D**ocker compose Practical from LMS .**

1. Image Should have proper name and tag
2. Application should be running on port 5000
3. Logs folders should be mounted on the local disk - that means if someone restarts the container logs folder's data should be persisted.
4. Application should run in "Production Mode" not in "Local Mode"
5. Create a docker image an run it in your local environment

Do this task by docker - compose file .

Solution :
Git Link :
https://github.com/techay-mihir-simform/docker_LMS_task/tree/main/docker_compose_practical

Step 1: create docker-compose file according to requirement .
Docker-compose file :

```
#Docker-compose version
version: '3.8'
services:
 #Container name
 containerfirst:
   build: .
   #Image name which has created by build
   image: customnodeimage:latest
   #Access website at 5000 port
   ports:
     - 5000:5000
   volumes:
     - log_data:/app/logs

volumes:
 log_data:
```

Server.js

```
require('dotenv').config()
const express = require('express')
const app = express();

app.get('/healthcheck', (req, res)=>{

   if(process.env.ENV_NODE === "production"){
       res.status(200).send({"code": 2000, "msg": "Production
Mode: Healthcheck is success!"})
       }
   else
       res.status(200).send({"code": 2000, "msg": "Local
Mode: Healthcheck is success!"})
})
app.get('/',(req,res)=>{
   res.end("hello node");
})


app.listen(5000, () =>{
   console.log("Server is running...");
})
```

=> Use docker image for build the custom image .

Dockerfile

```
#Base image as alpine
FROM alpine:latest

#Install nodejs and npm
RUN apk add --no-cache nodejs npm
```

```
#Working directory is /app
WORKDIR /app

#Copy package file to app folder
COPY ./package*.json /app/

#Install dependency
RUN npm install

#Copy all code to /app folder
COPY . /app

#Expose port
EXPOSE 5000

#env variable
ENV ENV_NODE=production

ENTRYPOINT ["node"]
CMD ["server.js"]
```

Step 2:
=> Now Run the application by following command

#docker-compose up --build

Down the container
#docker-compose down