BUILDING DOCKER IMAGE

sekarang demo menggunakan python

buat folder baru di workspace



masuk ke dalam folder yang di buat

cd my-phython-app/

buat projek

sudo nano app.py  


paste coding ini di dalam app.py  
from flask import Flask, jsonify

app = Flask(\_\_name\_\_)

@app.route('/hello', methods=['GET'])

def hello\_world():

response = {

"return\_code": 200,

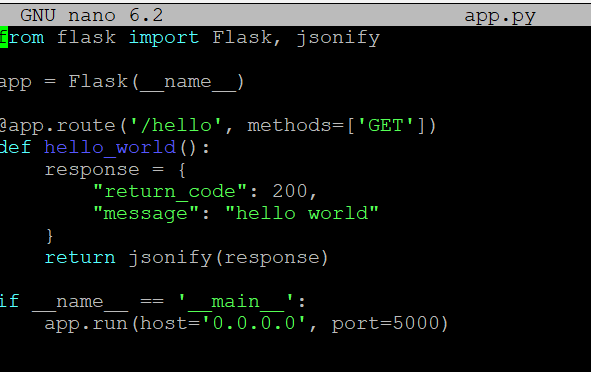
"message": "hello world"

}

return jsonify(response)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)



sudo nano Dockerfile



copy perintah ini  
# Use an official Python runtime as a parent image

FROM python:3.9-slim

# Set the working directory in the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install any needed packages specified in requirements.txt

RUN pip install --no-cache-dir flask

# Make port 5000 available to the world outside this container

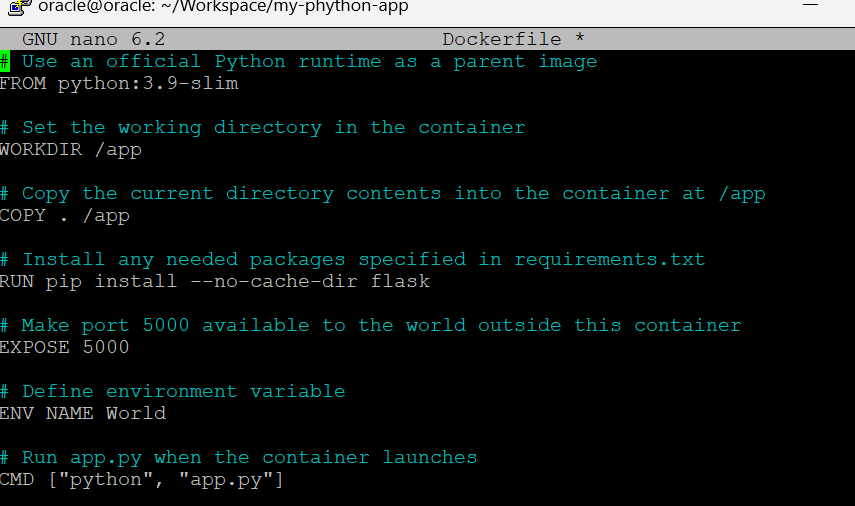
EXPOSE 5000

# Define environment variable

ENV NAME=World

# Run app.py when the container launches

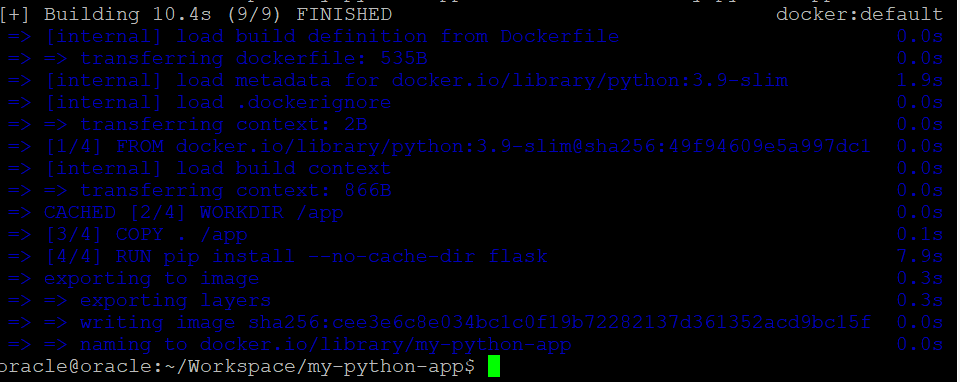
CMD ["python", "app.py"]



