

Deploy MERN Stack Application using Docker Compose

In this we containerize a MERN Stack application using Docker and then later deploy it with Docker Compose. We then build each environment using docker and run each application in separate containers, and finally deploy the entire stack using a single YAML file with Docker Compose.

Before knowing what is a MERN Stack application is, let's understand what is Three-Tier-Architecture.

In simple terms, Three-Tier-Application means Three Main Layers.

1. Presentation Layer: This is the UI or Frontend, where the users will actually interact with the application.
2. Business logic: This is the Backend layer, where all the data processing happens.
3. Data Layer: This is the Database layer or storage where all the data is stored and fetched from. This is typically how a Three-Tier-Architecture looks and functions.

MERN Stands for

M: MongoDB (Data Layer)

E: Express.js (Business logic layer)

R: React.js (Presentation Layer)

N: Node.js (Backend or Server)

Together, these technologies can create a scalable application.

Pre-requisites:

- Fork/clone or download the code to your local
- Docker

Steps:

1. Clone the project to your local

```
git clone https://github.com/bhargavdevopsaws/MERN-Docker-Compose-.git
```

```
bharg@DESKTOP-JVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose (master)
$ git clone https://github.com/bhargavdevopsaws/MERN-Docker-Compose.git
Cloning into 'MERN-Docker-Compose'...
remote: Enumerating objects: 55, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 55 (delta 2), reused 55 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (55/55), 86.83 KiB | 290.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.
```

2. In this application, we will create Dockerfile for frontend and backend, and database we will deploy directly using Mongo image.
3. So, to create a Dockerfile for the MERN application for frontend copy the below code in to the dockerfile.

```
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose (master)
$ cd MERN-Docker-Compose

bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose (main)
$ cd mern/frontend

bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/frontend (main)
$ nano Dockerfile
```

- Cd MERN-Docker-Compose
- Cd mern/frontend
- nano Dockerfile

#Using the Official Base image

FROM node:18.9.1 as build

#Create a Work directory

WORKDIR /app

#Copy package.json files from host to the container

COPY package*.json .

#Install the dependencies

RUN npm install

#Copy rest of the files inside the container

COPY . .

#List the port in which the application is running

EXPOSE 5173

#Run the application

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

Save and Exit:

1. Press Ctrl+O to save the file.
2. Press Enter to confirm the file name.
3. Press Ctrl+X to exit the editor.

4. After creating a Dockerfile, let's build this dockerfile and create an image.

- `docker build -t mern-frontend .`

```
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/frontend (main)
$ docker build -t mern-frontend .
#0 building with "desktop-linux" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 689B 0.0s done
#1 DONE 0.1s

#2 [internal] load metadata for docker.io/library/node:18.9.1
#2 DONE 1.5s

#3 [internal] load .dockerignore
#3 transferring context: 2B 0.0s done
#3 DONE 0.1s

#4 [1/5] FROM docker.io/library/node:18.9.1@sha256:d6ed353d022f6313aa7c3f3df69f3a216f1c9f8c3374502eb5e6c45088ce68e8
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 1.17kB 0.0s done
#5 DONE 0.0s

#6 [2/5] WORKDIR /app
#6 CACHED

#7 [3/5] COPY package.json .
#7 CACHED

#8 [4/5] RUN npm install
#8 CACHED

#9 [5/5] COPY . .
#9 CACHED

#10 exporting to image
#10 exporting layers done
#10 writing image sha256:cb1ab2e5f46093f00161f30ef0591d4f3d9882d770fe42404c45f480c04e8b46 done
#10 naming to docker.io/library/mern-frontend 0.0s done
#10 DONE 0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/u2ub0k2chs3xi7l366se8bjtg
```

5. Now run the frontend container to check if it is running.

- `docker run -d -p 5173:5173 --name=frontend --network=mern mern-frontend`

```
MINGW64:/c:/Users/bharg/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/frontend (main)
$ docker run -d -p 5173:5173 --name=frontend --network=mern mern-frontend
4f70299ce15cd27e7ee04b35f9aecc48028035f71442c71cd51951c771e462d4
```

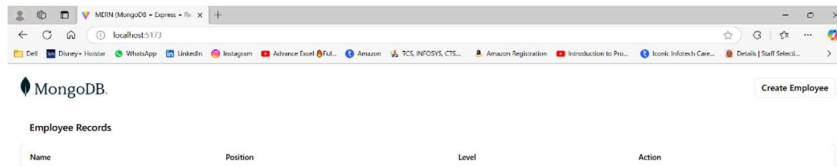
6. We will check if the container is running.

