DevOps

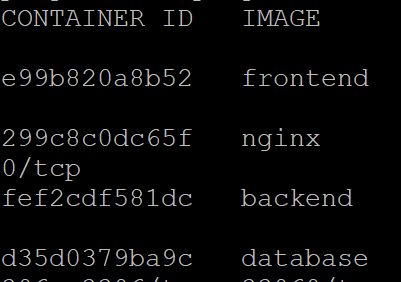
1. Infrastructure Architecture Diagram

Add image

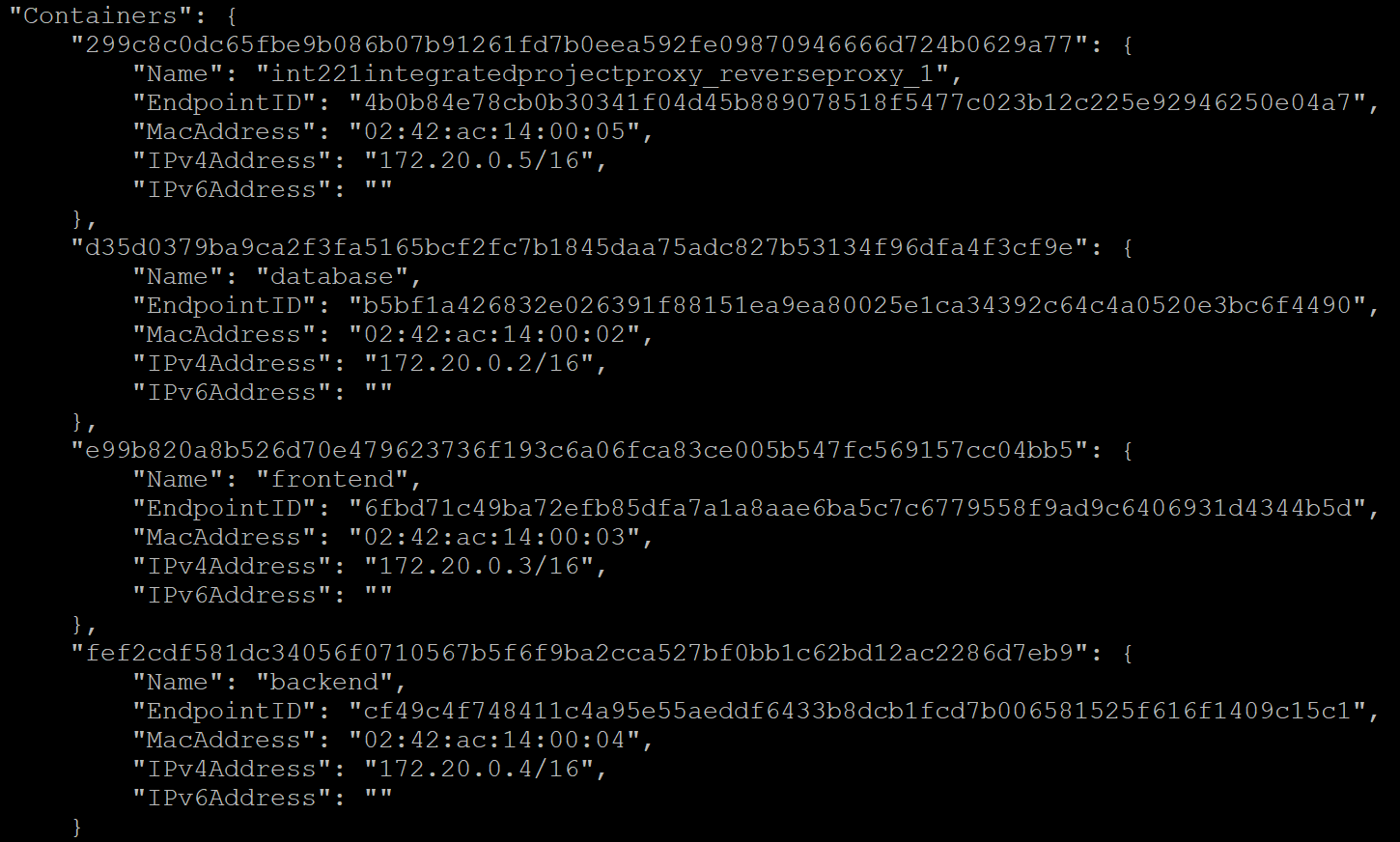
* 1. proxy :

http://104.215.186.188/

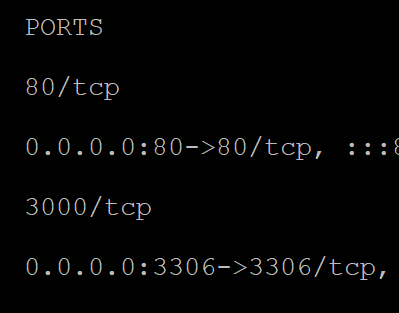
* 1. containers :