jika sudah selesai buka file nya di vscode

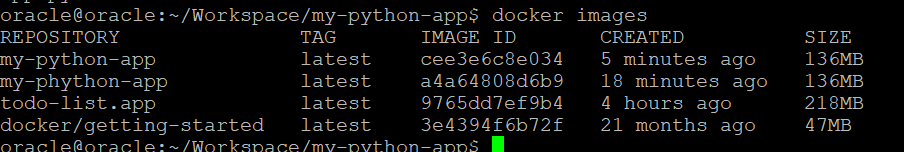
docker build -t My-phython-app .



berhasil di build  


lalu buka cek docker images

docker images



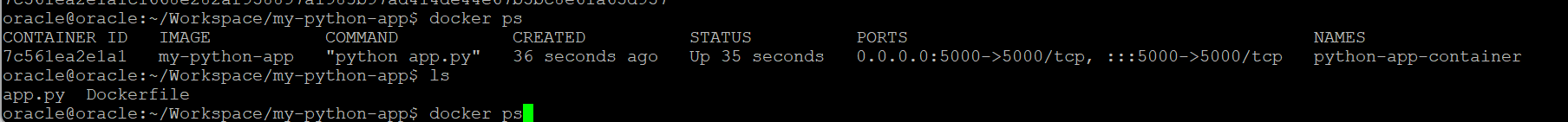
cari file yang baru di buat tadi. kebetulan ada file sama yang ada di pling atas

oracle@oracle:~/Workspace/my-python-app$ docker run -dp 5000:5000 --name python-app-container my-python-app

