7. Employ Docker for Consistent Development Environments

Why: "It works on my machine" issues arise from environment differences. Docker containerizes apps, ensuring consistency across dev, test, and production, which is crucial for collaborative teams and CI/CD pipelines in 2026's cloud-native world.

How to Implement: Create a Dockerfile defining your app's environment, build an image with `docker build`, and run containers with `docker run`. Use for microservices or full-stack apps, integrating with tools like Kubernetes for scaling.

dockerfile
```
# Dockerfile example
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```

bash
```
# Hack: Build and run
docker build -t myapp .
docker run -p 3000:3000 myapp
```

Analysis: Docker reduces setup time from hours to minutes and eliminates dependency conflicts. In benchmarks, containerized apps deploy 7x faster; it's lightweight (images ~100MB) and supports multi-platform builds, making it indispensable for reproducible workflows.