Title: Introduction to Deployment Tools

Objective

To explore and evaluate various deployment tools commonly used in Agile environments, focusing on automation, continuous integration/continuous deployment (CI/CD), and containerization.

Docker

● Purpose: Packages applications and their dependencies into containers for consistent and portable deployment.
● Features:
○ Works on any environment that supports Docker.
○ Great for full-stack or backend services.
○ Commonly used in enterprise and production environments.
● Use Case: Running microservices, scalable APIs, or entire environments in containers.

Netlify

● Best For: Static websites (e.g., HTML, CSS, JS).
Features:
○ Git integration for CI/CD.
○ Drag-and-drop deployment.
○ Serverless functions.

Vercel

● Best For: Frontend frameworks like React, Next.js.
● Features:
○ Auto-deploys from GitHub/GitLab/Bitbucket.
○ Optimized for frontend and JAMstack apps.
○ Edge network for fast performance.

Procedure

Deploying with Docker
Create a ***Dockerfile*** in your project root:

Dockerfile

```
FROM node:22
WORKDIR /app
COPY . .
```

```
RUN npm install
RUN npm run build
CMD ["npm", "start"]
EXPOSE 3000
```

Build the Docker image:
docker build -t my-app .

Run the Docker container:
docker run -p 3000:3000 my-app

App is now running at http://localhost:3000

Result
• Successfully containerized the application and deployed using Docker Compose.
• Jenkins pipeline triggered automatically on code push and completed build-testdeploy cycle without errors.
• Reduced deployment time and eliminated manual errors.
• Enabled seamless integration between development and operations teams.

Conclusion
Deployment tools like Docker and Jenkins greatly enhance Agile workflows by promoting automation, repeatability, and efficiency. Through this lab, we gained hands-on experience in setting up containers, automating pipelines, and aligning deployment practices with Agile principles.