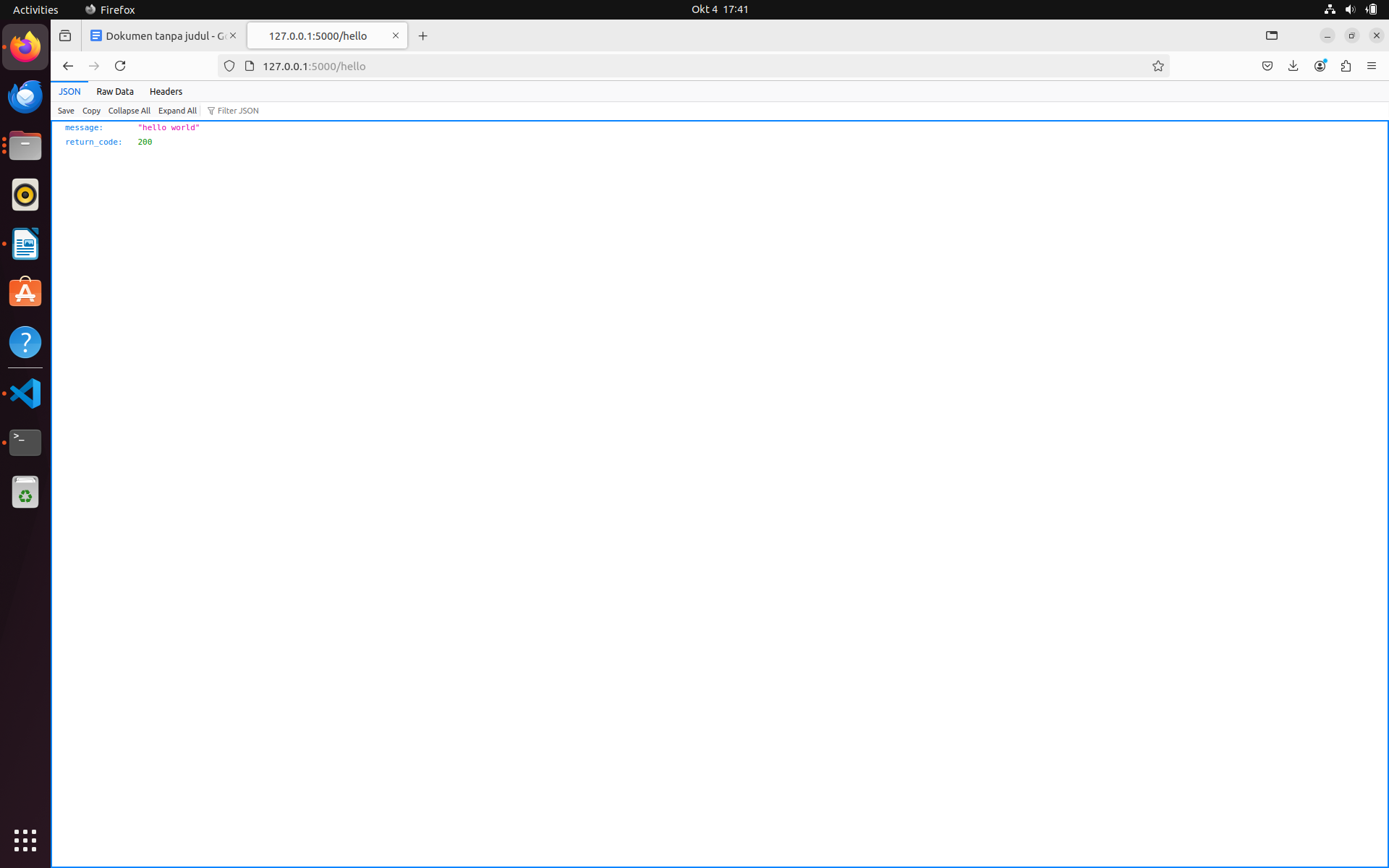


docker ps

image sudah jadi

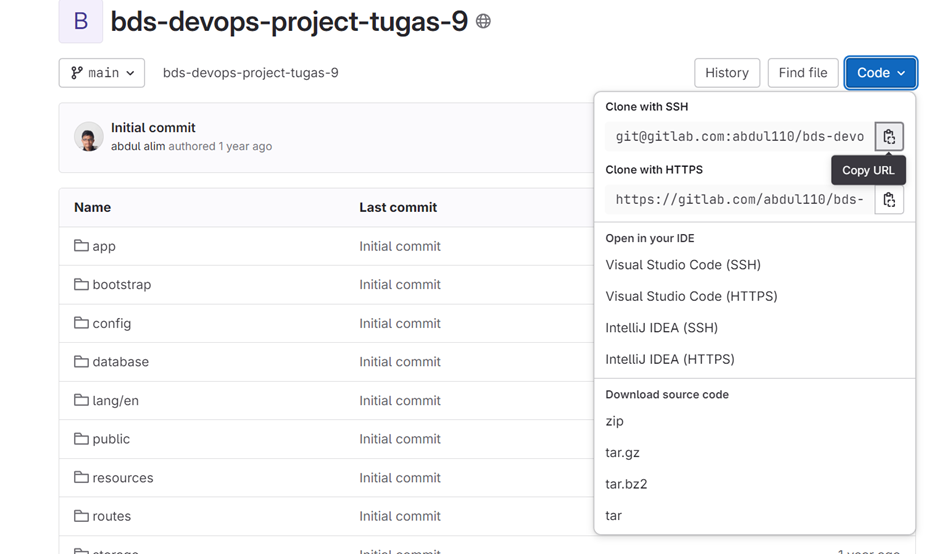


buka browser dan tada sudah terinstall



Tugas INSTALL Laravel dengan Docker

Download file tersebut   
https://gitlab.com/abdul110/bds-devops-project-tugas-9



Download file ini lalu simpan di Folder Workspace atau di mana pun ituhh.