- `docker ps`

```
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/frontend (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
4f70299ce15c   mern-frontend  "/docker-entrypoint.s..." 15 seconds ago Up 14 seconds 0.0.0.0:5173->5173/tcp   frontend
1096e0745d1    nginx         "/docker-entrypoint.s..." 7 minutes ago  Up 7 minutes  8080/tcp                 k8s_my-app-my-app-7bdf9f8dff-zwksm_default_39c762a7-565f-4708-8dad-e4ea2ec3246_3
d01f122a0f979  nginx         "/docker-entrypoint.s..." 7 minutes ago  Up 7 minutes  8080/tcp                 k8s_my-app-my-app-7bdf9f8dff-qtk1t_default_c6822765-8873-4403-b544-245dfc960eb_3
864af0589fec    nginx         "/docker-entrypoint.s..." 7 minutes ago  Up 7 minutes  8080/tcp                 k8s_my-app-my-app-7bdf9f8dff-pcdeb_default_b9f45845-b40d-49cb-8a09-9a4690d24f5f_3
b9a739e8a8a9    649df1af0b6e  "/usr/sbin/nginx -g ..." 7 minutes ago  Up 7 minutes  8080/tcp                 k8s_blog-container_blog-app-678866dcf9-rzmwz_ingress-testing_e4b30467-6bfa-466e-85b6-3abb6a9cd9b_14
64386b1f2734    d23615dc2f51  "/usr/sbin/nginx -g ..." 7 minutes ago  Up 7 minutes  8080/tcp                 k8s_stream-container_stream-app-6fd5c45d5d-5qd52_ingress-testing_63f8f6db-89fd-4ed3-9830-a14b27f3b9ee_4
9782b15f02cd    5a3c47128078  "/usr/bin/dumb-init ..." 7 minutes ago  Up 7 minutes  8080/tcp                 k8s_controller_infra-nginx-ingress-trial-ingress-nginx-controller-8575475ak7mt_default_f05f2204-d0b8-4692-b3b5-6a3236c2c1a2_4
06408f240a9     ffcc66479b5b  "/usr/bin/dumb-init ..." 7 minutes ago  Up 7 minutes  8080/tcp                 k8s_controller_ingress-nginx-controller-7dcdbcf84-4j52q_ingress-nginx_aad1397f-3adc-4acf-8ba5-e011f5218d76_4
```

7. Open your browser and run it using localhost

- <http://localhost:5173/>



8. Frontend is running. So, let's create & deploy backend as well in the next step.

9. Before deploying backend, let's run the mongodb.

10. We are running mongodb directly by using mongo latest image, so run the below docker command to start the mongo container.

- `docker run -d -p 27017:27017 --name=mongodb --network=mern -v mongo-data:/data/db mongo:latest`

```
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/frontend (main)
$ docker run -d -p 27017:27017 --name=mongodb --network=mern -v mongo-data:/data/db mongo:latest
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
de44b265507a: Pulling fs layer
4d73348b9ac0: Pulling fs layer
6f1309d23164: Pulling fs layer
e4bde7d374c0: Pulling fs layer
9a0eb01246c7: Pulling fs layer
d2b3dabec753: Pulling fs layer
8606fc5459f6: Pulling fs layer
8f50be4b980b: Pulling fs layer
9a0eb01246c7: Waiting
d2b3dabec753: Waiting
8606fc5459f6: Waiting
8f50be4b980b: Waiting
e4bde7d374c0: Waiting
4d73348b9ac0: Verifying Checksum
4d73348b9ac0: Download complete
6f1309d23164: Verifying Checksum
6f1309d23164: Download complete
9a0eb01246c7: Verifying Checksum
9a0eb01246c7: Download complete
d2b3dabec753: Verifying Checksum
d2b3dabec753: Download complete
e4bde7d374c0: Download complete
8f50be4b980b: Download complete
de44b265507a: Download complete
de44b265507a: Pull complete
4d73348b9ac0: Pull complete
6f1309d23164: Pull complete
e4bde7d374c0: Pull complete
9a0eb01246c7: Pull complete
d2b3dabec753: Pull complete
8606fc5459f6: Verifying Checksum
8606fc5459f6: Download complete
8606fc5459f6: Pull complete
8f50be4b980b: Pull complete
Digest: sha256:8565ecda5b221016d70f7745ac1ba0b97ccb05836157f8a343e987338fdc8350
Status: Downloaded newer image for mongo:latest
e577a309dbd497cbf09af5780abd3635264892e3ecd71c1126063a0db15b0930
```

