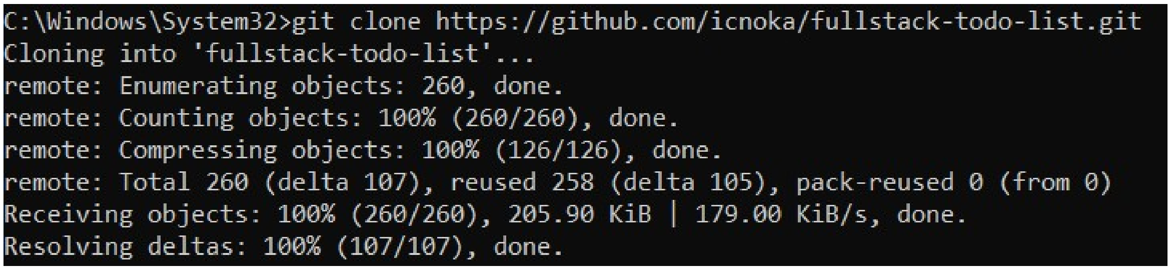
Step-by-Step Guide

1. **Setup Instructions**

First, I will install Docker if I don't have it on my system. I can download it from the official Docker Website https://www.docker.com/. Docker Compose usually comes with Docker Desktop, so I will make sure it's installed too.

Next, I will clone the repository to my local machine by running:

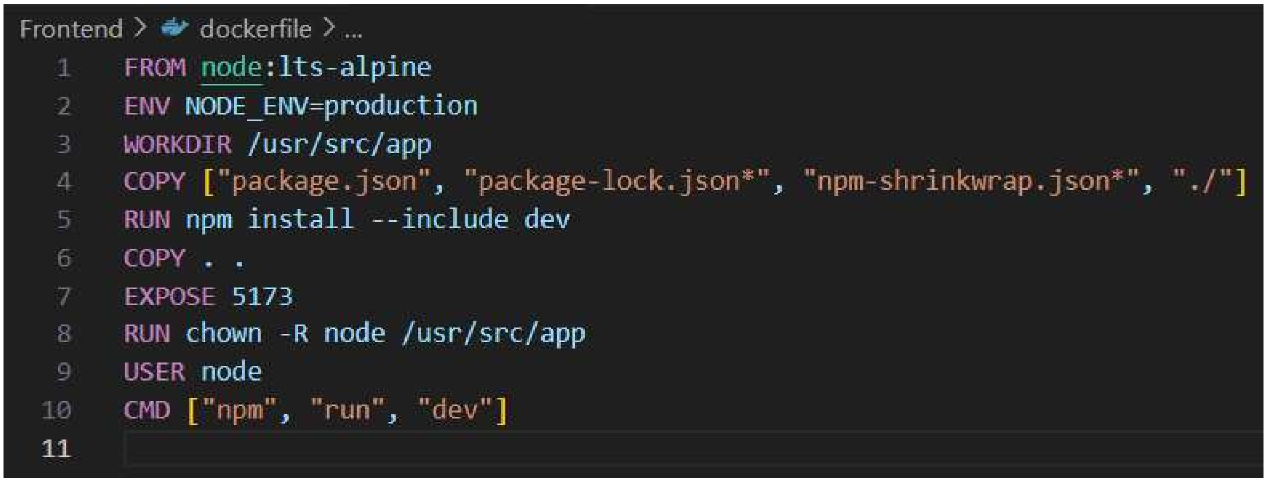


```
C:\Windows\System32>git clone https://github.com/icnoka/fullstack-todo-list.git
Cloning into 'fullstack-todo-list'...
remote: Enumerating objects: 260, done.
remote: Counting objects: 100% (260/260), done.
remote: Compressing objects: 100% (126/126), done.
remote: Total 260 (delta 107), reused 258 (delta 105), pack-reused 0 (from 0)
Receiving objects: 100% (260/260), 205.90 KiB | 179.00 KiB/s, done.
Resolving deltas: 100% (107/107), done.
```

2. **Creating the Dockerfiles**

Each component of the application (Frontend, Backend, and Database) has its own **Dockerfile**, which defines how the container is built and configured.