* 1. IP addresses (host, containers) :



* 1. open ports (host, containers) :



* 1. App URL (at proxy) :

<http://www.int221projectcars.games/> @ http://104.215.186.188/

1. List of Docker images that you used with versions/tags

**REPOSITORY TAG**

frontend latest

database latest

backend latest

node latest

openjdk 11.0-slim

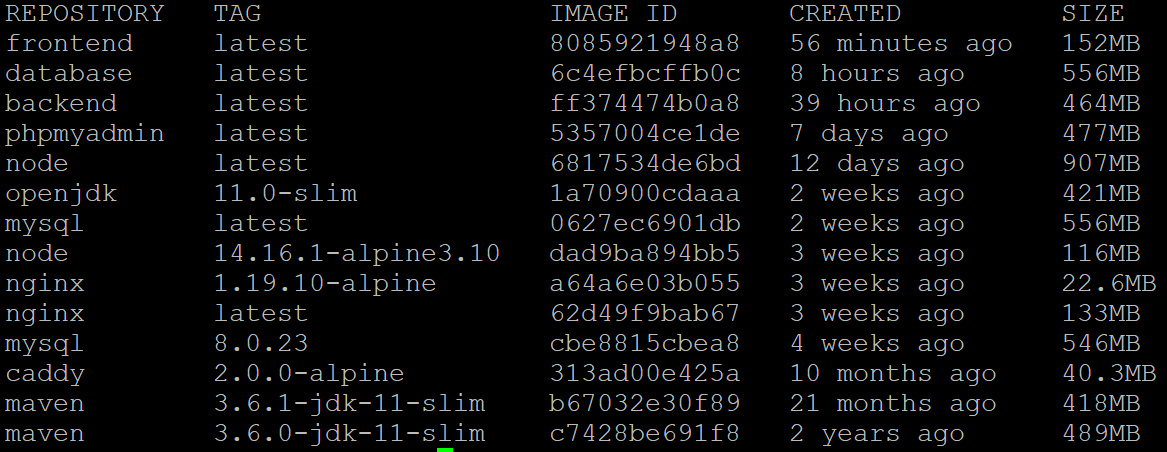
mysql latest

node 14.16.1-alpine3.10

nginx 1.19.10-alpine

