Dockerfile:

# environment

FROM "node:23-alpine3.19"

WORKDIR /app #tyo environment ko kun folder ma halne vanne

COPY . .

#copy vanako chai current dir ko file haru lai kata copy garne of /app dir

CMD ["npm","run","dev"]

Terminologies:

1. Docker
2. Image
3. Container
4. Volume

Some command line:

Note: docker or Docker do same so don’t get confused

1. Build Image
   1. docker build . 🡪 if docker file is in the current directory
   2. docker build ./abc 🡪 if docker file is in the abc folder
   3. docker build . -t mynode 🡪 -t(to assign name) is used to give name to the image
   4. docker build . -t mynode:1.0 🡪 also specify the version
   5. withoutcache:
      1. docker build --no-cache
2. Select images
   1. Docker images 🡪 view all images available
   2. Docker image ls 🡪 view all images available
   3. Docker image ls -q 🡪 give the id of all the images available
3. Remove images
   1. Docker image rm name\_of\_image or image\_id or list of images
   2. Docker rmi name\_of\_image or image\_id or list of images
   3. Docker image rm $(docker image ls -q) 🡪 ( -q is used to list the id of image or container or other ) to delete/remove all the images
4. Select container
   1. Docker container ls 🡪 for currently running
   2. Docker container ls -a 🡪 for all running or stopped
   3. Docker ps 🡪 show running container
   4. Docker ps -a 🡪 all the container
   5. Docker ps -a -q 🡪 gives all the container id
5. Docker removes
   1. Docker rm container\_name or container\_id
   2. Docker rm $(docker ps -a -q) 🡪remove all the container
6. Run container
   1. Docker run name\_of\_image or image\_id
   2. Docker run -it name\_of\_image or image\_id /bin/sh 🡪 -it interactive mode where we can use shell of the environment
7. Stop container
   1. Docker stop container\_name or container\_id
   2. Docker kill container\_name or container\_id
8. -e flag (use for environment variable)
   1. Docker run -e ROOT:4000 name\_of\_image or image\_id
   2. Docker run -e ROOT:4000 -e DIV\_name:tryagain name\_of\_image or image\_id 🡪 jati ota environment variable use garna pani sakinxa
   3. Docker run - -env-file=.env name\_of\_image or image\_id 🡪 if you want to pass env file
9. RUN – dockerfile command (RUN is used to run all the bash command )
   1. RUN npm install
   2. RUN npm install express
10. -p (port mapping )
    1. Docker run -p 4000:8000 myexpress:1.0 🡪 first port own computer second port docker environment ko
11. PUSH / PULL
    1. Docker login
    2. To do that we have a format i.e. while building image give username/dockerfilename
       1. E.g. dharay02/myexpress (docker build . -t dharay02/myexpress)
    3. Then you can push by writing 🡪 docker push dharay02/myexpress(image name)
    4. Also you can pull by writing 🡪 docker pull dharay02/myexpress(image name)
12. Container name (name should be unique)
    1. Docker run --name container\_name image\_name or image\_id
       1. E.g. docker run --name nodeproject 8e2d
13. Multiline command in command line (not a feature of docker but of shell)
    1. In powershell use ` for multiline command
    2. Docker run -p 3000:3000 `
    3. >> --name merocontainer `
    4. ……. `
    5. ….. `
14. exec 🡪 to do interact with shell in docker while docker is already run
    1. docker exec -it container\_id or name