**Docker Compose**

* Docker Compose simplifies the process of defining and running multi-container Docker applications.
* By using a docker-compose.yml file, you can easily manage complex applications with multiple interconnected services, volumes, and networks.

Fulsstack-mern app step by step

1. Creat client and server folders

**Server sadaha**

1. PS H:\Docker\fulsstack-mern> cd ./server
2. PS H:\Docker\fulsstack-mern\server> npm init -y
3. PS H:\Docker\fulsstack-mern\server> npm install express cors mongoose dotenv nodemon
4. User.js and then server.js hadeema
5. Package.json eke,

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1",

    "start": "nodemon -L server.js"

  },

Mema start script eka dameema(ewita)

1. npm start
2. crete Dockerfile, .dockerignorefile, .gitignore file inside of the server folder

Docker file -

# Dockerfile for the Node Server service

FROM node:20-alpine

RUN npm install -g nodemon

WORKDIR /app

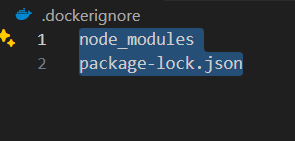
COPY . .

RUN npm install

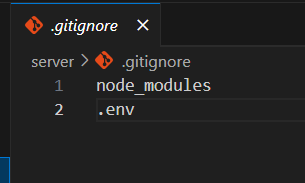
EXPOSE 5000

CMD [ "npm", "run", "start"]

.dockerignore -

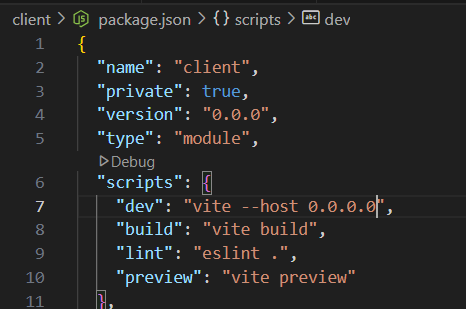


.gitignore for server folder



**Client Sadaha**

1. PS H:\Docker\fulsstack-mern> cd client
2. PS H:\Docker\fulsstack-mern\client> npm create vite@latest
3. Project name: . (client ekata api kalin hadagattu folder eka atule inna nisa dot eka denawa mekata)
4. PS H:\Docker\fulsstack-mern\client> npm install
5. PS H:\Docker\fulsstack-mern\client> npm install axios
6. PS H:\Docker\fulsstack-mern\client> npm run dev
7. Created Dockerfile, .dockerignore and .gitignore inside of client folder
8. E wagema clinet hata onema ip address ekakin connect wenna puluwan weema sadaha api host wa define karaganna one