Lalu jangan lupa unzip

Jika tidak download file tersebut lalu masukan ke file git kalian sendiri. Jangan lupa di pull



Lalu buat file Dockerfile

isi nya  
# Menggunakan base image PHP 8.1

FROM php:8.1

# Install dependencies

RUN apt-get update -y && apt-get install -y \

openssl \

zip \

unzip \

git \

libonig-dev \

nodejs \

npm && \

apt-get clean && rm -rf /var/lib/apt/lists/\*

# Install Composer secara global

RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer

# Install PHP extensions yang dibutuhkan

RUN docker-php-ext-install pdo mbstring

# Set working directory

WORKDIR /app

# Copy seluruh project ke dalam container

COPY . /app

# Install dependencies composer

RUN composer install --no-interaction --prefer-dist --optimize-autoloader

# Install dependencies npm

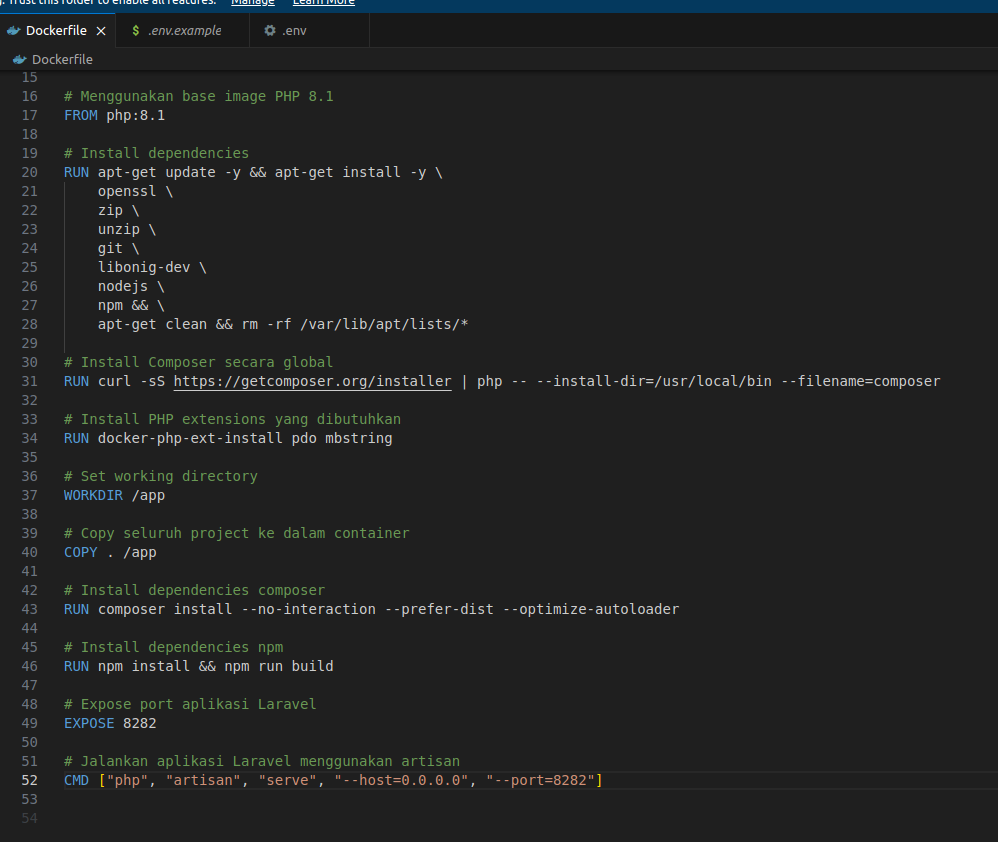
RUN npm install && npm run build

# Expose port aplikasi Laravel

EXPOSE 8282

# Jalankan aplikasi Laravel menggunakan artisan

CMD ["php", "artisan", "serve", "--host=0.0.0.0", "--port=8282"]

