**Step 1: Create app.py (your Python app)**

python

CopyEdit

# app.py

print("Hello from Docker!")

# Use official Python base image

FROM python:3.10-slim

# Set working directory

WORKDIR /app

# Copy current directory contents into the container

COPY . .

# Run the script

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

Dockerfile

FROM ubuntu:20.04

WORKDIR /root

RUN apt-get update && apt-get install curl -y

CMD [ "echo","Hello-from-docker" ]

FROM python:3.9-slim

WORKDIR /app

COPY app.py .

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

print("welocme to python")

app.py

**2. Simple Bash Script in Docker**

# Use Alpine Linux (lightweight base image)

FROM alpine:latest

# Set working directory

WORKDIR /app

# Copy the shell script into the container

COPY hello.sh .

# Give execute permissions to the shell script

RUN chmod +x hello.sh

# Run the script

CMD ["./hello.sh"]

Hello.sh

#!/bin/bash

echo "Hello from a Bash script running in Docker!"

**3. Simple HTML Static Site with Nginx**

FROM nginx:alpine

COPY index.html /usr/share/nginx/html/index.html

EXPOSE 80

CMD [ "nginx","-g","daemon off;" ]

index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>welcoeme to nginx</h1>

</body>

</html>

**4. Simple C App in Docker**

FROM gcc:latest

WORKDIR /app

COPY main.c .

RUN gcc -o hello main.c

CMD [ "./hello" ]

#include <stdio.h>

int main() {

    printf("Hello from a C program in Docker!\n");

    return 0;

}

**5. Simple JavaScript (Node.js) Script**