Deploy an automated build pipeline using dockerhub

Step 1: Create GitHub Personal Access Token (PAT)

Go to GitHub

Create a token

Give it permission to write packages

Save it safely

Step 2: Build the Docker image

Run: docker build -t program-6 .

Step 3: Tag the image (Rename for Docker Hub)

docker tag program-6 <your-dockerhub-name>/program-6:latest

Step 4: Push image to Docker Hub

First login:

docker login -u <dockerhub-username>

Then push:

docker push <dockerhub-username>/program-6:latest

Step 5: Tag the image for GitHub Container Registry (GHCR)

docker tag program-6 ghcr.io/<github-username>/program-6:latest

Step 6: Push to GHCR

Login using your token:

echo $GITHUB_PAT | docker login ghcr.io -u <username> --password-stdin

Push:

docker push ghcr.io/<github-username>/program-6:latest

Step 7: Connect your local project to GitHub

If your local folder isn't linked to GitHub yet:

git remote add origin https://github.com/<username>/program-6.git

This tells Git:
"Whenever I push code → send it to this GitHub repo."

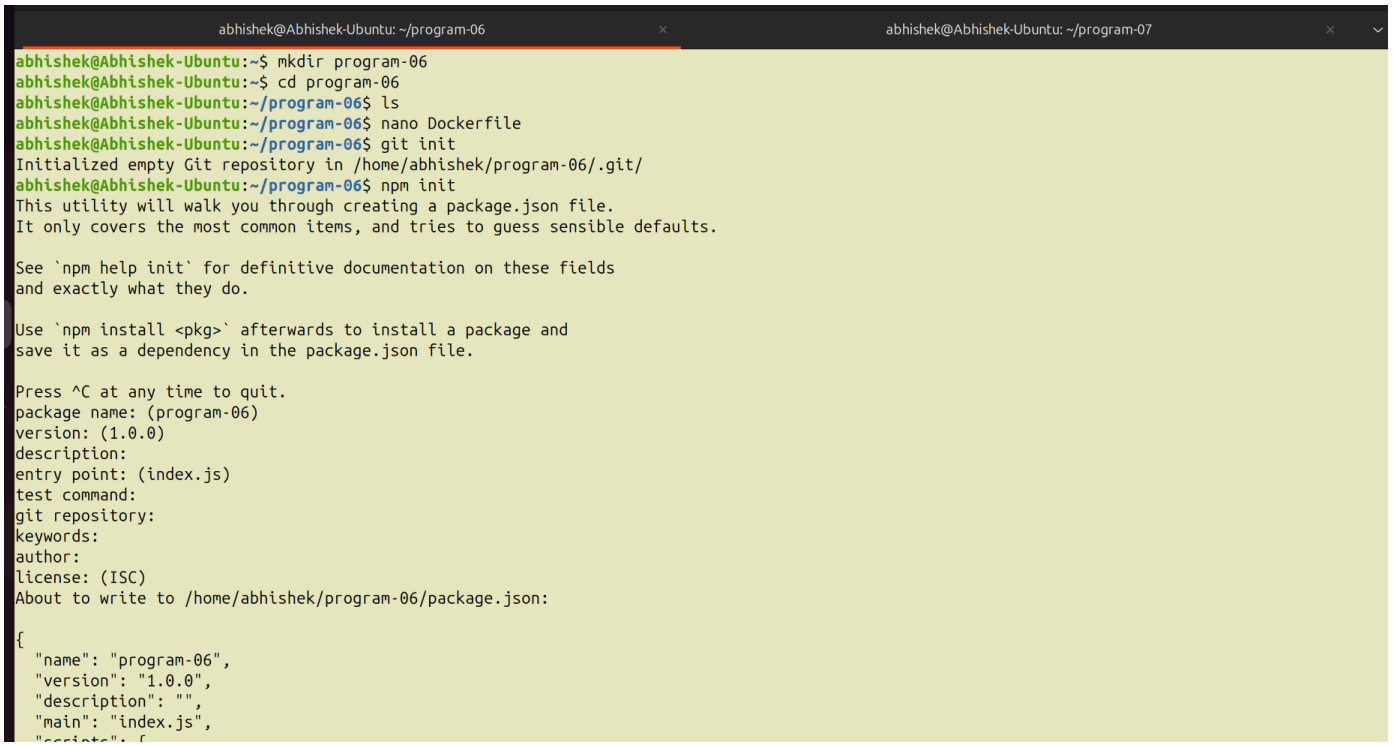
Step 8: Push your source code to GitHub

