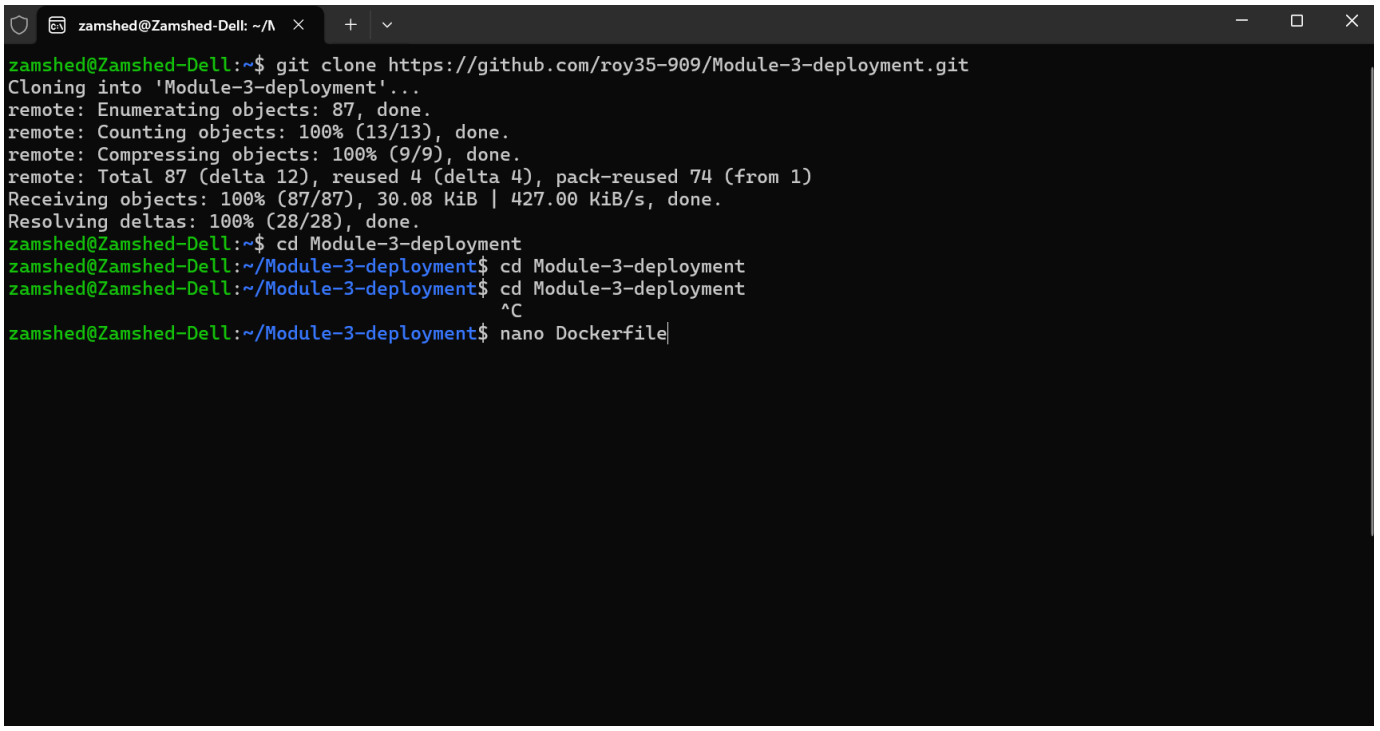
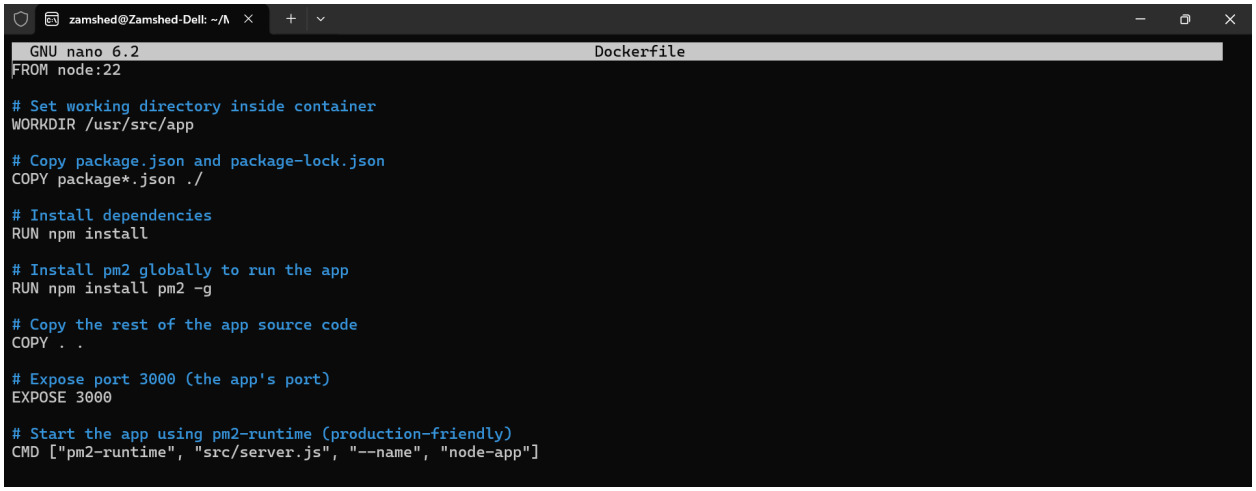
1.Cloning the express.js project repo



