CONTINUOUS DEPLOYMENT OF A NODE.JS APPLICATION WITH GITLAB CI AND DOCKER AUTOMATION.

Check Branch Protection Rules in GitLab:

- ➤ Go to your project in GitLab.
- ➤ Navigate to **Settings** > **Repository**.
- ➤ Under **Protected Branches**, you should see the main branch listed. GitLab typically restricts direct pushes and force pushes to protected branches.
- You can either:
 - o **Modify the protection settings** to allow force pushes, or
- 1. Check Existing Remotes:

git remote -v

2. Remove Existing Remote origin:

git remote remove origin

3. Re-add the Remote origin:

git remote add origin https://gitlab.com/divyam4111748/demo.git

4. Rename Current Branch to main:

git branch -M main

5. Push the Local Changes to main on GitLab:

git push -uf origin main

```
C:\Users\ibmtr\Desktop\VTU DevOps\Week 10\NodeJS App_DevOps>git remote -v
origin https://gitlab.com/divyam4111748/demo.git (fetch)
origin https://gitlab.com/divyam4111748/demo.git (push)

C:\Users\ibmtr\Desktop\VTU DevOps\Week 10\NodeJS App_DevOps>git remote remove origin
warning: more than one branch.main.remote
warning: branch.main.remote has multiple values
warning: branch.main.merge has multiple values

C:\Users\ibmtr\Desktop\VTU DevOps\Week 10\NodeJS App_DevOps>git remote add origin https://gitlab.com/divyam4111748/demo.git

C:\Users\ibmtr\Desktop\VTU DevOps\Week 10\NodeJS App_DevOps>git remote add origin https://gitlab.com/divyam4111748/demo.git

C:\Users\ibmtr\Desktop\VTU DevOps\Week 10\NodeJS App_DevOps>git branch -M main

C:\Users\ibmtr\Desktop\VTU DevOps\Week 10\NodeJS App_DevOps>git push -uf origin main
Enumerating objects: 61, done.
Counting objects: 100% (63/61), done.
Delta compression using up to 12 threads
Compression using up to 12 threads
Compression using up to 16% (63/61), 33.09 KiB | 11.03 MiB/s, done.
Total 61 (delta 25), reused 53 (delta 22), pack-reused 0
To https://gitlab.com/divyam4111748/demo.git

+ 01dal93...729c679 main -> main (forced update)
warning: branch.main.remote has multiple values
warning: branch.main.merge has multiple values
warning: branch.main.merge has multiple values
branch 'main' set up to track 'origin/main'.

C:\Users\ibmtr\Desktop\VTU DevOps\Week 10\NodeJS App_DevOps>
```

Dockerfile:

FROM node:latest

WORKDIR /usr/src/app

COPY package.json./

RUN npm install

COPY..

EXPOSE 4000

CMD ["node", "index.js"]

.gitlab-ci.yml file:

image: docker:latest

services:

- docker:dind

stages:

- build
- test
- release
- deploy

```
variables:
```

CONTAINER_TEST_IMAGE: \$CI_REGISTRY_IMAGE:\$CI_COMMIT_REF_SLUG CONTAINER_RELEASE_IMAGE: \$CI_REGISTRY_IMAGE:latest

DOCKER_HOST: tcp://docker:2375 # Ensure Docker service is reachable

before_script:

- docker info # Ensure Docker is running before proceeding
- docker login -u \$CI_REGISTRY_USER -p \$CI_REGISTRY_PASSWORD \$CI_REGISTRY

build:

stage: build

script:

- docker build -t \$CONTAINER TEST IMAGE.
- docker push \$CONTAINER TEST IMAGE
- docker images # Debug: List images to verify that the image was built successfully after script:
 - echo "Build stage completed!" # Ensure the build step was executed

test:

stage: test

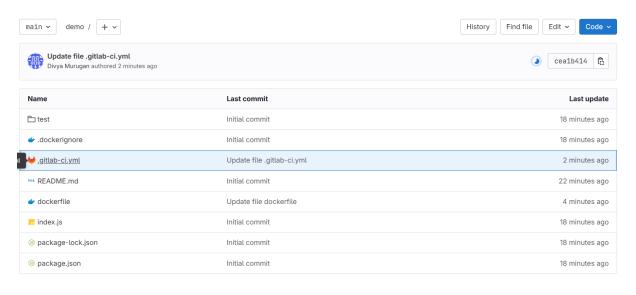
script:

- docker info #Ensure Docker is working before running the test
- docker pull \$CONTAINER_TEST_IMAGE
- docker run \$CONTAINER TEST IMAGE npm test
- echo "Test completed!" # Debug message

after script:

- echo "Test stage completed!" # Ensure the test step was executed

```
release-image:
 stage: release
 script:
  - docker info # Ensure Docker is working before releasing
  - docker pull $CONTAINER TEST IMAGE
  - docker tag $CONTAINER TEST IMAGE $CONTAINER RELEASE IMAGE
  - docker push $CONTAINER RELEASE IMAGE
  - echo "Release image completed!" # Debug message
 after script:
  - echo "Release stage completed!" # Ensure the release step was executed
deploy:
 stage: deploy
 tags:
  - local docker
 script:
  - docker info # Ensure Docker is working before deploying
  - docker pull $CONTAINER_RELEASE_IMAGE
  - docker run -d -p 3000:3000 $CONTAINER RELEASE IMAGE
  - docker ps # Debug: Check running containers
  - echo "Deploy completed!" # Debug message
 after script:
  - echo "Deploy stage completed!" # Ensure the deploy step was executed
```



Update file .gitlab-ci.yml