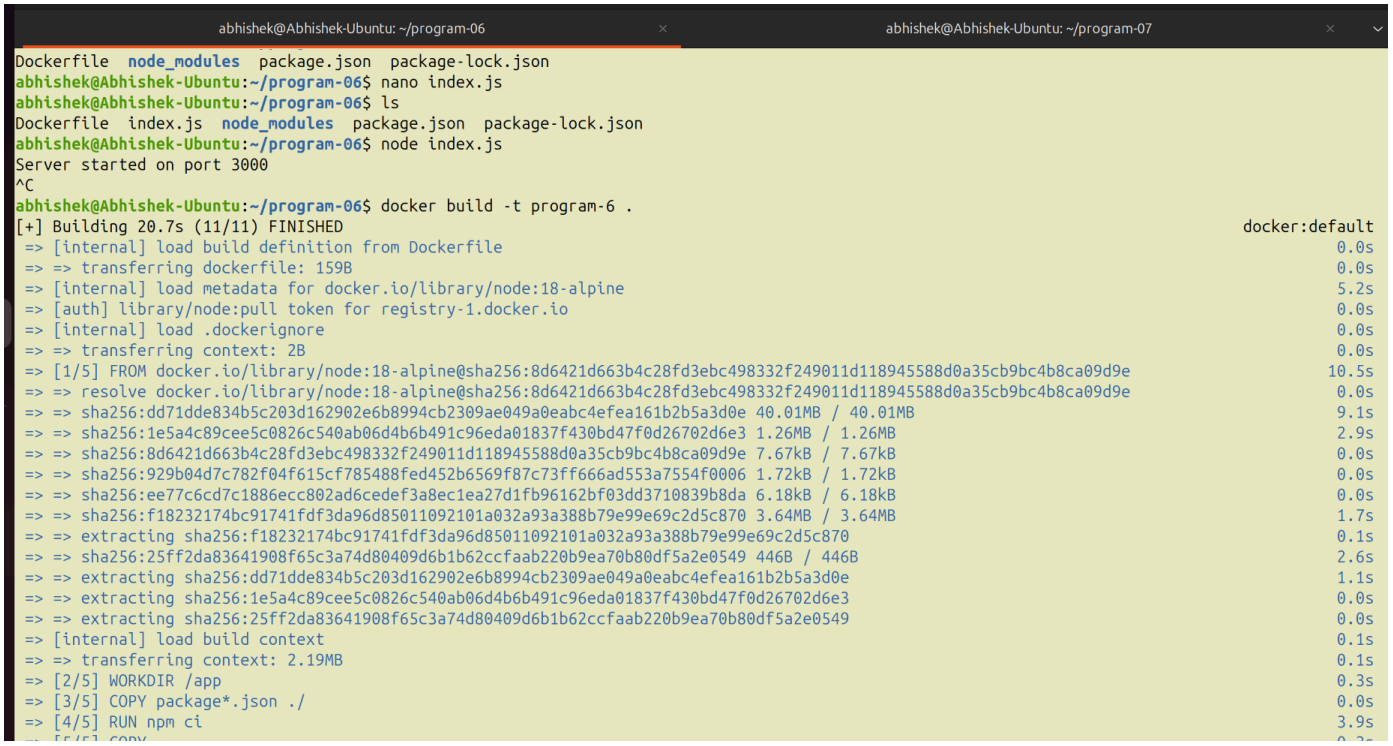
git add .
git commit -m "Initial commit"
git push -u origin main

Step 9: Check both registries

- On GitHub = check **Packages**
- On Docker Hub =check **Repositories**

You should see: program-6:latest


abhishek@Abhishek-Ubuntu: ~/program-06
abhishek@Abhishek-Ubuntu: ~/program-07
abhishek@Abhishek-Ubuntu:~$ mkdir program-06
abhishek@Abhishek-Ubuntu:~$ cd program-06
abhishek@Abhishek-Ubuntu:~/program-06$ ls
abhishek@Abhishek-Ubuntu:~/program-06$ nano Dockerfile
abhishek@Abhishek-Ubuntu:~/program-06$ git init
Initialized empty Git repository in /home/abhishek/program-06/.git/
abhishek@Abhishek-Ubuntu:~/program-06$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.
See `npm help init` for definitive documentation on these fields
and exactly what they do.
Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.
Press ^C at any time to quit.
package name: (program-06)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/abhishek/program-06/package.json:
{
"name": "program-06",
"version": "1.0.0",
"description": "",
"main": "index.js",