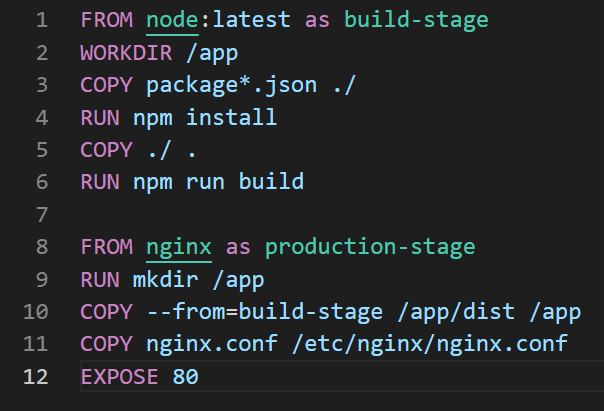
nginx latest

maven 3.6.1-jdk-11-slim

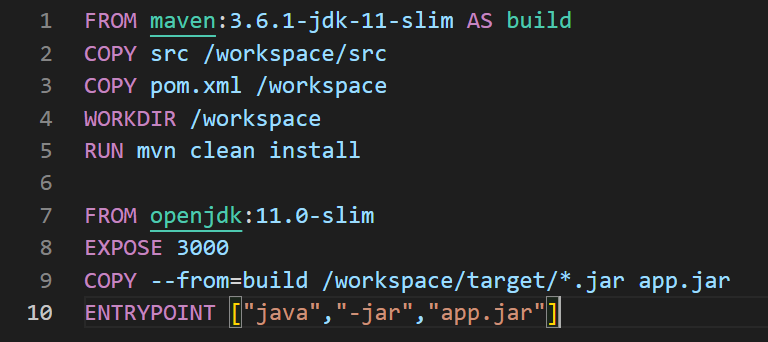


1. Configuration/script files with explanations where required
   1. Dockerfile(s)

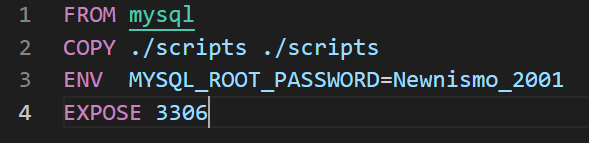
Frontend



Backend

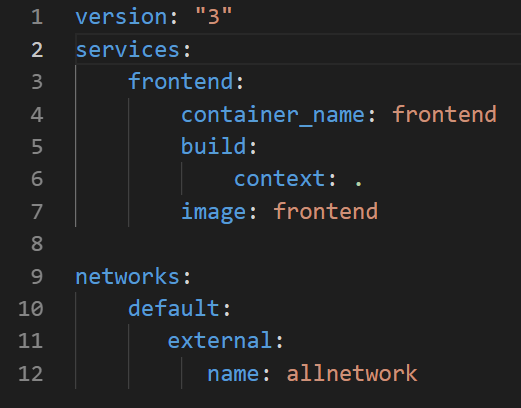


Database

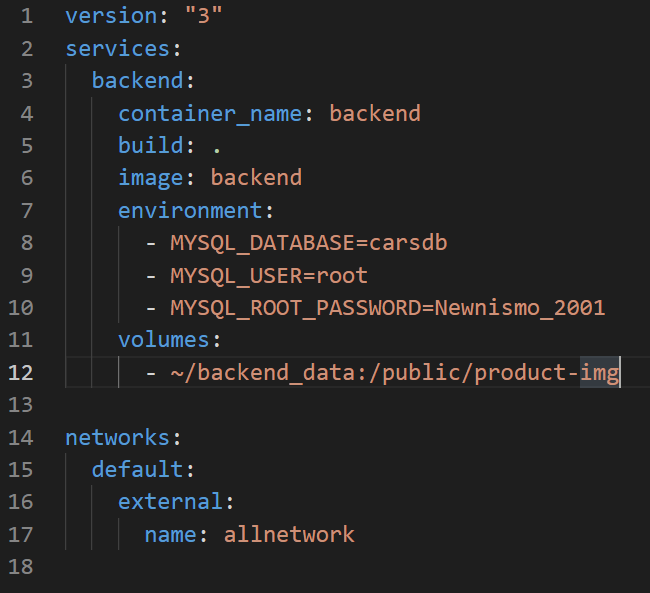


* 1. Docker compose file

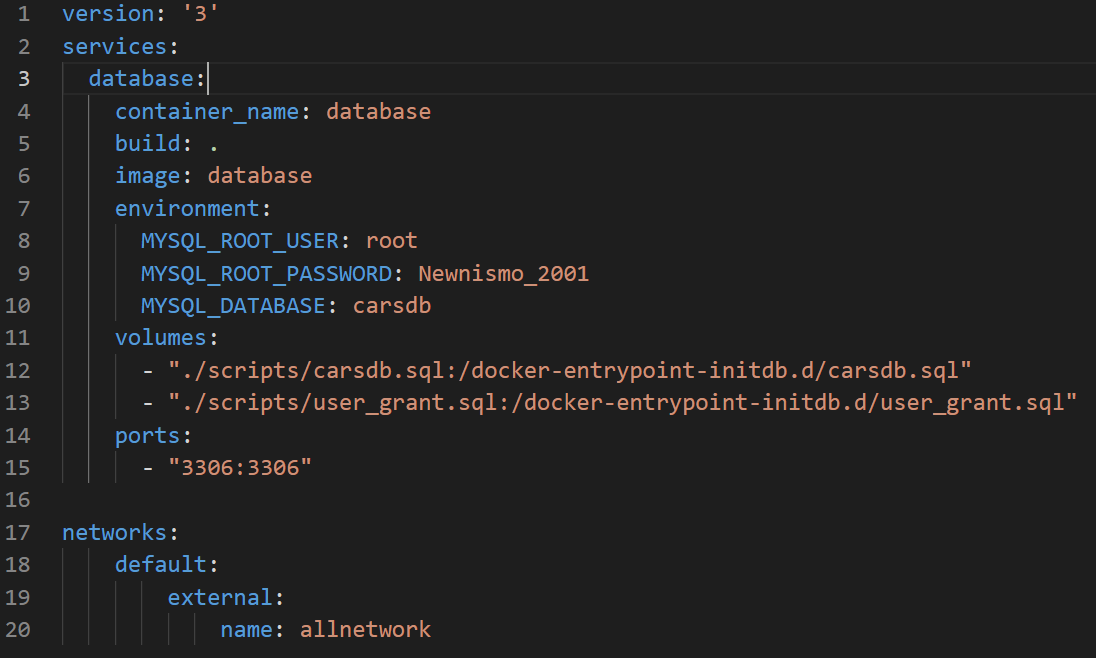
Frontend



Backend



Database