11. Check if the mongodb container is running.

```
mingw64@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/backend (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
f605ec16d426   mern-backend   "docker-entrypoint.s..." 33 seconds ago Up 30 seconds 0.0.0.0:5050->5050/tcp             backend
e577a309dbd4   mongo:latest   "docker-entrypoint.s..." 53 minutes ago Up 53 minutes 0.0.0.0:27017->27017/tcp           mongod
1096e00745d1    nginx          "/docker-entrypoint..." About an hour ago Up About an hour 0.0.0.0:5173->5173/tcp             frontend
4017122afb79    nginx          "/docker-entrypoint..." About an hour ago Up About an hour                               k8s_my-app_my-app-7bdf9f8dff-zkxsm_default_39c762a7-565f-4708-8da4-e4ea27ec3246_3
964ef0589fec    nginx          "/docker-entrypoint..." About an hour ago Up About an hour                               k8s_my-app_my-app-7bdf9f8dff-qtklr_default_d6421765-b873-4403-b544-245dfc960ecb_3
b9a739e8a8a9    nginx          "/usr/sbin/nginx -g ..." About an hour ago Up About an hour                               k8s_my-app_my-app-7bdf9f8dff-pcndb_default_c9f45845-b40d-49cb-8a09-9a0d90d245f3_3
b6a8cd9b1_4     d23615dc2f51   "/usr/sbin/nginx -g ..." About an hour ago Up About an hour                               k8s_blog-container_blog-app-678866dcf9-rzowc_ingress-testing_e4b30467-6bfa-466e-85b6-3ab
94388b1f2724     5a3c47128078   "/usr/bin/dumb-init ..." About an hour ago Up About an hour                               k8s_stream-container_stream-app-6fd5c45d5d-5qd52_ingress-testing_63f8f6db-89fd-4ed3-9830
9792b15f02cd     ffc66479b5b    "/usr/bin/dumb-init ..." About an hour ago Up About an hour                               k8s_controller_infra-nginx-ingress-trial-ingress-nginx-controller-85f54752k7mt_default_f
05f2204-9088-4692-b3b5-aa3226cc2c12_4
064087f240a9     f-8ba5-ed11f5218d76_4
```

12. create a Dockerfile for the backend.

```
MINGW64:/c:/Users/bharg/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern (main)
$ cd ..

bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose (main)
$ cd mern/backend

bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/backend (main)
$ nano Dockerfile
```

#Using the Official Base image

FROM node:18.9.1

#Create a Work directory

WORKDIR /app

#Copy package.json files from host to the container

COPY package.json .

#Install dependencies

RUN npm install

#Copy rest of the files inside the container

COPY . .

#List the port in which the application is running

EXPOSE 5050

#Run the application

CMD ["npm", "start"]

14. Once the Dockerfile is created, build the image

- `docker build -t mern-backend .`

```
MINGW64/c/Users/bharg/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/backend (main)
$ docker build -t mern-backend .
#0 building with "desktop-linux" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 374B 0.0s done
#1 DONE 0.1s

#2 [internal] load metadata for docker.io/library/node:18.9.1
#2 ...

#3 [auth] library/node:pull token for registry-1.docker.io
#3 DONE 0.0s

#2 [internal] load metadata for docker.io/library/node:18.9.1
#2 DONE 5.8s

#4 [internal] load .dockerignore
#4 transferring context: 2B done
#4 DONE 0.0s

#5 [1/5] FROM docker.io/library/node:18.9.1@sha256:d6ed353d022f6313aa7c3f3df69f3a216f1c9f8c3374502eb5e6c45088ce68e8
#5 DONE 0.0s

#6 [internal] load build context
#6 transferring context: 36.19kB 0.1s done
#6 DONE 0.1s

#7 [2/5] WORKDIR /app
#7 CACHED

#8 [3/5] COPY package*.json ./
#8 DONE 0.1s

#9 [4/5] RUN npm install
#9 8.225
#9 8.225 added 77 packages, and audited 78 packages in 6s
#9 8.225
#9 8.226 11 packages are looking for funding
#9 8.226 run 'npm fund' for details
#9 8.241
#9 8.241 6 vulnerabilities (3 low, 3 high)
#9 8.241
#9 8.241 To address all issues, run:
#9 8.241 npm audit fix
#9 8.241
#9 8.241 Run 'npm audit' for details.
#9 8.245 npm notice
#9 8.251 npm notice New major version of npm available! 8.19.1 -> 10.9.2
#9 8.251 npm notice Changelog: <https://github.com/npm/cli/releases/tag/v10.9.2>
#9 8.251 npm notice Run 'npm install -g npm@10.9.2' to update!
#9 8.251 npm notice
#9 DONE 8.4s

#10 [5/5] COPY . .
#10 DONE 0.1s

#11 exporting to image
#11 exporting layers
#11 exporting layers 0.3s done
#11 writing image sha256:158382f58bab3cdeb88f47f7be39b1a2fb4553a532a4fca38210ea214992e8d4e done
```