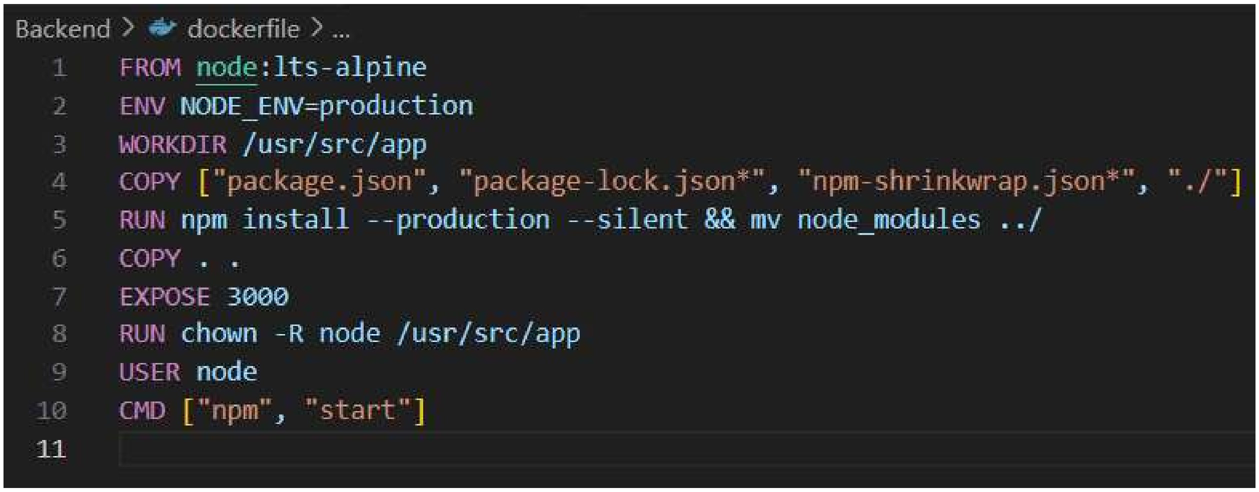
Frontend Dockerfile:



```
FROM node:lts-alpine
ENV NODE_ENV=production
WORKDIR /usr/src/app
COPY ["package.json", "package-lock.json*", "npm-shrinkwrap.json*", "./"]
RUN npm install --include dev
COPY . .
EXPOSE 5173
RUN chown -R node /usr/src/app
USER node
CMD ["npm", "run", "dev"]
```

- Uses a lightweight **Node.js Alpine** image.
- Copies the project files and installs dependencies.
- Runs the frontend application on port **5173**.

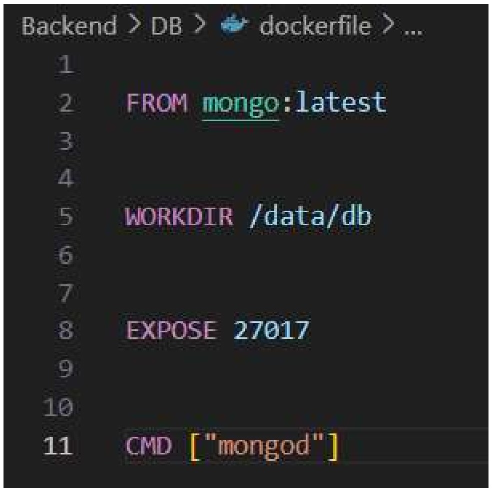
Backend Dockerfile:



```dockerfile
FROM node:lts-alpine
ENV NODE_ENV=production
WORKDIR /usr/src/app
COPY ["package.json", "package-lock.json*", "npm-shrinkwrap.json*", "./"]
RUN npm install --production --silent && mv node_modules ../
COPY . .
EXPOSE 3000
RUN chown -R node /usr/src/app
USER node
CMD ["npm", "start"]

```

- Uses **Node.js Alpine** for efficiency.
- Installs only production dependencies to optimize the build.
- Runs the backend on port **3000**.

Database Dockerfile:

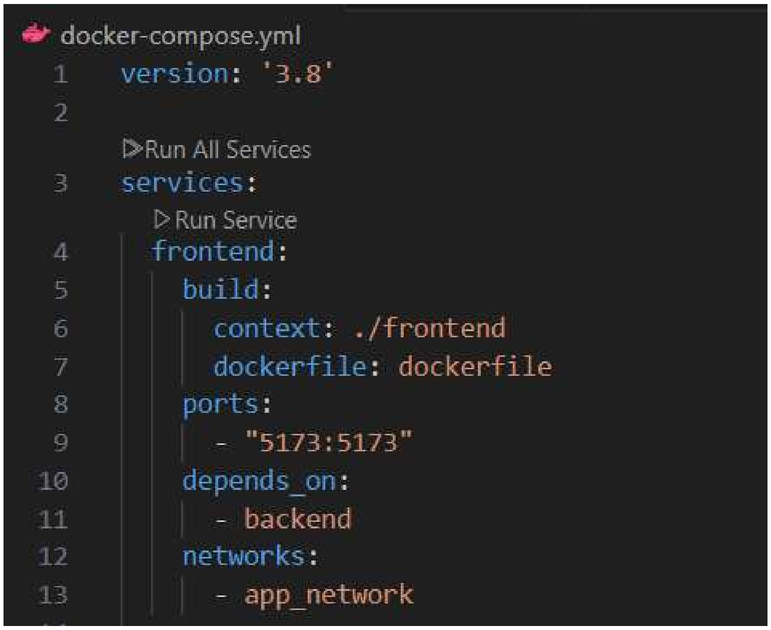


```dockerfile

FROM mongo:latest


WORKDIR /data/db


EXPOSE 27017


CMD ["mongod"]
```

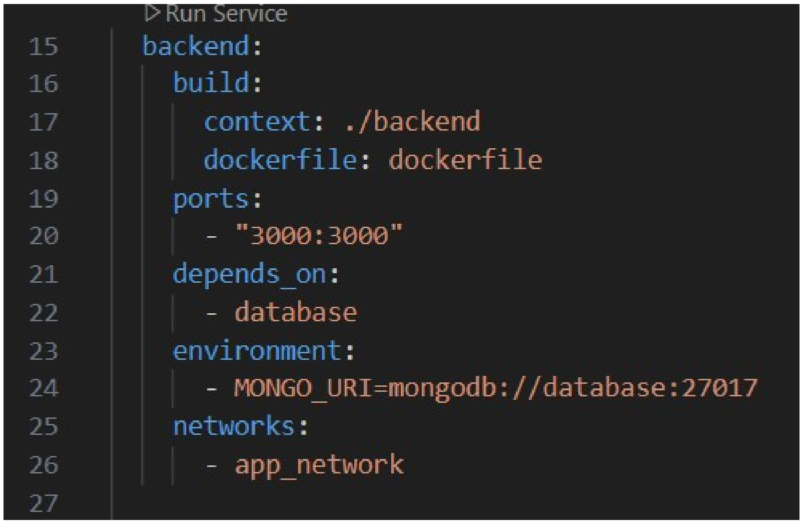
- Uses the **MongoDB** official image.
- Exposes **port 27017**, which is MongoDB's default port.
- Runs the **MongoDB daemon** (mongod).

3. Creating the Docker Compose file

The docker-compose.yml file defines how all three services (Frontend, Backend, and Database) work together.



```yaml
version: '3.8'

services:
  frontend:
    build:
      context: ./frontend
      dockerfile: dockerfile
    ports:
      - "5173:5173"
    depends_on:
      - backend
    networks:
      - app_network
```



```yaml
  backend:
    build:
      context: ./backend
      dockerfile: dockerfile
    ports:
      - "3000:3000"
    depends_on:
      - database
    environment:
      - MONGO_URI=mongodb://database:27017
    networks:
      - app_network

```

Defines **three services**: frontend, backend, and database.

Networks:

- All services are connected using a custom **bridge network** (app_network).

Volumes:

- The database uses a persistent volume (mongo_data) to store data.

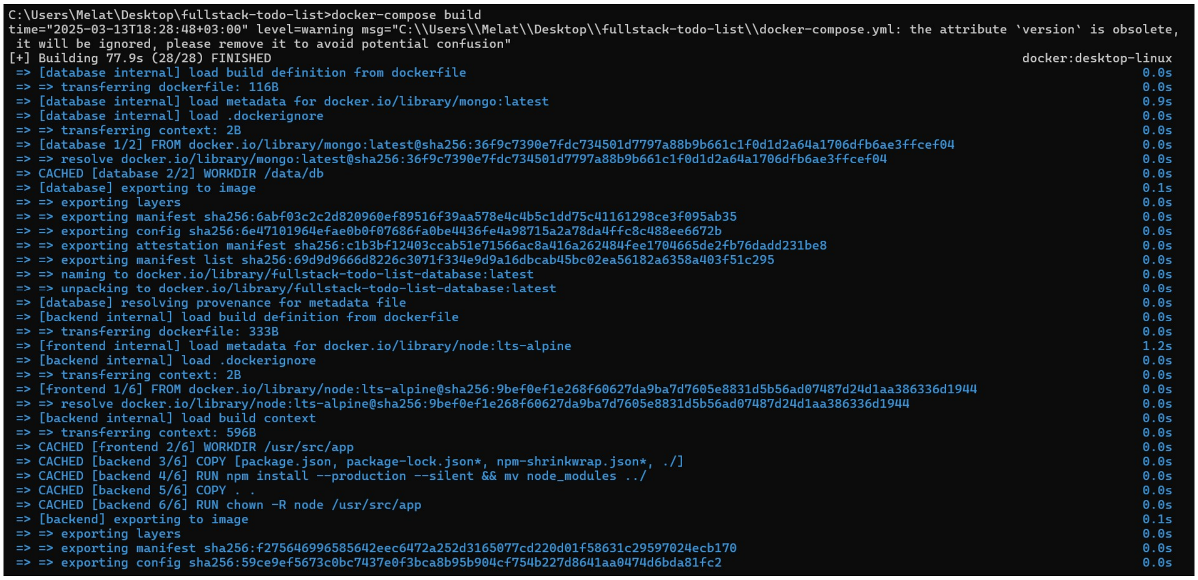
Dependencies:

- The **Frontend** depends on the **Backend**.
- The **Backend** depends on the **Database**.

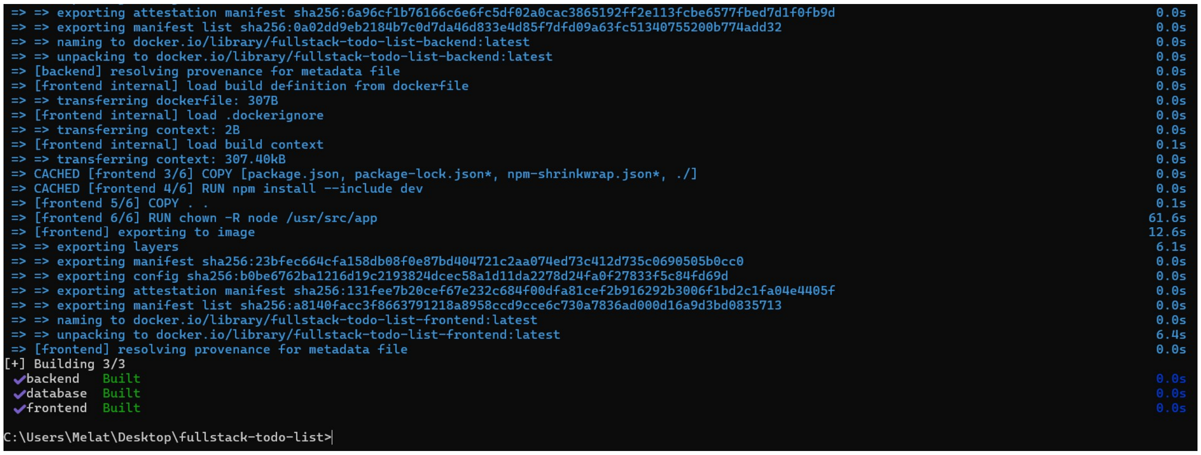
4. Containers

Build the Containers:

Use Docker Compose to build the necessary containers for the frontend, backend, and database:

