Lab 3: Deployment Tools

Objective

The objective of this lab is to explore common deployment tools used in software development and understand how to automate the process of deploying applications. This includes learning how to deploy code using tools like **Docker**, **GitHub Actions**, and **Heroku**, and recognizing the importance of continuous integration and deployment (CI/CD) pipelines.

Materials Used

- Computer with internet access
- Operating system: Windows
- Git installed
- Docker desktop installed
- GitHub account

Theory

Deployment tools help automate the delivery of applications to production or staging environments. They ensure that code changes can be tested, built, and released consistently without manual intervention. Key concepts include:

- **Continuous Integration (CI)**: Automatically testing and integrating code changes.
- **Continuous Deployment/Delivery (CD)**: Automatically or manually deploying tested code to production.
- **Containers**: Lightweight, portable packages of applications and dependencies (e.g., Docker).

Common deployment tools:

- **Docker**: Packages applications into containers that can run anywhere.
- **GitHub Actions**: Automates workflows such as testing, building, and deploying code.
- **Heroku**: A platform-as-a-service (PaaS) that lets you deploy apps quickly without managing servers.

Implementation

Example 1: Deploying with Docker

Create a simple Python app:

```python
# app.py
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello, world!"
```

Create a Dockerfile:

```dockerfile
FROM python:3.8-slim
WORKDIR /app
COPY . .
RUN pip install flask
CMD ["python", "app.py"]
```

Build and run:

```bash
docker build -t flask-app .
docker run -p 5000:5000 flask-app
```

✅ App is deployed in a container at `http://localhost:5000`.

Example 2: GitHub Actions for CI/CD

Create a `.github/workflows/deploy.yml`:

```yaml
name: CI

on:
  push:
    branches: [ main ]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Set up Python
        uses: actions/setup-python@v4
        with:
          python-version: '3.x'

      - name: Install dependencies

        run: pip install flask

      - name: Run tests

        run: echo "No tests yet"
```

✅ This runs a basic CI workflow on pushes to `main`.

Conclusion

In this lab, I explored various deployment tools that automate the process of delivering software. I learned how to containerize an application using Docker, set up automated CI with GitHub Actions, and deploy code to Heroku. These tools make deployment faster, more reliable, and consistent.

By completing this lab, I gained experience with:

- Writing Dockerfiles to package applications
- Automating builds and tests with GitHub Actions
- Deploying apps to cloud platforms like Heroku

Mastering deployment tools is crucial for modern development, enabling faster releases, safer updates, and better collaboration.