14. Run the backend container using the below code.

- `docker run -d -p 5050:5050 --name=backend --network=mern mern-backend`

```
MINGW64/c/Users/bharg/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/backend (main)
$ docker run -d -p 5050:5050 --name=backend --network=mern mern-backend
fe05ec164e26560ad37988934eaea4b1b0aee496a8625603cf5eacfc1c71c782
```

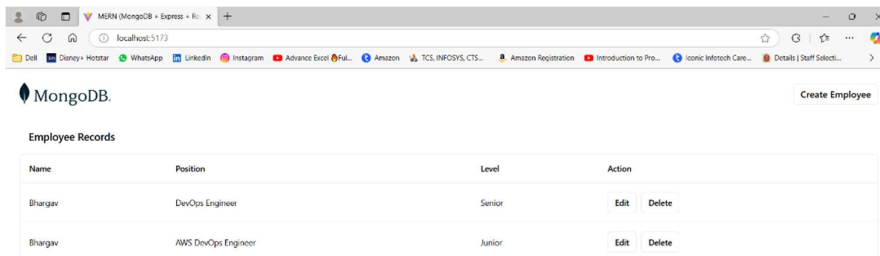
15. Check if the backend container is running.

```
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/backend (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
fe05ec164e26   mern-backend   "docker-entrypoint.s..." 33 seconds ago Up 30 seconds 0.0.0.0:5050->5050/tcp             backend
e277a309b0d4   mongo:latest   "docker-entrypoint.s..." 53 minutes ago Up 53 minutes 0.0.0.0:27017->27017/tcp             mongo
4f7099cc15c    mern-frontend   "docker-entrypoint.s..." 57 minutes ago Up 56 minutes 0.0.0.0:5173->5173/tcp             frontend
1096e0745d1    nginx          "/docker-entrypoint..." About an hour ago Up About an hour                    k8s_my-app_my-app-7bdf9f8dff-zwksm_default_39c762a7-565f-4708-8da4-e4ea2ec3246_3
401f122a7b79   nginx          "/docker-entrypoint..." About an hour ago Up About an hour                    k8s_my-app_my-app-7bdf9f8dff-qtktir_default_d8421765-b873-4403-b544-2450fc90e0cb_3
964af0589fec   nginx          "/docker-entrypoint..." About an hour ago Up About an hour                    k8s_my-app_my-app-7bdf9f8dff-ncdb_default_b9f43845-b402-4903-84d9-3ad690d24f5f_3
9ba739e8a8a9   649d1af0b6e    "/usr/sbin/nginx -g ..." About an hour ago Up About an hour                    k8s_blog-container_blog-app-67886dcf9-rzmwz_ingress-testing_e4b30467-6bfa-466e-85b6-3ab
d6eacdb14      64386b1f2734   d23615dc2f51          "/usr/sbin/nginx -g ..." About an hour ago Up About an hour                    k8s_stream-container_stream-app-6fd5c45dd-3qd52_ingress-testing_63f8f6db-89fd-4ed3-9830
e14b2773b9ea4   9792015f02cd   3a3c47128078          "/usr/bin/dumb-init ..." About an hour ago Up About an hour                    k8s_controller_infra-nginx-ingress-trial-ingress-nginx-controller-85754752k7et_default_f
05f2204-d0b8-4692-bb55-6a3236c2c1a2_4
064087f240a9   ffcc66479db5   "/usr/bin/dumb-init ..." About an hour ago Up About an hour                    k8s_controller_ingress-nginx-controller-7dcdbcf84-4j32q_ingress-nginx_aad1397f-3adc-4ac
f-bba5-ed11f5218d76_4
```

16. Now all the containers are running, check if our application is running fine, and are we able to save the employee record and edit is as well.