2.Creating & Editing Dockerfile

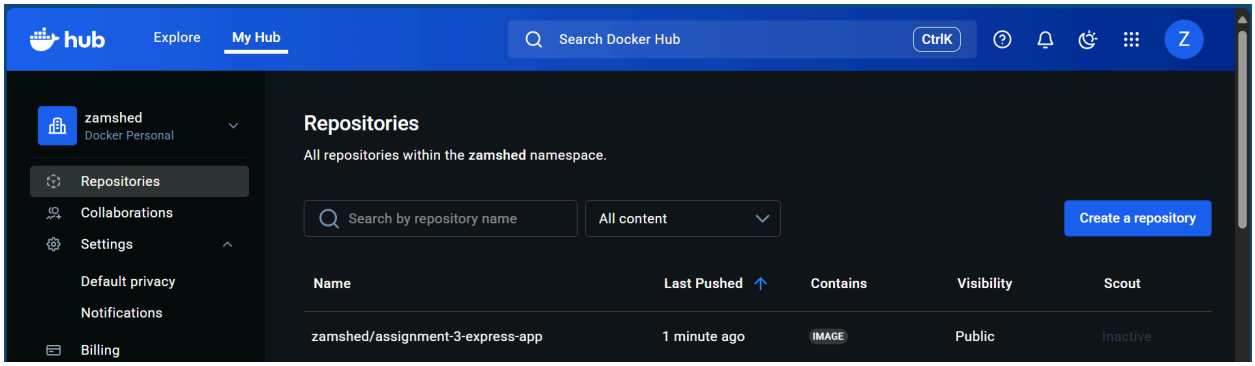


3. Logging in Dockerhub

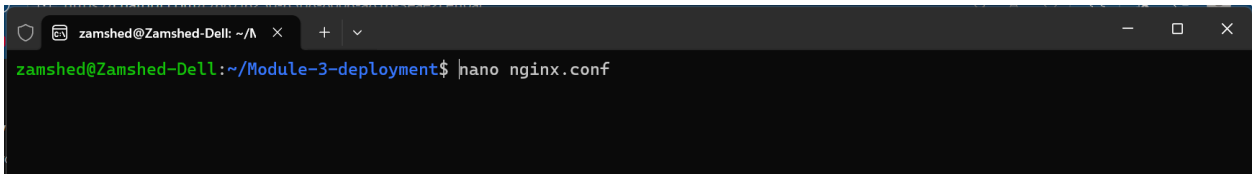
```
zamshed@Zamshed-Dell:~/Module-3-deployment$ docker login

USING WEB-BASED LOGIN

i 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
```

