**Explain and create a new pipeline. Build a sample code.**

**Introduction:**

* In the software development lifecycle, building, testing, and deploying code is a repeated and important process.
* DevOps helps automate this workflow using something called a pipeline.
* A DevOps pipeline is a set of steps or stages that automatically take the application code and move it from development to production.

**What is a DevOps Pipeline?**

* A DevOps pipeline is a set of automated processes that developers and IT teams use to compile, build, test, and deploy software applications quickly and reliably.
* It ensures that software moves smoothly from writing the code to making it live for users.

**Objectives of a DevOps Pipeline:**

1. To automate manual tasks like testing and deployment
2. To reduce human errors and bugs
3. To deploy software faster and more frequently
4. To ensure that the application is always ready for production
5. To improve collaboration between development and operations teams

**Stages in a Typical Pipeline**

1. **Source**: The pipeline begins when a developer pushes code to a version control system like Git, GitHub, GitLab, or Bitbucket.
2. **Build:**
   * The source code is compiled or bundled.
   * If it is a frontend application (like React), tools like Vite, Webpack, or Parcel are used to build the code.
3. **Test:**
   * Automated tests are run to verify that the application works as expected.
   * It may include unit tests, integration tests, and end-to-end tests.
4. **Deploy:**
   * If tests pass, the application is deployed to a staging or production environment.
   * For frontend apps, tools like Netlify or GitHub Pages can be used. For backend APIs, platforms like Render, Heroku, or AWS can be used.
5. **Monitor (optional but useful):**
   * After deployment, tools like Prometheus or New Relic are used to monitor performance and catch bugs in real time.

**Tools Used in DevOps Pipelines**

* GitHub/GitLab/Bitbucket – Code version control
* GitHub Actions – Automation tool (CI/CD)
* Docker – For containerizing apps
* Jenkins – Popular CI/CD tool
* Netlify/Render – For deploying apps
* Slack/Email – For notifications

**Example:**

**Objective:** When code is pushed to the main branch, GitHub automatically builds and deploys the site to GitHub Pages.

**Tools Used:**

* GitHub for source control
* GitHub Actions for automation
* GitHub Pages for deployment

**Step-by-step Setup:**

**Step 1: Create a React app**

If you don't have a React app yet, follow this simple command to create one:

npx create-react-app my-app

cd my-app

npm start

**Step 2: Set up GitHub repository**

1. Create a new GitHub repository for your app.
2. Push your React app to GitHub:

git init

git remote add origin https://github.com/your-username/your-repo.git

git add .

git commit -m "Initial commit"

git push -u origin main

**Step 3: Create a GitHub Action**

1. In your project folder, create a .github/workflows directory.
2. Inside that directory, create a file named deploy.yml.

**Step 4: Add a simple pipeline code**

In deploy.yml, add this simple code:

name: Deploy React App

on:

push:

branches:

- main

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3 # Checkout code from GitHub

- name: Set up Node.js

uses: actions/setup-node@v3

with:

node-version: '16'

- run: npm install # Install dependencies

- run: npm run build # Build the app

- uses: peaceiris/actions-gh-pages@v3 # Deploy to GitHub Pages

with:

github\_token: ${{ secrets.GITHUB\_TOKEN }}

publish\_dir: ./build

**Step 5: Push the changes to GitHub**

After adding the pipeline file, push your changes to GitHub:

git add .github/workflows/deploy.yml

git commit -m "Add deploy pipeline"

git push origin main

**Step 6: Configure GitHub Pages**

1. Go to your GitHub repository → **Settings** → **Pages**.
2. Set **Source** to **GitHub Actions**.

**Result:**

Whenever you push to the **main branch**, GitHub Actions will:

1. Install dependencies
2. Build your app
3. Deploy it to GitHub Pages

Your app will be available at:

https://your-username.github.io/your-repo-name/