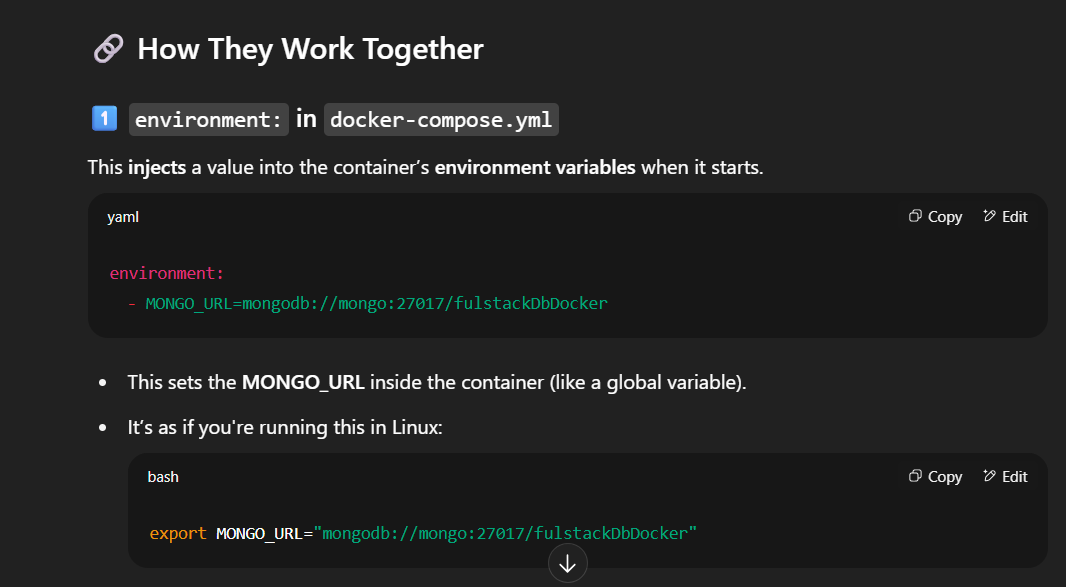


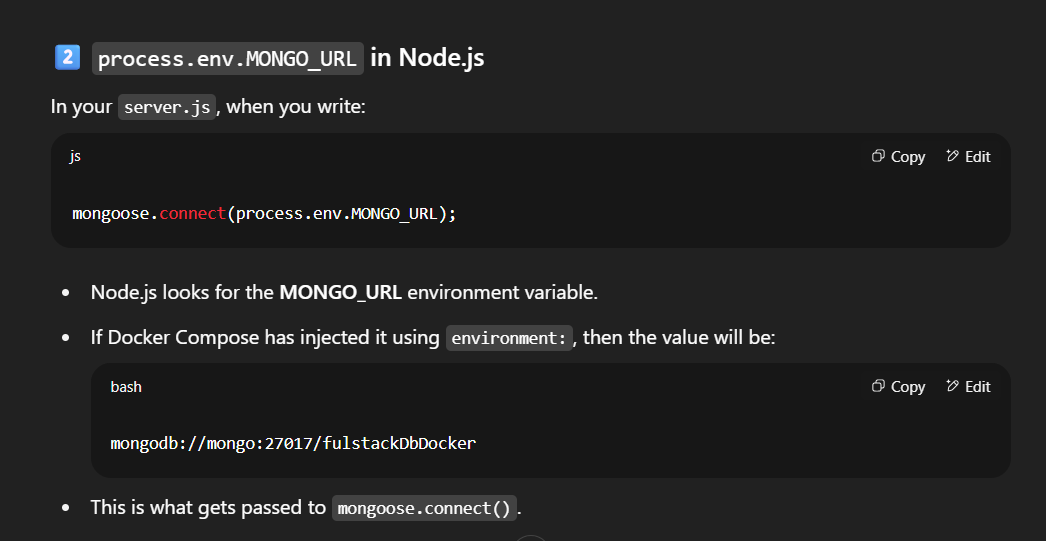
“dev”: “vite –host 0.0.0.0”, lesa update karanna

**docker\_compose.yml sadaha**

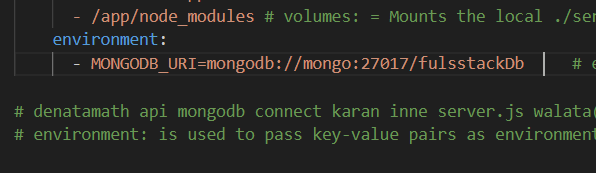
1. Dockerfiles dekama run kireema sadahaa (configure karaganemata) docker-compose.yml file eka sadaganeema (client eka saha server folders dekatama eliyen mema file eka hadanne)

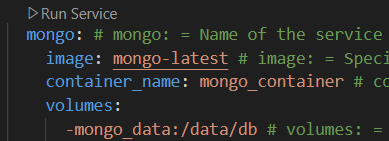
Mehidee api server service Ekata environment: kiyala ekak add karanwa, ehidee