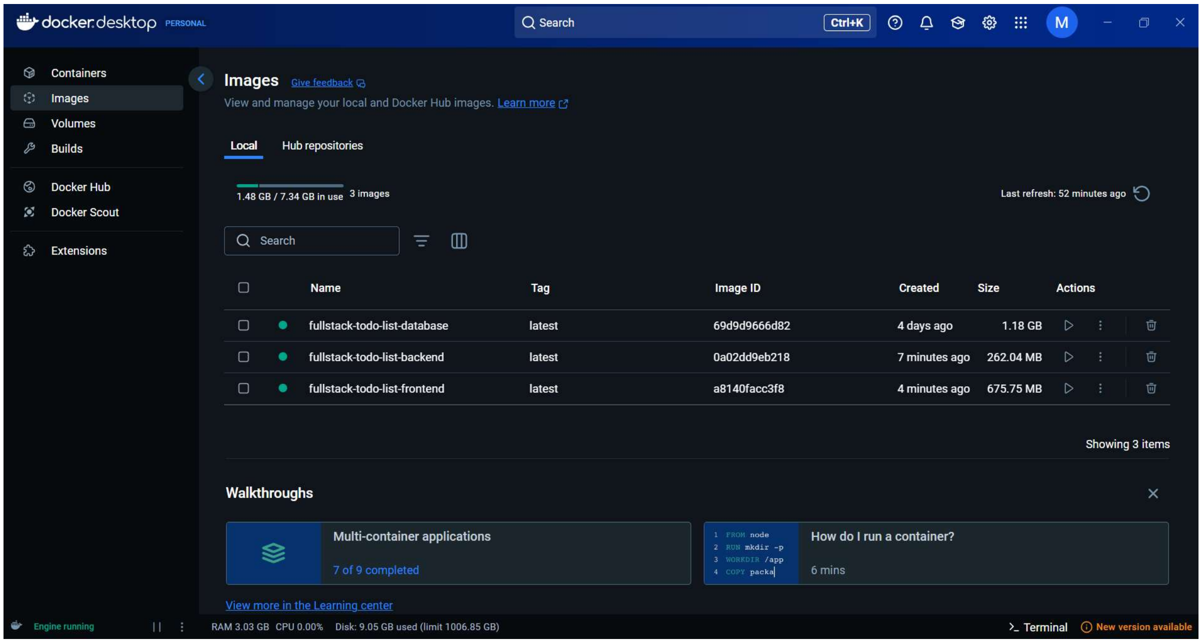


```
C:\Users\Melat\Desktop\fullstack-todo-list>docker-compose build
time="2025-03-13T18:28:48+03:00" level=warning msg="C:\\Users\\Melat\\Desktop\\fullstack-todo-list\\docker-compose.yml: the attribute `version` is obsolete,
 it will be ignored, please remove it to avoid potential confusion"
[+] Building 77.9s (28/28) FINISHED                                                                                  docker:desktop-linux
 => [database internal] load build definition from dockerfile                                                                        0.0s
 => => transferring dockerfile: 116B                                                                                                 0.0s
 => [database internal] load metadata for docker.io/library/mongo:latest                                                             0.9s
 => [database internal] load .dockerignore                                                                                           0.0s
 => => transferring context: 2B                                                                                                      0.0s
 => [database 1/2] FROM docker.io/library/mongo:latest@sha256:36f9c7390e7fdc734501d7797a88b9b661c1f0d1d2a64a1706dfb6ae3ffcef04      0.0s
 => => resolve docker.io/library/mongo:latest@sha256:36f9c7390e7fdc734501d7797a88b9b661c1f0d1d2a64a1706dfb6ae3ffcef04               0.0s
 => CACHED [database 2/2] WORKDIR /data/db                                                                                           0.0s
 => [database] exporting to image                                                                                                    0.1s
 => => exporting layers                                                                                                              0.0s
 => => exporting manifest sha256:6abf03c2c2d820960ef89516f39aa578e4c4b5c1dd75c41161298ce3f095ab35                                   0.0s
 => => exporting config sha256:6e47101964efae0b0f07686fa0be4436fe4a98715a2a78da4ffc8c488ee6672b                                     0.0s
 => => exporting attestation manifest sha256:c1b3bf12403ccab51e71566ac8a416a262484fee1704665de2fb76dadd231be8                       0.0s
 => => exporting manifest list sha256:69d9d9666d8226c3071f334e9d9a16dbcab45bc02ea56182a6358a403f51c295                              0.0s
 => => naming to docker.io/library/fullstack-todo-list-database:latest                                                              0.0s
 => => unpacking to docker.io/library/fullstack-todo-list-database:latest                                                           0.0s
 => [database] resolving provenance for metadata file                                                                               0.0s
 => [backend internal] load build definition from dockerfile                                                                        0.0s
 => => transferring dockerfile: 333B                                                                                                 0.0s
 => [frontend internal] load metadata for docker.io/library/node:lts-alpine                                                         1.2s
 => [backend internal] load .dockerignore                                                                                            0.0s
 => => transferring context: 2B                                                                                                      0.0s
 => [frontend 1/6] FROM docker.io/library/node:lts-alpine@sha256:9bef0ef1e268f60627da9ba7d7605e8831d5b56ad07487d24d1aa386336d1944   0.0s
 => => resolve docker.io/library/node:lts-alpine@sha256:9bef0ef1e268f60627da9ba7d7605e8831d5b56ad07487d24d1aa386336d1944            0.0s
 => [backend internal] load build context                                                                                            0.0s
 => => transferring context: 596B                                                                                                    0.0s
 => CACHED [frontend 2/6] WORKDIR /usr/src/app                                                                                       0.0s
 => CACHED [backend 3/6] COPY [package.json, package-lock.json*, npm-shrinkwrap.json*, ./]                                          0.0s
 => CACHED [backend 4/6] RUN npm install --production --silent && mv node_modules ../                                               0.0s
 => CACHED [backend 5/6] COPY . .                                                                                                    0.0s