17. Go to the frontend and try to save an employee record.

- <http://localhost:5173/>



18. Instead of running multiple commands and executing numerous steps, we can deploy everything with a single command. We will use a YAML file, specifically the Docker Compose file.

19. Now stop all the containers and delete all the containers.

- `docker rm -f fe05ec164e26 e577a309dbd4 4f70299ce15c`

```
bharg@DESKTOP-JVVFHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/backend (main)
$ docker rm -f fe05ec164e26 e577a309dbd4 4f70299ce15c
fe05ec164e26
e577a309dbd4
4f70299ce15c
```

20. Check if any container is running.

- `docker ps`

21. create a file called docker compose. This is the file with which we will run all the containers using a single command.

- `nano docker-compose.yml`

Enter the YAML Script:

services:

backend:

build: ./mern/backend


```
ports:
- "5050:5050"
networks:
- mern
depends_on:
- mongodb
frontend:
build: ./mern/frontend
ports:
- "5173:5173"
networks:
- mern
mongodb:
image: mongo:latest
```

Deploy MERN Stack Application using Docker Compose 9

```
ports:
- "27017:27017"
networks:
- mern
volumes:
- mongo-data:/data/db
networks:
mern_network:
driver: bridge
volumes:
mongo-data:
driver: local
```

* Press CTRL+O, then press Enter to save the file. Next, press CTRL+X to exit the nano editor.

Save this file, and run the below commands.

- `Cd ..`
- `nano docker-compose.yml`

```
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern/backend (main)
$ cd ..

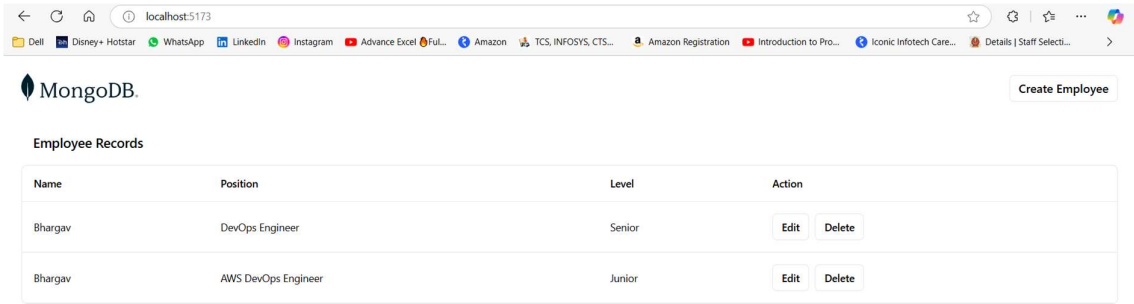
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern (main)
$ nano docker-compose.yml
```

- `docker-compose up -d`

```
bharg@DESKTOP-JVVVHFL MINGW64 ~/OneDrive/Desktop/Docker Compose/MERN-Docker-Compose/mern (main)
$ docker-compose up -d
time="2024-12-06T21:19:29+05:30" level=warning msg="C:\\Users\\bharg\\OneDrive\\Desktop\\Docker Compose\\MERN-Docker-Compose\\mern\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
Network mern_mern Creating
Volume "mern_mongo-data" Creating
Volume "mern_mongo-data" Created
Container mern-mongodb-1 Creating
Container mern-mongodb-1 Created
Container mern-frontend-1 Creating
Container mern-frontend-1 Created
Container mern-mongodb-1 Created
Container mern-backend-1 Creating
Container mern-backend-1 Created
Container mern-frontend-1 Starting
Container mern-mongodb-1 Starting
Container mern-frontend-1 Started
Container mern-mongodb-1 Started
Container mern-backend-1 Starting
Container mern-backend-1 Started
```

check if the site is working.

- <http://localhost:5173>



Conclusion:

Dockerizing a MERN Stack application from setting up individual containers, creating network and managing all the containers in a separate isolated network. And in the later stage, we are able to manage everything with Docker Compose file.