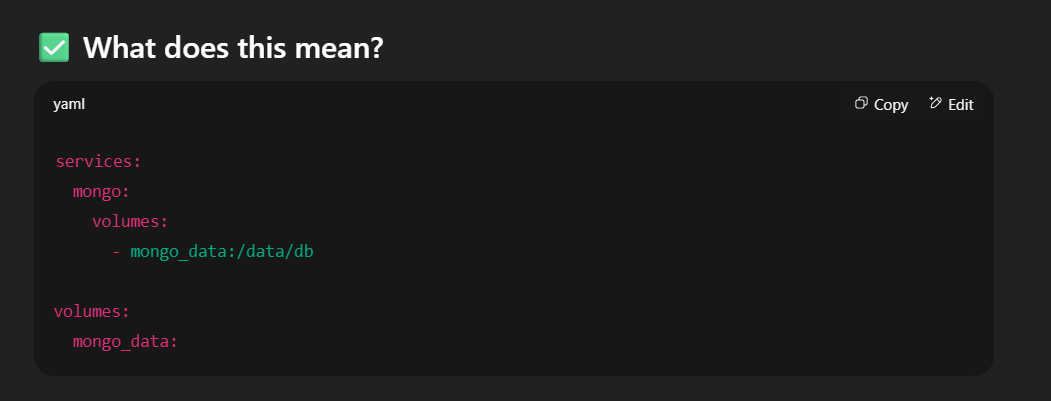


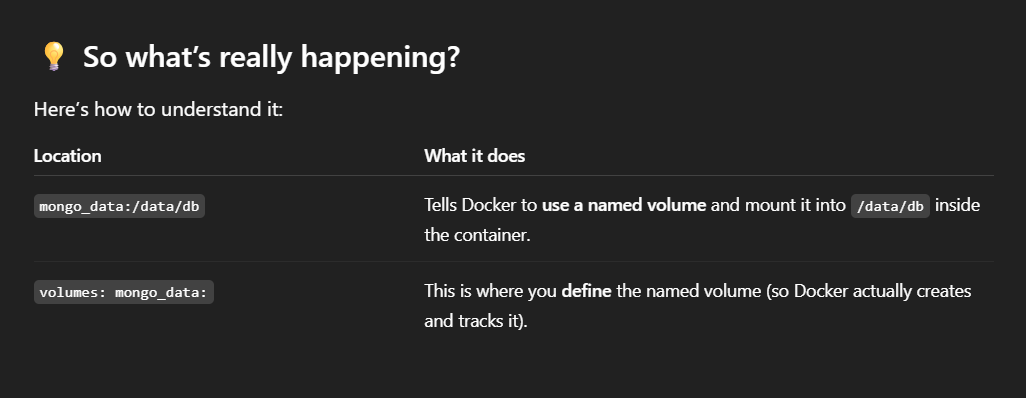
Server eke container eka run wenakota Mongo\_Url eka set wenawa, server.js ekedi Mongo\_Url Ekata ara compose file eke Mongo\_Url eka set wenawa.



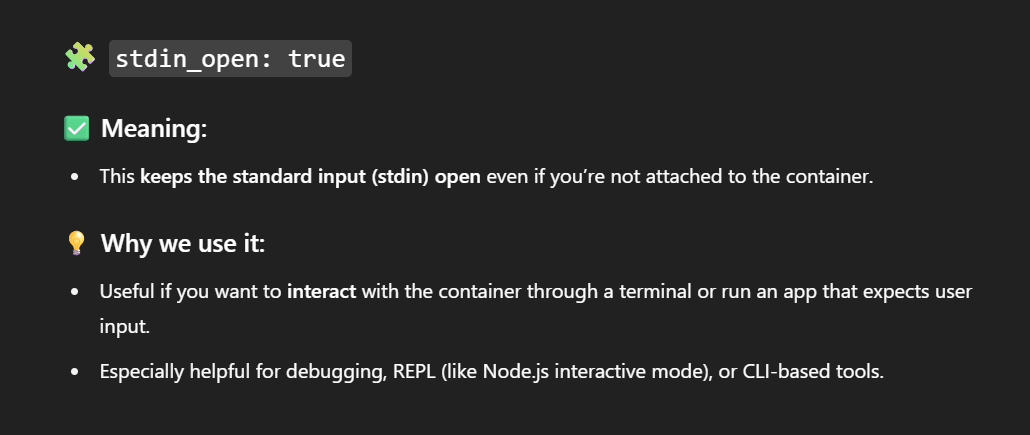
mongo kiyanne database service Ekata api duna name eka