 => CACHED [backend 6/6] RUN chown -R node /usr/src/app                                                                              0.0s
 => [backend] exporting to image                                                                                                     0.1s
 => => exporting layers                                                                                                              0.0s
 => => exporting manifest sha256:f275646996585642eec6472a252d3165077cd220d01f58631c29597024ecb170                                   0.0s
 => => exporting config sha256:59ce9ef5673c0bc7437e0f3bca8b95b904cf754b227d8641aa0474d6bda81fc2                                     0.0s
```



```
=> => exporting attestation manifest sha256:6a96cf1b76166c6e6fc5df02a0cac3865192ff2e113fcbe6577fbed7d1f0fb9d                         0.0s
=> => exporting manifest list sha256:0a02dd9eb2184b7c0d7da46d833e4d85f7dfd09a63fc51340755200b774add32                                0.0s
=> => naming to docker.io/library/fullstack-todo-list-backend:latest                                                                 0.0s
=> => unpacking to docker.io/library/fullstack-todo-list-backend:latest                                                              0.0s
=> [backend] resolving provenance for metadata file                                                                                  0.0s
=> [frontend internal] load build definition from dockerfile                                                                         0.0s
=> => transferring dockerfile: 307B                                                                                                  0.0s
=> [frontend internal] load .dockerignore                                                                                            0.0s
=> => transferring context: 2B                                                                                                       0.0s
=> [frontend internal] load build context                                                                                            0.1s
=> => transferring context: 307.40kB                                                                                                 0.0s
=> CACHED [frontend 3/6] COPY [package.json, package-lock.json*, npm-shrinkwrap.json*, ./]                                           0.0s
=> CACHED [frontend 4/6] RUN npm install --include dev                                                                               0.0s
=> [frontend 5/6] COPY . .                                                                                                           0.1s
=> [frontend 6/6] RUN chown -R node /usr/src/app                                                                                    61.6s
=> [frontend] exporting to image                                                                                                    12.6s
=> => exporting layers                                                                                                               6.1s
=> => exporting manifest sha256:23bfec664cfa158db08f0e87bd404721c2aa074ed73c412d735c0690505b0cc0                                     0.0s
=> => exporting config sha256:b0be6762ba1216d19c2193824dcec58a1d11da2278d24fa0f27833f5c84fd69d                                       0.0s
=> => exporting attestation manifest sha256:131fee7b20cef67e232c684f00dfa81cef2b916292b3006f1bd2c1fa04e4405f                         0.0s
=> => exporting manifest list sha256:a8140facc3f8663791218a8958ccd9cce6c730a7836ad000d16a9d3bd0835713                                0.0s
=> => naming to docker.io/library/fullstack-todo-list-frontend:latest                                                                0.0s
=> => unpacking to docker.io/library/fullstack-todo-list-frontend:latest                                                             6.4s
=> [frontend] resolving provenance for metadata file                                                                                 0.0s
[+] Building 3/3
 ✔backend   Built                                                                                                                    0.0s
 ✔database  Built                                                                                                                    0.0s
 ✔frontend  Built                                                                                                                    0.0s

C:\Users\Melat\Desktop\fullstack-todo-list>
```

