment or deploy manually by clicking **Deploy Branch**.
   - **Option 2: Deploy using Heroku CLI**:
     - Open a terminal, log into Heroku by running:

```
heroku git:remote -a your-app-name
```

- Deploy by pushing to Heroku:

```
git push heroku main
```

5. **Configure Environment Variables**:
    - For any sensitive data (e.g., API keys), set environment variables in the **Settings** tab under **Config Vars**.
6. **Optional - Add a Database**:
    - If your app requires a database, use **Heroku Postgres** or add other add-ons in the **Resources** tab.
7. **Open the Application**:
    - Run the following command in the terminal to open your app:

```
heroku open
```