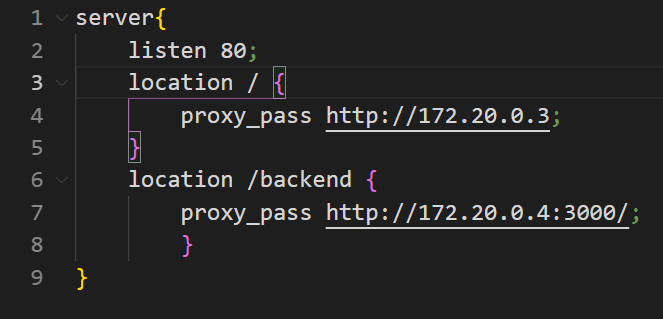


* 1. Proxy configuration file

Docker-compose.yml

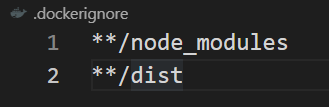


Nginx.conf



* 1. other files that you use with explanation of what you did

**.dockerignore(frontend)**



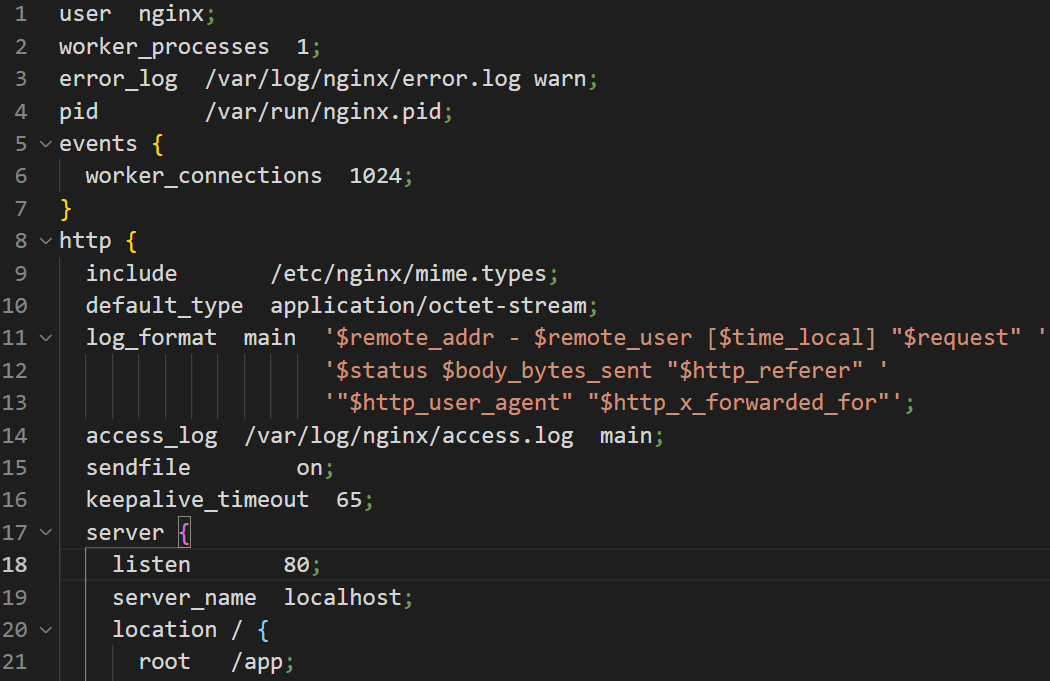
Setting up the .dockerignore file prevents node\_modules and any intermediate build artifacts from being copied to the image which can cause issues during building.