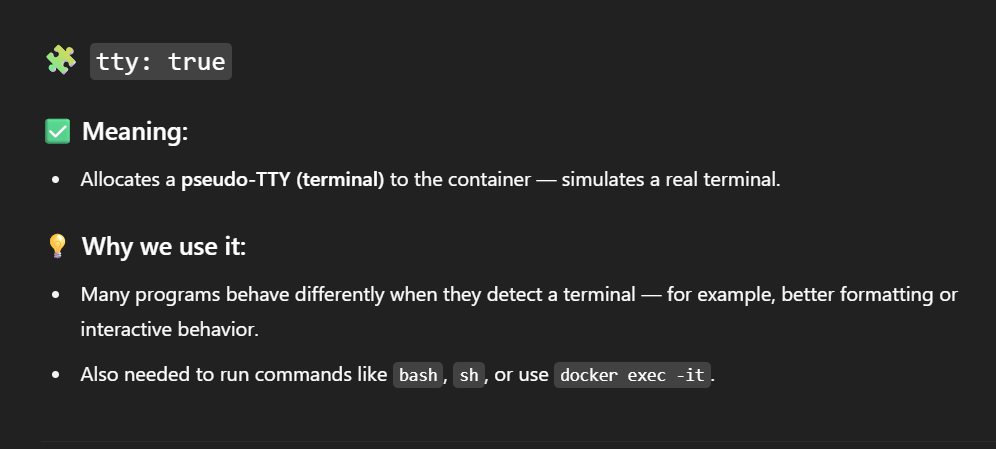
--



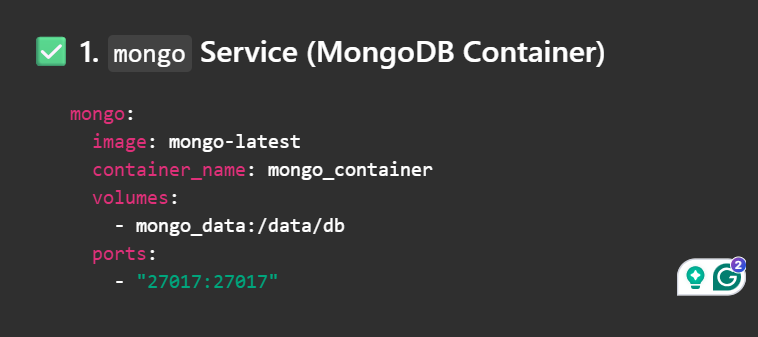


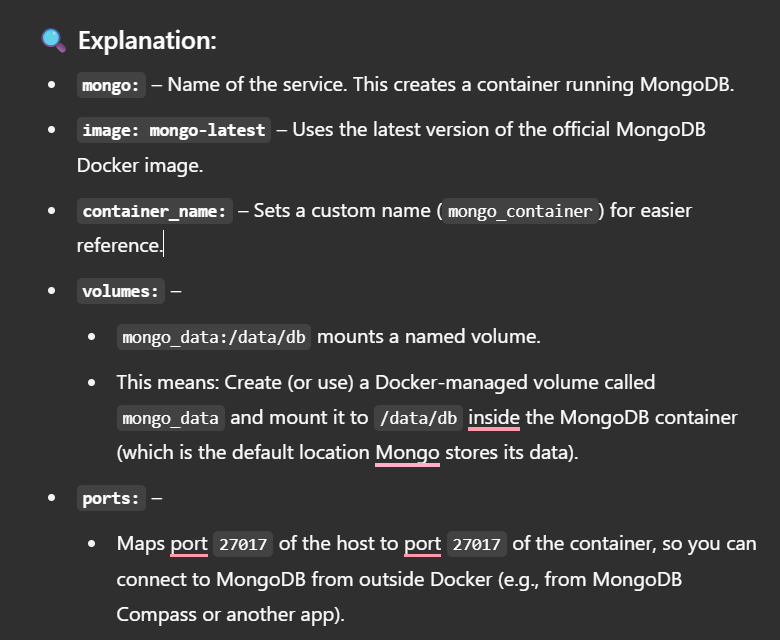
--

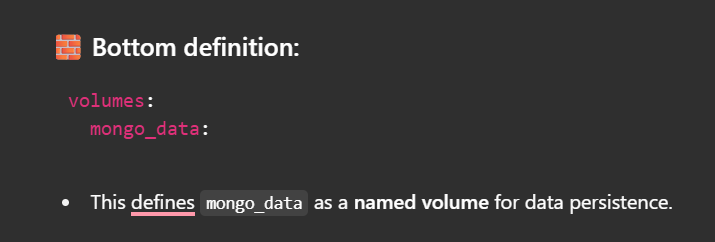


