

1. Cloning the [express.js](https://github.com/roy35-909/Module-3-deployment.git) project repo

```
zamshed@Zamshed-Dell: ~/Module-3-deployment$ git clone https://github.com/roy35-909/Module-3-deployment.git
Cloning into 'Module-3-deployment'...
remote: Enumerating objects: 87, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 87 (delta 12), reused 4 (delta 4), pack-reused 74 (from 1)
Receiving objects: 100% (87/87), 30.08 KiB | 427.00 KiB/s, done.
Resolving deltas: 100% (28/28), done.
zamshed@Zamshed-Dell:~/Module-3-deployment$ cd Module-3-deployment
zamshed@Zamshed-Dell:~/Module-3-deployment$ cd Module-3-deployment
^C
zamshed@Zamshed-Dell:~/Module-3-deployment$ nano Dockerfile
```

```
zamshed@Zamshed-Dell:~/Module-3-deployment$ ls
Dockerfile      eslint.config.js  package-lock.json  src              tsconfig.app.json  vite.config.ts
README.md       index.html        package.json        tailwind.config.js  tsconfig.json
docker-compose.yml  nginx.conf        postcss.config.js  test             tsconfig.node.json
zamshed@Zamshed-Dell:~/Module-3-deployment$
```

2.Creating & Editing Dockerfile

```
zamshed@Zamshed-Dell: ~/Module-3-deployment$ nano Dockerfile
```

```
GNU nano 6.2 Dockerfile
FROM node:22

# Set working directory inside container
WORKDIR /usr/src/app

# Copy package.json and package-lock.json
COPY package*.json ./

# Install dependencies
RUN npm install

# Install pm2 globally to run the app
RUN npm install pm2 -g

# Copy the rest of the app source code
COPY . .

# Expose port 3000 (the app's port)
EXPOSE 3000

# Start the app using pm2-runtime (production-friendly)
CMD ["pm2-runtime", "src/server.js", "--name", "node-app"]
```

3. Logging in Dockerhub

```
zamshed@Zamshed-Dell:~/Module-3-deployment$ docker login

USING WEB-BASED LOGIN

Info → To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: KSVB-ZXNP
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

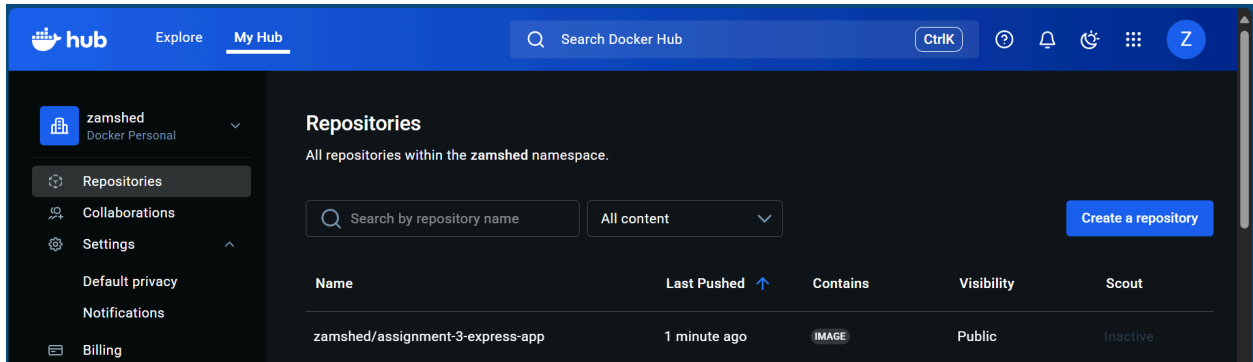
Waiting for authentication in the browser...

WARNING! Your credentials are stored unencrypted in '/home/zamshed/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

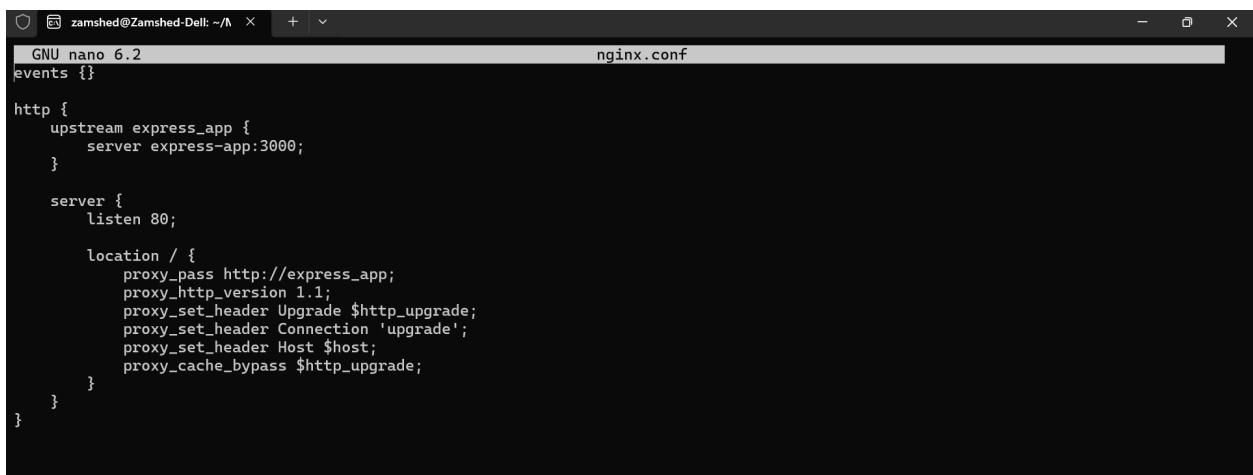
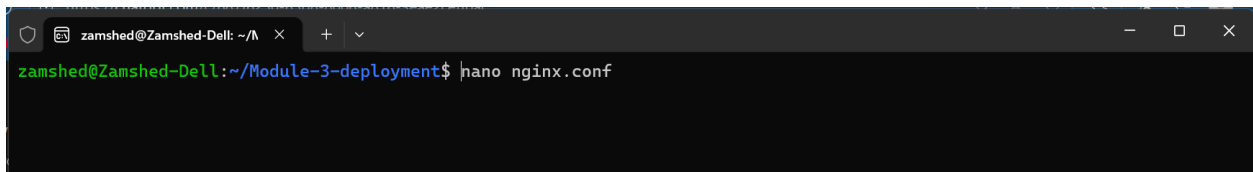
Login Succeeded
```