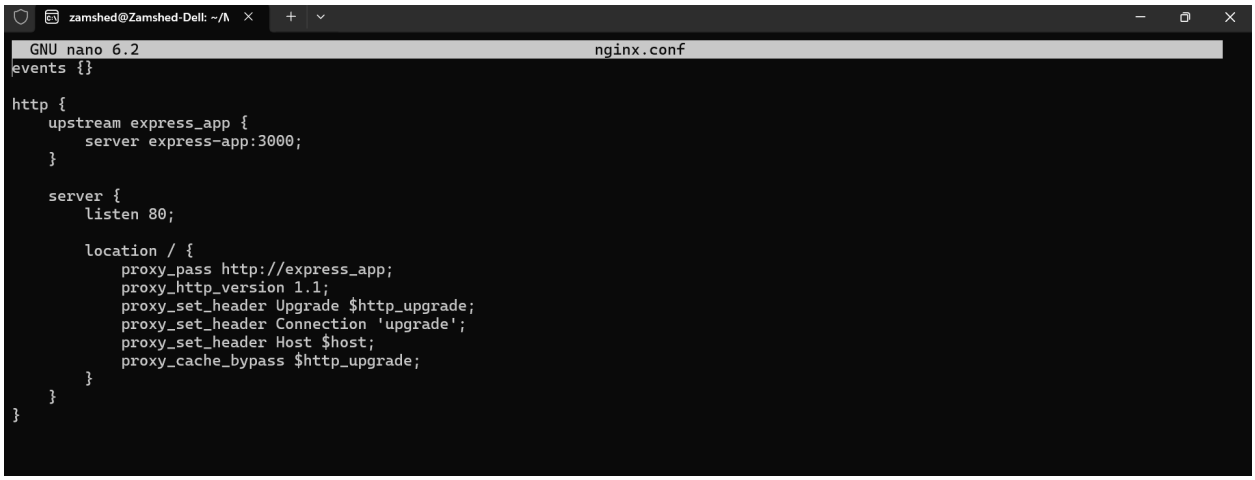
4.Building , then pushing image in Dockerhub

```
zamshed@Zamshed-Dell:~/Module-3-deployment$ docker build -t zamshed/assignment-3-express-app:latest .
[+] Building 2.4s (12/12) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 496B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/node:22                                                                    2.1s
 => [auth] library/node:pull token for registry-1.docker.io                                                                   0.0s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [1/6] FROM docker.io/library/node:22@sha256:079b6a683dc47a87673a6159c9e9b22b0687d04533087cf144c96fac8c26ecd3               0.0s
 => [internal] load build context                                                                                             0.0s
 => => transferring context: 3.31kB                                                                                           0.0s
 => CACHED [2/6] WORKDIR /usr/src/app                                                                                         0.0s
 => CACHED [3/6] COPY package*.json ./                                                                                        0.0s
 => CACHED [4/6] RUN npm install                                                                                              0.0s
 => CACHED [5/6] RUN npm install pm2 -g                                                                                       0.0s
 => [6/6] COPY . .                                                                                                            0.1s
 => exporting to image                                                                                                        0.1s
 => => exporting layers                                                                                                       0.0s
 => => writing image sha256:6c29e9699bb7249181b92360b226bae55695125b74a4c2c1cb5be8b69c6af873                                  0.0s
 => => naming to docker.io/zamshed/assignment-3-express-app:latest                                                            0.0s
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
```