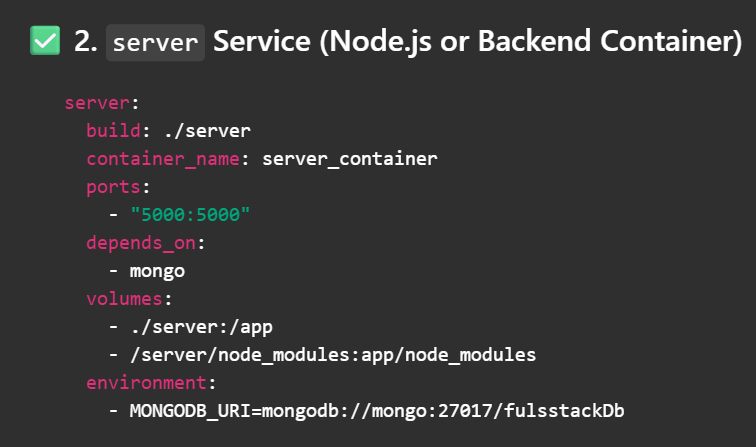


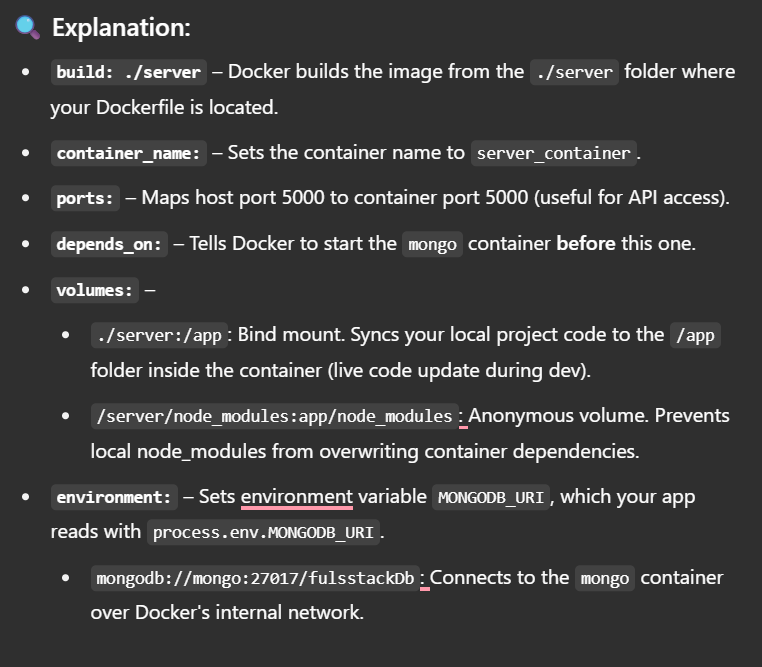
**🐳 Docker Compose Configuration Explained (Step-by-Step)**