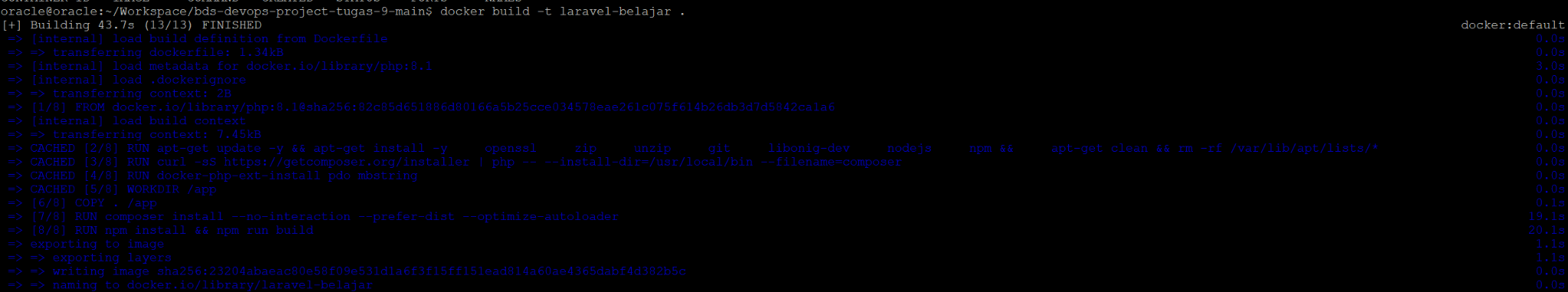


buat file .env

tinggal copy file .env.example



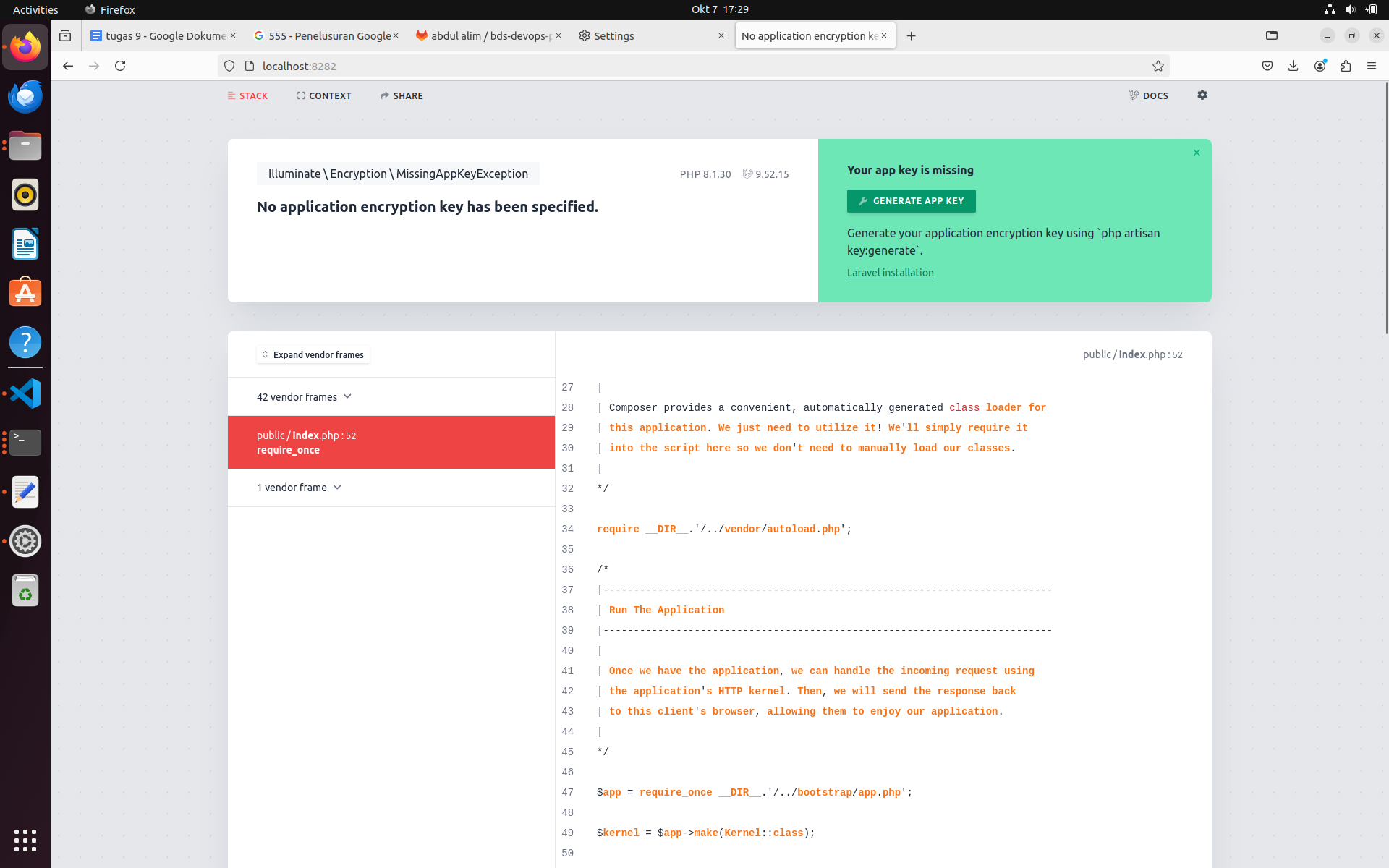
Build docker  
docker build -t laravel-belajar .