We can see our images on the Docker Desktop Software



Images Give feedback

View and manage your local and Docker Hub images. Learn more

1.48 GB / 7.34 GB in use 3 images

Last refresh: 52 minutes ago

□		Name	Tag	Image ID	Created	Size	Actions
□	●	fullstack-todo-list-database	latest	69d9d9666d82	4 days ago	1.18 GB	
□	●	fullstack-todo-list-backend	latest	0a02dd9eb218	7 minutes ago	262.04 MB	
□	●	fullstack-todo-list-frontend	latest	a8140facc3f8	4 minutes ago	675.75 MB	

Showing 3 items

Walkthroughs

Multi-container applications

7 of 9 completed

```
1 FROM node
2 RUN mkdir -p
3 WORKDIR /app
4 COPY packa
```

How do I run a container?

6 mins

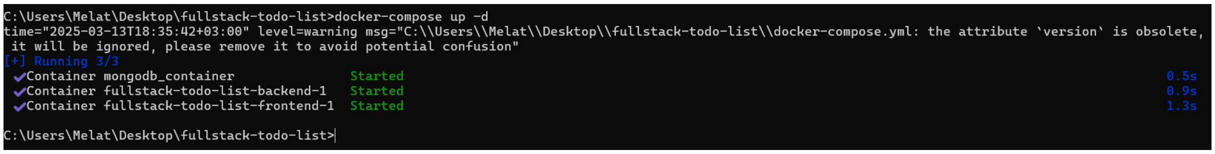
View more in the Learning center

Engine running

RAM 3.03 GB CPU 0.00% Disk: 9.05 GB used (limit 1006.85 GB)

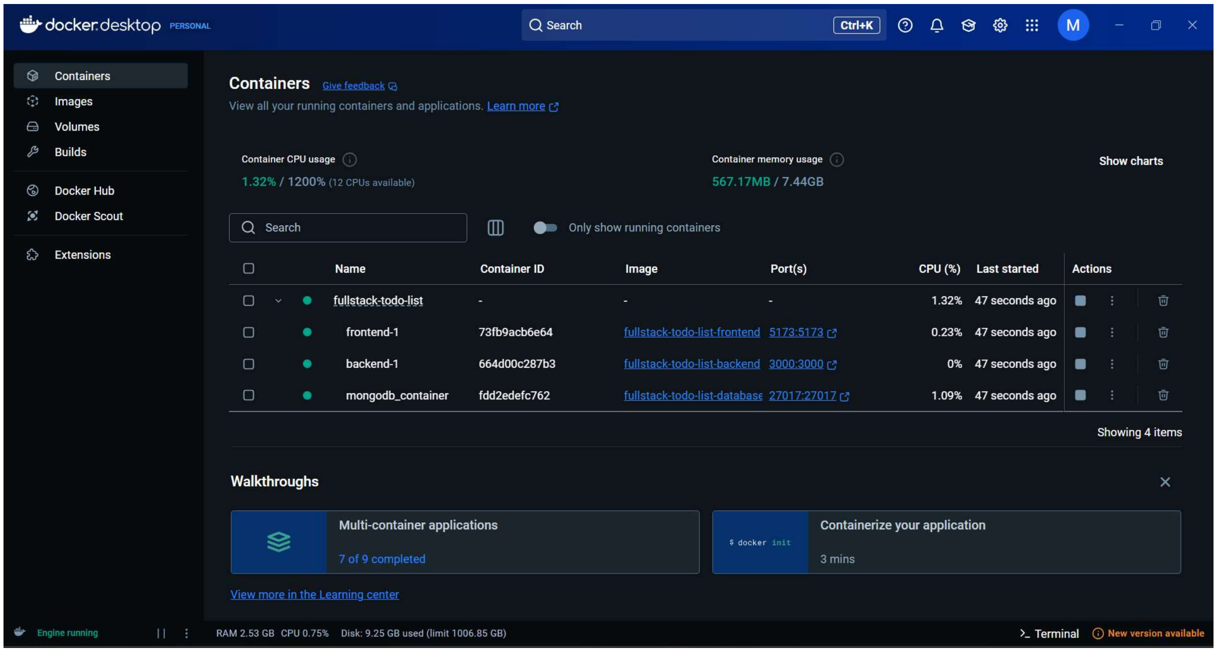
Start the Containers:

Run the containers



```
C:\Users\Melat\Desktop\fullstack-todo-list>docker-compose up -d
time="2025-03-13T18:35:42+03:00" level=warning msg="C:\\Users\\Melat\\Desktop\\fullstack-todo-list\\docker-compose.yml: the attribute `version` is obsolete,
 it will be ignored, please remove it to avoid potential confusion"
[+] Running 3/3
 ✔Container mongodb_container              Started                                                                                          0.5s
 ✔Container fullstack-todo-list-backend-1  Started                                                                                          0.9s
 ✔Container fullstack-todo-list-frontend-1 Started                                                                                          1.3s

C:\Users\Melat\Desktop\fullstack-todo-list>
```

And we can see our containers on The Desktop Docker Application



Containers

View all your running containers and applications. Learn more

Container CPU usage 1.32% / 1200% (12 CPUs available)