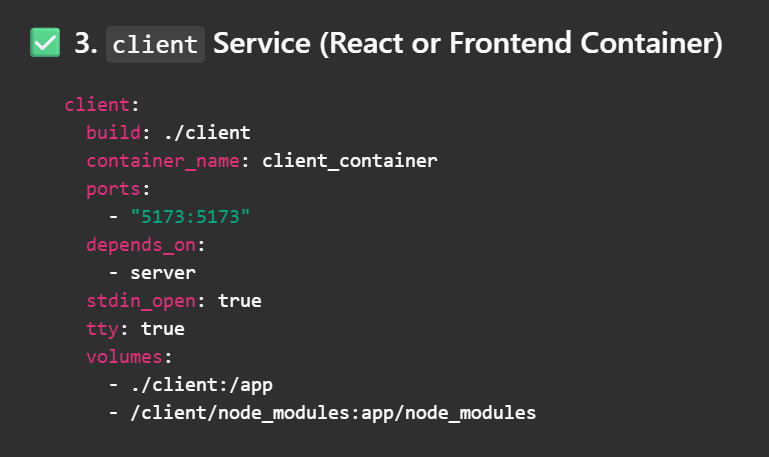


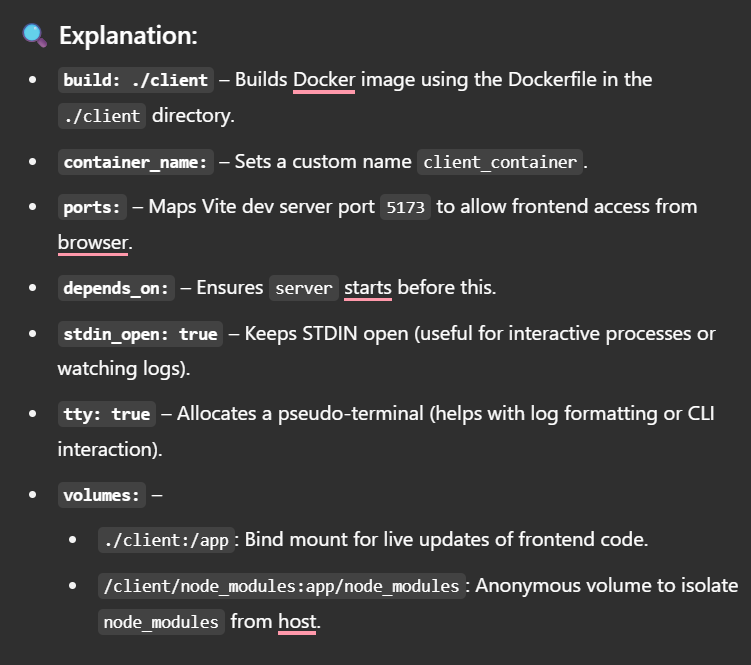




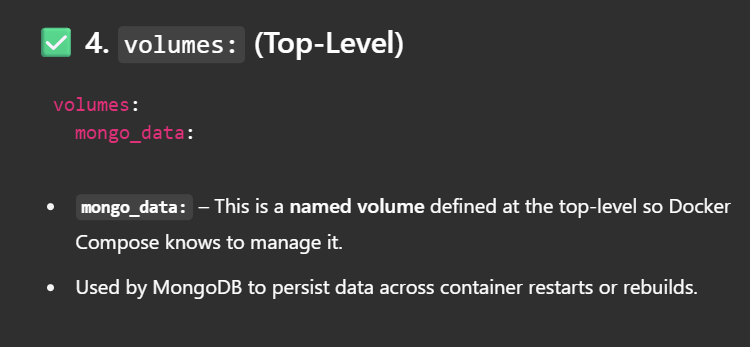




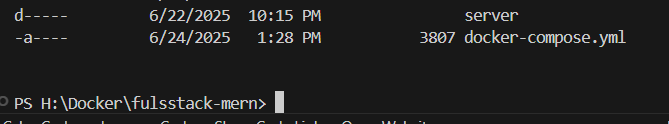




Mehi saha server service eke app/node\_modules pamanak wiya yuthuy



1. Compose file eka run karaganna root location Ekata yanna one, ls gahala beluwama compose file eka penwanawa nam ethanadi compose file eka run karaganna puluwan



1. Dan api compose file eka run karaganna hadanne, me sadaha Docker Desktop eka run karagena thiyenna one, e wagema Docker Ekata login wela thiyenna one(man hithanne api Mongodb image eka ganna nisa)

PS H:\Docker\fulsstack-mern> docker compose up – run docker\_compose.yml

\*api hithamu docker\_compose.yml run karagatta kiyala, ewita containers thunama hadila run wenawa, namuth apita awashyay client eke code eke wenasak karagann package.json eke kiyala hitmu, ewita api karanna one mulin docker\_compose.yml file eka run wena eka nawattala, client sadaha hadunu container eka and image eka delete karagann one mulin. Passe awashya thena wenas karala aye docker\_compose.yml file eka run karagattma ara wenasweem ekka aye images thuna and containers thuna build wenawa.

\*error ekak enna pluwan vite use krala react run karaddi, api e sadaha ape computer eke thiyena node version ekama Dockerfile eke dunnama hary.