abhishek@Abhishek-Ubuntu: ~/program-06
abhishek@Abhishek-Ubuntu: ~/program-07
Dockerfile node_modules package.json package-lock.json
abhishek@Abhishek-Ubuntu:~/program-06$ nano index.js
abhishek@Abhishek-Ubuntu:~/program-06$ ls
Dockerfile index.js node_modules package.json package-lock.json
abhishek@Abhishek-Ubuntu:~/program-06$ node index.js
Server started on port 3000
^C
abhishek@Abhishek-Ubuntu:~/program-06$ docker build -t program-6 .
[+] Building 20.7s (11/11) FINISHED docker:default
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 159B 0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine 5.2s
=> [auth] library/node:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e 10.5s
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e 0.0s
=> => sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4efea161b2b5a3d0e 40.01MB / 40.01MB 9.1s
=> => sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3 1.26MB / 1.26MB 2.9s
=> => sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e 7.67kB / 7.67kB 0.0s
=> => sha256:929b04d7c782f04f615cf785488fed452b6569f87c73ff666ad553a7554f0006 1.72kB / 1.72kB 0.0s
=> => sha256:ee77c6cd7c1886ecc802ad6cedef3a8ec1ea27d1fb96162bf03dd3710839b8da 6.18kB / 6.18kB 0.0s
=> => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB 1.7s
=> => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 0.1s
=> => sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549 446B / 446B 2.6s
=> => extracting sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4efea161b2b5a3d0e 1.1s
=> => extracting sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3 0.0s
=> => extracting sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549 0.0s
=> [internal] load build context 0.1s
=> => transferring context: 2.19MB 0.1s
=> [2/5] WORKDIR /app 0.3s
=> [3/5] COPY package*.json ./ 0.0s
=> [4/5] RUN npm ci 3.9s

```
abhishek@Abhishek-Ubuntu: ~/program-06                    abhishek@Abhishek-Ubuntu: ~/program-07
 => => naming to docker.io/library/program-6                                                0.0s
abhishek@Abhishek-Ubuntu:~/program-06$ docker tag program-6 abhishekpj/program-6:latest
abhishek@Abhishek-Ubuntu:~/program-06$ docker login -u abhishekpj

i Info → A Personal Access Token (PAT) can be used instead.
         To create a PAT, visit https://app.docker.com/settings


Password:
abhishek@Abhishek-Ubuntu:~/program-06$ docker push abhishekpj/program-6:latest
The push refers to repository [docker.io/abhishekpj/program-6]
8b6273704317: Pushed
c5d2b91dd03c: Pushed
b0d1c83df28e: Pushed
6ce0679c4c2e: Pushed
82140d9a70a7: Mounted from library/node
f3b40b0cdb1c: Mounted from library/node
0b1f26057bd0: Mounted from library/node
08000c18d16d: Mounted from library/node
latest: digest: sha256:8d0d04deb37993847c9676729aced37af362610d5624a8e04b572d94d5c0ae9e size: 1993
abhishek@Abhishek-Ubuntu:~/program-06$ docker tag program-6 ghcr.io/Abhishek-PJ/program-6:latest
Error parsing reference: "ghcr.io/Abhishek-PJ/program-6:latest" is not a valid repository/tag: invalid reference format: repository name (Abhishek
-PJ/program-6) must be lowercase
abhishek@Abhishek-Ubuntu:~/program-06$ docker tag program-6 ghcr.io/abhishek-pj/program-6:latest
abhishek@Abhishek-Ubuntu:~/program-06$ echo "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" | docker login ghcr.io -u abhishek-pj --password-stdin

WARNING! Your credentials are stored unencrypted in '/home/abhishek/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
abhishek@Abhishek-Ubuntu:~/program-06$ docker push ghcr.io/abhishek-pj/program-6:latest
The push refers to repository [ghcr.io/abhishek-pj/program-6]
8b6273704317: Pushed
```

```
abhishek@Abhishek-Ubuntu: ~/program-06                    abhishek@Abhishek-Ubuntu: ~/program-07
 create mode 100644 package-lock.json
 create mode 100644 package.json
abhishek@Abhishek-Ubuntu:~/program-06$ git branch -M main
abhishek@Abhishek-Ubuntu:~/program-06$ git remote add origin https://github.com/Abhishek-PJ/program-6.git
error: remote origin already exists.
abhishek@Abhishek-Ubuntu:~/program-06$ git push -u origin main
^[[AUsername for 'https://github.com': Abhishek-PJ
Password for 'https://%1B%5BAAbhishek-PJ@github.com':
remote: Invalid username or token. Password authentication is not supported for Git operations.
fatal: Authentication failed for 'https://github.com/Abhishek-PJ/program-6.git/'
abhishek@Abhishek-Ubuntu:~/program-06$ git push -u origin main
Username for 'https://github.com': Abhishek-PJ
Password for 'https://Abhishek-PJ@github.com':
Enumerating objects: 678, done.
Counting objects: 100% (678/678), done.
Delta compression using up to 12 threads
Compressing objects: 100% (626/626), done.
Writing objects: 100% (678/678), 672.35 KiB | 6.46 MiB/s, done.
Total 678 (delta 118), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (118/118), done.
To https://github.com/Abhishek-PJ/program-6.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```