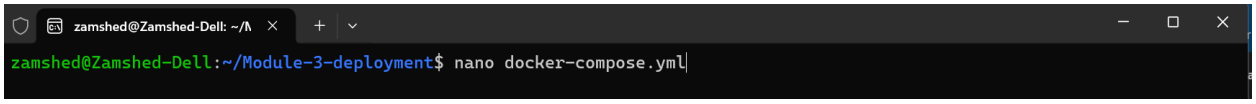


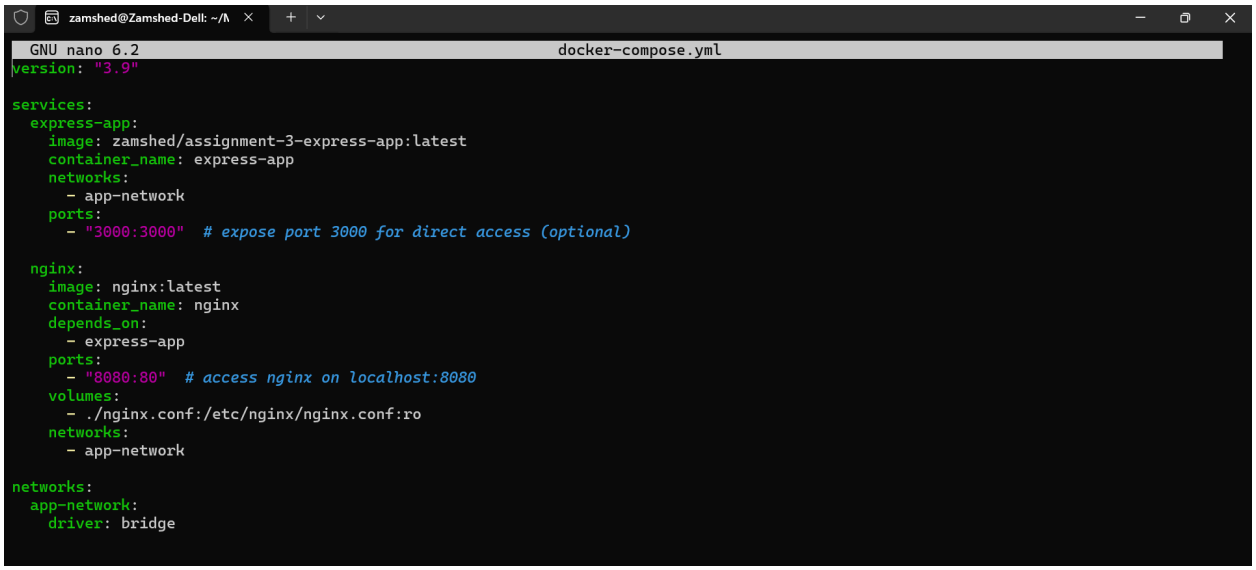
5. Editing & Creating nginx.conf file



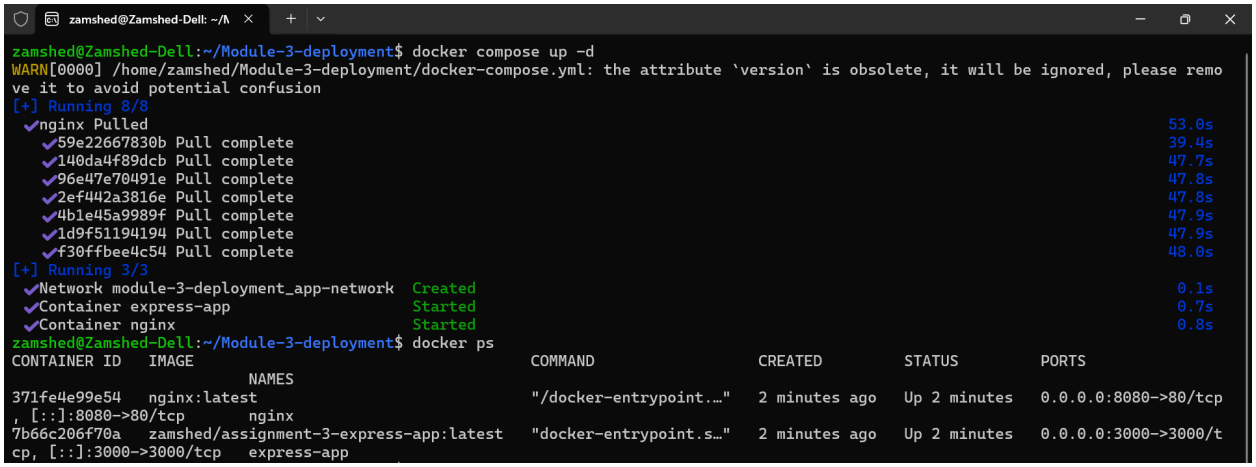


6. Editing & Creating docker-compose.yaml file





7. Crating container using the compose file



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