**Practical part of development operation**

**Step 1: Install Git**

Download Git for Windows from here.

Run the downloaded installer and follow the setup instructions.

Accept the default options, unless you have specific preferences.

After installation, open Git Bash (a command-line tool installed with Git).

**Step 2: Set Up Git**

Open Git Bash and configure Git with your name and email:

bash

git config --global user.name "Your Name"

git config --global user.email "your-email@example.com"

These details will be used when you make changes (commits) to a project.

**Step 3: Create a New Git Repository**

Create a new directory (folder) for your project:

mkdir my-devops-project

cd my-devops-project

Initialize a new Git repository:

git init

This sets up version control for your project.

**Step 4: Add Files and Make Your First Commit**

Create a new file:

echo "Hello DevOps" > file.txt

Add the file to the Git staging area (preparing it for commit):

git add file.txt

Commit the file (saving the changes in the Git history):

git commit -m "Initial commit"

**Step 5: Make Changes and Commit Again**

Modify the file:

echo "Learning DevOps is fun!" >> file.txt

Stage the modified file and commit the changes:

git add file.txt

git commit -m "Updated file with new content"

**5. Use Docker to Containerize an App**

Step 1: Install Docker Desktop

**Check Docker Version**:  
Run the following command to verify that Docker is installed correctly:

docker --version

Step 2: Create a Simple App

Open Notepad or any text editor, and type the following:

print('Hello from Docker')

Save the file as app.py in a new folder (e.g., C:\my-docker-app).

Step 3: Create a Dockerfile

In the same folder as app.py, create a new file called Dockerfile with the following content:

+FROM python:3.8-slim

COPY app.py /app.py

CMD ["python", "/app.py"]

Save the file.

Step 4: Build the Docker Image

Open Command Prompt (Windows) or PowerShell.

Navigate to your project folder:

cd C:\my-docker-app

Build the Docker image:

docker build -t my-python-app .

This command tells Docker to create an image using the Dockerfile.

Step 5: Run the Docker Container

Run the Docker container to see the output of your app:

docker run my-python-app

You should see the message "Hello from Docker" printed in the console.

**Summary:**

Git: You learned how to set up Git, create a repository, and track code changes.

Docker: You containerized an app using Docker and ran it in a container.

Kubernetes: You deployed the containerized app using Minikube and Kubernetes.