jika berhasil run container nyah  
docker run -dp 8282:8282 laravel-belajar

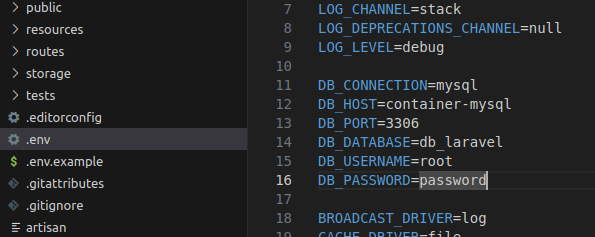
<http://localhost:8282/>

laravel berhasil ter devlop



Menhubungkah ke database

buat koneksi db terlebih dahulu di file .env laravel seperti ini



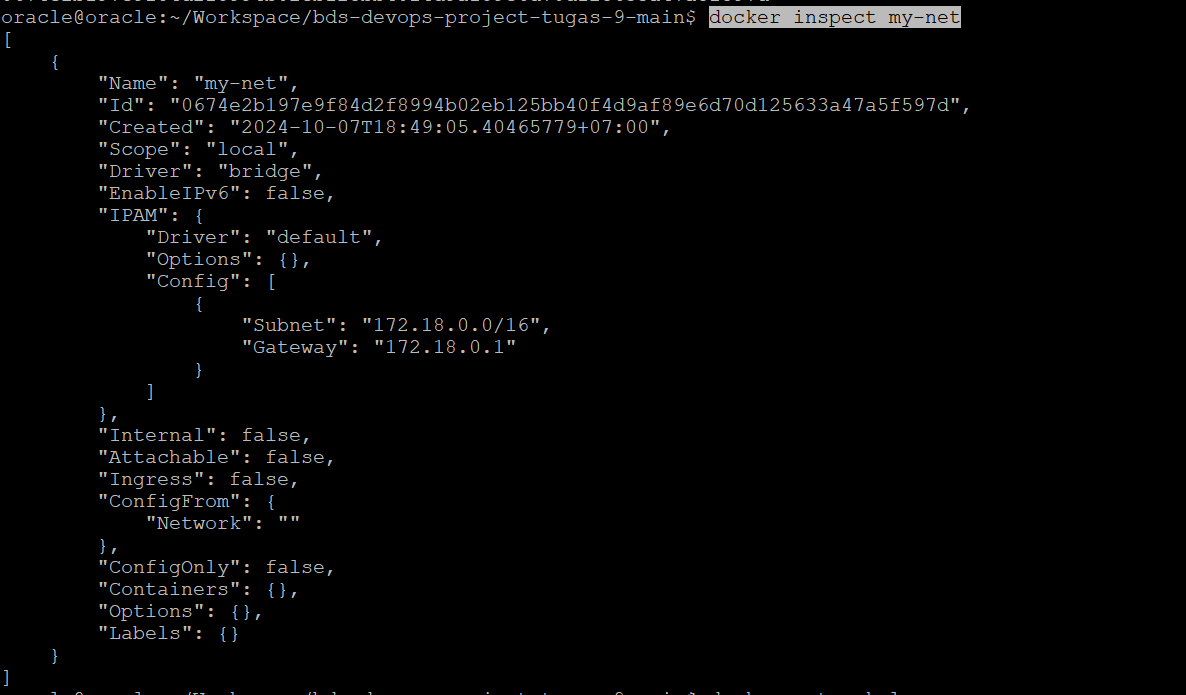
membuat network

docker network create -d bridge my-net



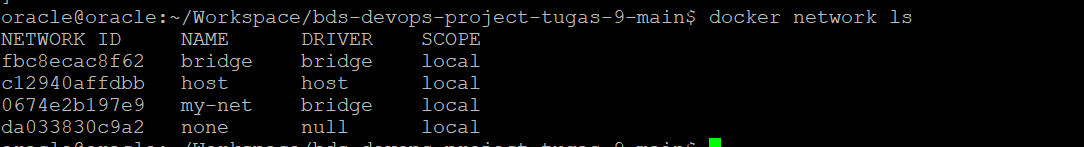
cek network docker atau inspek

docker inspect my-net

