

Nestjs , Docker and Prisma Common Problem when built

Best way to fix prisma common problem with nestjs and docker

```
nma-auth | PrismaClientInitializationError: Prisma Client could not locate the Query Engine for runtime "debian-openssl-1.1.x".
```

```
nma-auth |
```

```
nma-auth | This happened because Prisma Client was generated for "debian-openssl-3.0.x", but the actual deployment required "debian-openssl-1.1.x".
```

```
nma-auth | Add "debian-openssl-1.1.x" to `binaryTargets` in the "schema.prisma" file and run `prisma generate` after saving it:
```

```
nma-auth |
```

```
nma-auth | generator client {
```

```
nma-auth |   provider      = "prisma-client-js"
```

```
nma-auth |   binaryTargets = ["native", "linux-musl-openssl-3.0.x", "debian-openssl-1.1.x"]
```

```
nma-auth | }
```

```
nma-auth |
```

```
nma-auth | The following locations have been searched:
```

```
nma-auth |   /usr/src/app/node_modules/.prisma/client
```

```
nma-auth |   /usr/src/app/dist/apps
```

```
nma-auth |   /home/son/nestjs/nestjs-microservices-auth/apps/auth/node_modules/.prisma/client
```

```
nma-auth |   /usr/src/app/.prisma/client
```

```
nma-auth |   /tmp/prisma-engines
```

```
nma-auth |   at tl (webpack://nestjs-microservices-auth/apps/auth/node_modules/.prisma/client/runtime/library.js:64:758)
```

```
nma-auth |   at Object.loadLibrary (webpack://nestjs-microservices-auth/apps/auth/node_modules/.prisma/client/runtime/library.js:111:8974)
```

```
nma-auth |   at _r.loadEngine (webpack://nestjs-microservices-auth/apps/auth/node_modules/.prisma/client/runtime/library.js:112:405)
```

```
nma-auth |   at _r.instantiateLibrary (webpack://nestjs-microservices-auth/apps/auth/node_modules/.prisma/client/runtime/library.js:111:12308)
```

```
nma-auth |   at _r.start (webpack://nestjs-microservices-auth/apps/auth/node_modules/.prisma/client/runtime/library.js:112:1965)
```

```
nma-auth |   at Proxy.onModuleInit (webpack://nestjs-microservices-auth/apps/auth/src/prisma.service.ts:16:9)
```

```
nma-auth |   at async Promise.all (index 0)
nma-auth |   at async callModuleInitHook (/usr/src/app/node_modules/@nestjs/core/hooks/on-module-
nma-auth |   init.hook.js:43:5)
nma-auth |   at async NestApplication.callInitHook (/usr/src/app/node_modules/@nestjs/core/nest-application-
nma-auth |   context.js:234:13)
nma-auth |   at async NestApplication.init (/usr/src/app/node_modules/@nestjs/core/nest-application.js:100:9) {
nma-auth |     clientVersion: '6.2.1',
nma-auth |     errorCode: undefined
nma-auth |   }
nma-auth |
nma-auth | Node.js v21.7.3
nma-auth | No errors found.
```

The solution:

edit the config docker compose file and add the volume to set fixed path on the node modules of the external host (which is the docker container):

```
volumes:
  - ./usr/src/app
  - /usr/src/app/node_modules/.prisma
```

see my docker-compose.yaml

```
services:
  auth:
    build:
      context: .
      dockerfile: ./apps/auth/Dockerfile
      target: development
    command: npm run start:debug auth
    container_name: nma-auth
    env_file:
      - ./apps/auth/.env
    ports:
      - '3000:3000'
      - '9229:9229'
    volumes:
      - ./usr/src/app
      - /usr/src/app/node_modules/.prisma
```

Revision #2

Created 11 January 2025 02:53:42 by Son.KyLuat

Updated 11 January 2025 03:00:34 by Son.KyLuat