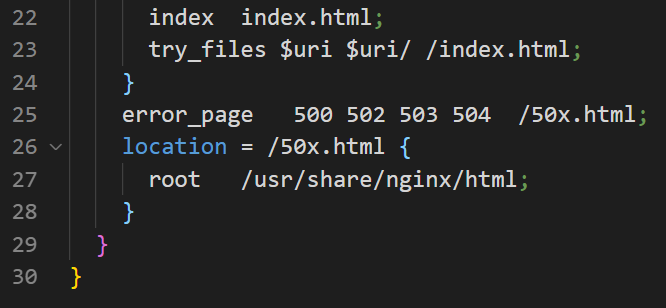
**.env(frontend)**



Set api to backend

**Nginx.conf(frontend)**

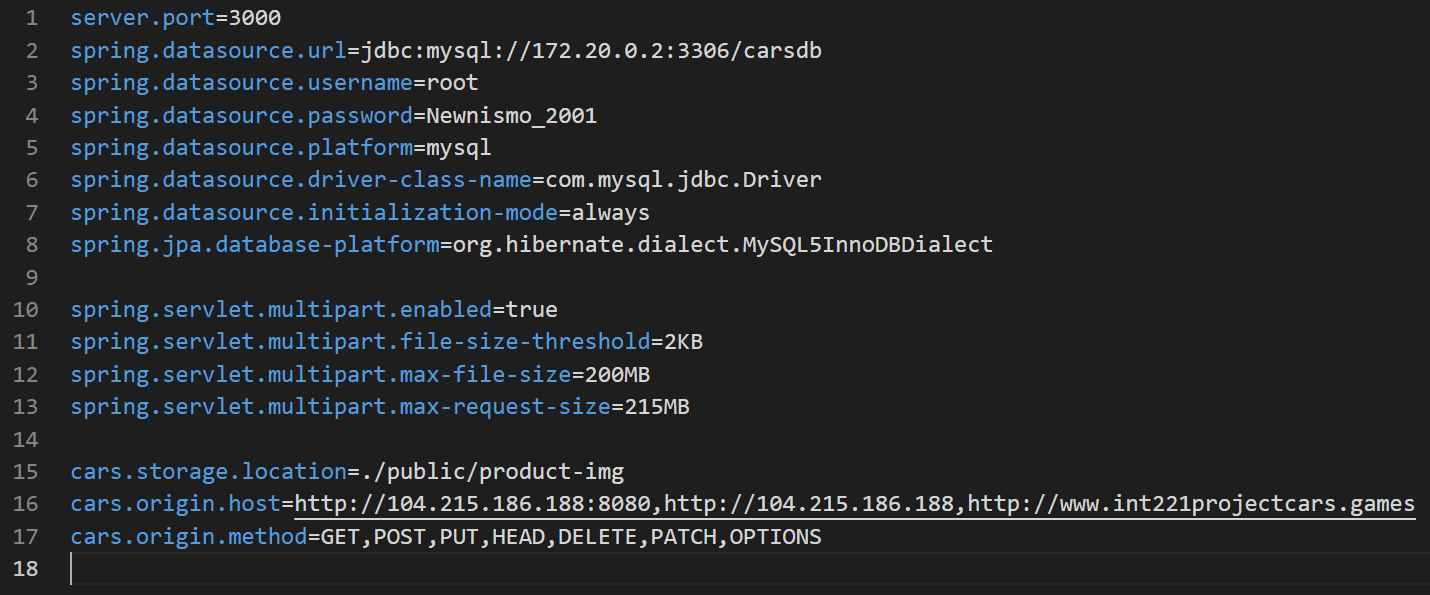




Nginx is an HTTP(s) server that will run in your docker container. It uses a configuration file to determine how to serve content/which ports to listen on/etc. See the nginx configuration documentation for an example of all of the possible configuration options.

The following is a simple nginx configuration that serves your vue project on port 80. The root index.html is served for page not found / 404 errors which allows us to use pushState() based routing.

**Application.properties(backend)**



Use to set origin host fom frontend and lock inner IP of MySQL database

1. Describe configurations that you did to set the environment for FE, BE, proxy

* Frontend

For Dockerfile we use the node image version latest and nginx image for create the image name’s “frontend” and copy the environment from frontend source code to build the image. Use port 80 for container port.

For docker-compose.yml we build container name “frontend” with using image name’s “frontend” and docker network name’s “allnetwork”.

* Backend

For Dockerfile we use maven image for mvn clean install and use openjdk for build image with .jar file that we got from mvn clean install. The image name’s “backend”. Use port 3000 for container port.

For docker-compose.yml we build container name “backend” with using image name’s “backend” and build the environment with database connection. We kept the product-picture in the backend\_data file.

And finally using docker network name’s “allnetwork”.

Application.properties we set server port for 3000 and set the database connect to inner IP network connection with platform mysql.

* Database

For Dockerfile we use mysql image name’s “database” with container port 3306 and set mysql root password.

For docker-compose.yml we build container name’s “database” with using image name’s “database” and set database connection. We run scripts from the scripts file and set server port to 3306.

* Proxy

For docker-compose.yml we build the container name’s “reverseproxy” with using image nginx and kept nginx.conf to default.conf in root. We use port 80 for server and port 80 for container.

For nginx.conf file use server port 80 go to proxy pass to home page website with the path “/” and proxy pass to backend page with path “/backend”.