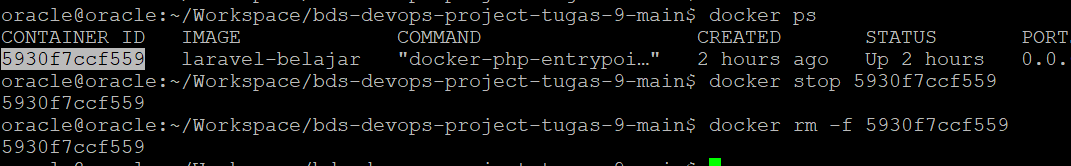


daftar id network

docker network ls



build ulang projeck docket laravel tadi karna baru sajah edit file .env





docker run -dp 8282:8282 --name My-Laravel --network my-net laravel-belajar





docker run --name container-mysql --network my-net -e MYSQL\_ROOT\_PASSWORD=oracle -d mysql



buka

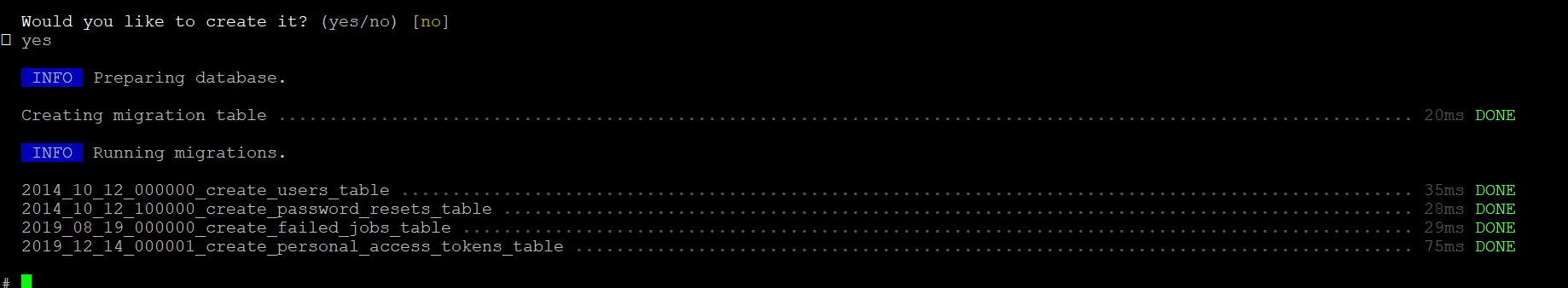
docker exec -it My-Laravel sh



lalu jalankan perintah

php artisan migrate





instalasi selesai di lakukan