4. Building , then pushing image in Dockerhub

```
zamshed@Zamshed-Dell:~/Module-3-deployment$ docker build -t zamshed/assignment-3-express-app:latest .
[+] Building 2.4s (12/12) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 496B
=> [internal] load metadata for docker.io/library/node:22
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/6] FROM docker.io/library/node:22@sha256:079b6a683dc47a87673a6159c9e9b22b0687d04533087cf144c96fac8c26ecd3
=> [internal] load build context
=> => transferring context: 3.31kB
=> CACHED [2/6] WORKDIR /usr/src/app
=> CACHED [3/6] COPY package*.json ./
=> CACHED [4/6] RUN npm install
=> CACHED [5/6] RUN npm install pm2 -g
=> [6/6] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:6c29e9699bb7249181b92360b226bae55695125b74a4c2c1cb5be8b69c6af873
=> => naming to docker.io/zamshed/assignment-3-express-app:latest
zamshed@Zamshed-Dell:~/Module-3-deployment$ docker push zamshed/assignment-3-express-app:latest
The push refers to repository [docker.io/zamshed/assignment-3-express-app]
859941d3fe22: Pushed
c1874fd4b311: Pushed
0ffbc96b4407: Pushed
c8d866e75318: Pushed
fce229a78ea4: Layer already exists
e5b77f05d0e5: Layer already exists
```



5. Editing & Creating nginx.conf file



6. Editing & Creating docker-compose.yaml file

```
zamshed@Zamshed-Dell: ~/Module-3-deployment$ nano docker-compose.yml
```

```
GNU nano 6.2 docker-compose.yml
version: "3.9"

services:
  express-app:
    image: zamshed/assignment-3-express-app:latest
    container_name: express-app
    networks:
      - app-network
    ports:
      - "3000:3000" # expose port 3000 for direct access (optional)

  nginx:
    image: nginx:latest
    container_name: nginx
    depends_on:
      - express-app
    ports:
      - "8080:80" # access nginx on localhost:8080
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf:ro
    networks:
      - app-network

networks:
  app-network:
    driver: bridge
```

7. Crating container using the compose file

```
zamshed@Zamshed-Dell: ~/Module-3-deployment$ docker compose up -d
WARN[0000] /home/zamshed/Module-3-deployment/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 8/8
  ✓ nginx Pulled 53.0s
  ✓ 59e22667830b Pull complete 39.4s
  ✓ 140da4f89dcb Pull complete 47.7s
  ✓ 96e47e70491e Pull complete 47.8s
  ✓ 2ef442a3816e Pull complete 47.8s
  ✓ 4b1e45a9989f Pull complete 47.9s
  ✓ 1d9f51194194 Pull complete 47.9s
  ✓ f30ffbee4c54 Pull complete 48.0s
[+] Running 3/3
  ✓ Network module-3-deployment_app-network Created 0.1s
  ✓ Container express-app Started 0.7s
  ✓ Container nginx Started 0.8s
zamshed@Zamshed-Dell:~/Module-3-deployment$ docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
371fe4e99e54	nginx:latest		"/docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	0.0.0.0:8080->80/tcp
7b66c206f70a	zamshed/assignment-3-express-app:latest	express-app	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	0.0.0.0:3000->3000/tcp

8. Final Output

URL	What to expect
http://localhost:3000/	Hello World page directly from app
http://localhost:3000/api	JSON response from app
http://localhost:8080/	Same as above, via Nginx proxy
http://localhost:8080/api	Same as above, via Nginx proxy