Container memory usage 567.17MB / 7.44GB

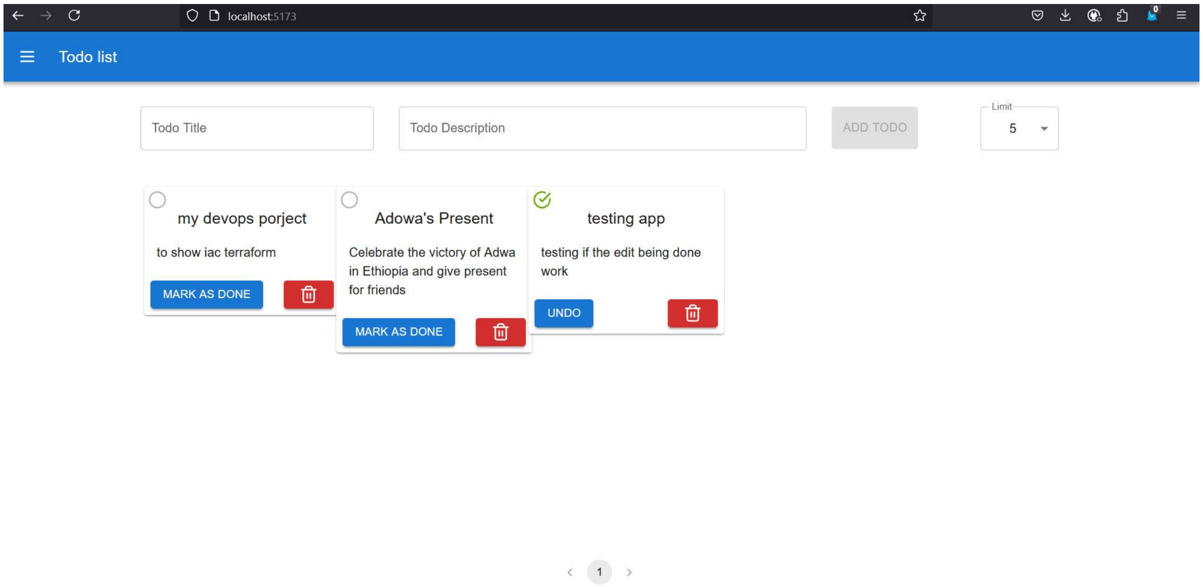
Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
fullstack-todo-list	-	-	-	1.32%	47 seconds ago	
frontend-1	73fb9acb6e64	fullstack-todo-list-frontend	5173:5173	0.23%	47 seconds ago	
backend-1	664d00c287b3	fullstack-todo-list-backend	3000:3000	0%	47 seconds ago	
mongodb_container	fdd2edefc762	fullstack-todo-list-database	27017:27017	1.09%	47 seconds ago	

Showing 4 items

RAM 2.53 GB CPU 0.75% Disk: 9.25 GB used (limit 1006.85 GB)

Access the Application:

Frontend: Open your browser and navigate to http://localhost:5173.



my devops porject

to show iac terraform

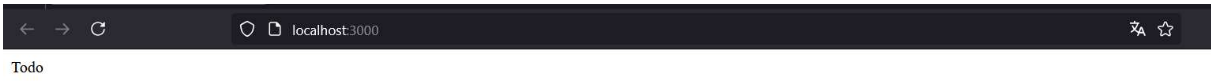
Adowa's Present

Celebrate the victory of Adwa in Ethiopia and give present for friends

testing app

testing if the edit being done work

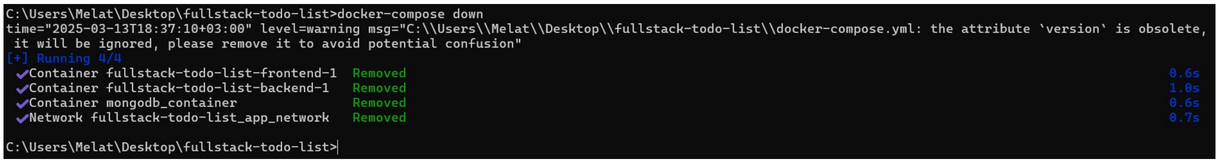
Backend: You can access the backend API through http://localhost:3000.



Todo

Stop the Containers:

To stop the containers, run:



```
C:\Users\Melat\Desktop\fullstack-todo-list>docker-compose down
time="2025-03-13T18:37:10+03:00" level=warning msg="C:\\Users\\Melat\\Desktop\\fullstack-todo-list\\docker-compose.yml: the attribute `version` is obsolete,
 it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
 ✔Container fullstack-todo-list-frontend-1  Removed                                                                                    0.6s
 ✔Container fullstack-todo-list-backend-1   Removed                                                                                    1.0s
 ✔Container mongodb_container               Removed                                                                                    0.6s
 ✔Network fullstack-todo-list_app_network   Removed                                                                                    0.7s

C:\Users\Melat\Desktop\fullstack-todo-list>
```