

Cloud Systems HW.1

Seif Yacoub

10/29/2022

GitHub Link: <https://github.com/Saya14798/Cloud-Systems-Homework-1>

A. Part 7:

1.Starting MySQL:

1.1 Creating the network.

The screenshot shows the Visual Studio Code interface with the following components:

- Explorer Panel (Left):** Displays the file structure of the project. The file `dockerfile` is selected and highlighted in blue.
- Editor Panel (Center):** Displays the content of the `dockerfile` file. The content is a Dockerfile for a Node.js application, starting with `FROM node:12-alpine` and including instructions for adding build tools, installing dependencies, and running the application.
- Terminal Panel (Bottom):** Shows the output of a Windows PowerShell command. The command executed is `docker network create to-do-app`, which successfully created the network. The output also shows the current directory path.

Dockerfile Content:

```

1 FROM node:12-alpine
2 # Adding build tools to make yarn install work on Apple silicon / arm64 machines
3 RUN apk add --no-cache python2 g++ make
4 RUN apk add bash
5 WORKDIR /app
6 COPY . .
7 RUN yarn install --production
8 CMD ["node", "src/index.js"]

```

Terminal Output:

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\USER\Desktop> docker network create to-do-app
4bf1422755692891238773238459563c36181d560925d6eabae14402919a
PS C:\Users\USER\Desktop>

```

1.2 Starting a MySQL container and attach it to the network.

The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor and a terminal window at the bottom.

File Explorer (Left Panel):

- EXPLORER
 - P2 - TODO WITH POSTGRES
 - vscode
 - launch.json
 - spec
 - src
 - persistenc
 - routes
 - static
 - index.js
 - dockerfile
 - package-lock.json
 - package.json
 - yarn.lock

Dockerfile (Editor):

```

1 FROM node:12-alpine
2 # Adding build tools to make yarn install work on Apple silicon / arm64 machines
3 RUN apk add --no-cache python2 g++ make
4 RUN apk add bash
5 WORKDIR /app
6 COPY . .
7 RUN yarn install --production
8 CMD ["node", "src/index.js"]
9

```

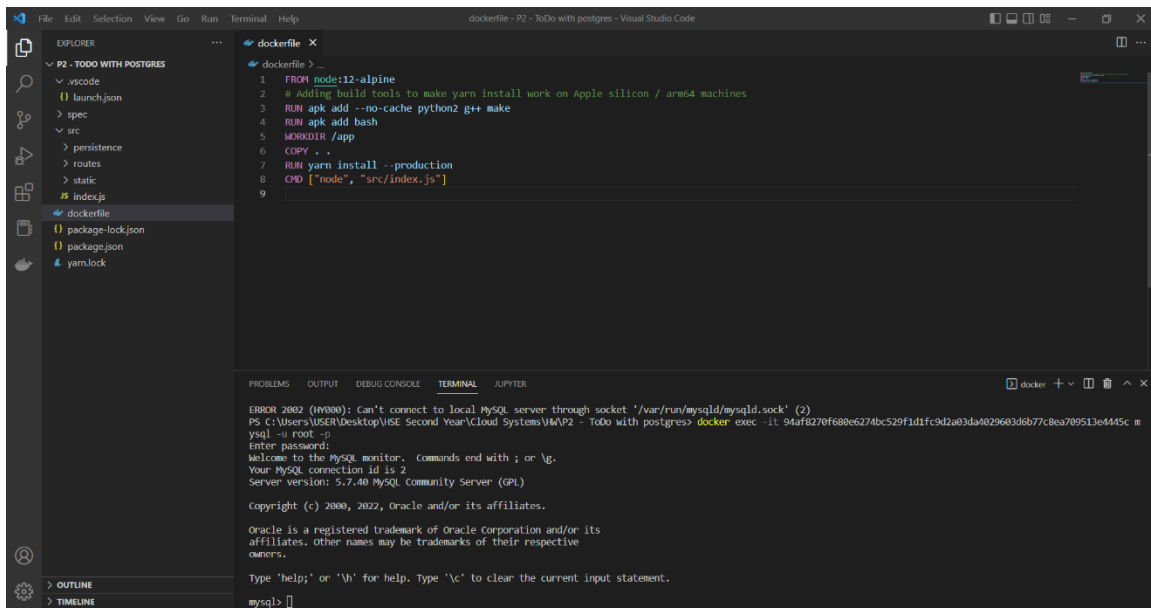
Terminal (Bottom Panel):

```

PS C:\Users\USER\Desktop\HSE Second Year\Cloud Systems\WAP2 - Todo with postgres> docker network create todo-app
Error response from daemon: network with name todo-app already exists
PS C:\Users\USER\Desktop\HSE Second Year\Cloud Systems\WAP2 - Todo with postgres> docker run -d \
>> --network todo-app --network-alias mysql \
>> -v todo-mysql-data:/var/lib/mysql \
>> -e MYSQL_ROOT_PASSWORD=secret \
>> -e MYSQL_DATABASE=todos \
>> mysql:5.7
3557352077cd87c0aa2ced8860c73cb3e508b0838d7c744a07577aab7875df
PS C:\Users\USER\Desktop\HSE Second Year\Cloud Systems\WAP2 - Todo with postgres>

```

1.3 confirming that I have the database up and it is running.



The screenshot shows a Visual Studio Code editor with a Dockerfile open in the editor pane. The Dockerfile contains the following instructions:

```
1 FROM node:12-alpine
2 # Adding build tools to make yarn install work on Apple silicon / arm64 machines
3 RUN apk add --no-cache python2 g++ make
4 RUN apk add bash
5 WORKDIR /app
6 COPY . .
7 RUN yarn install --production
8 CMD ["node", "src/index.js"]
9
```

The terminal pane at the bottom shows the output of a Docker command. It displays an error message about a connection to the local MySQL server, followed by a successful connection to the MySQL monitor. The output includes the MySQL version (5.7.40) and the server name (MySQL Community Server (GPL)).

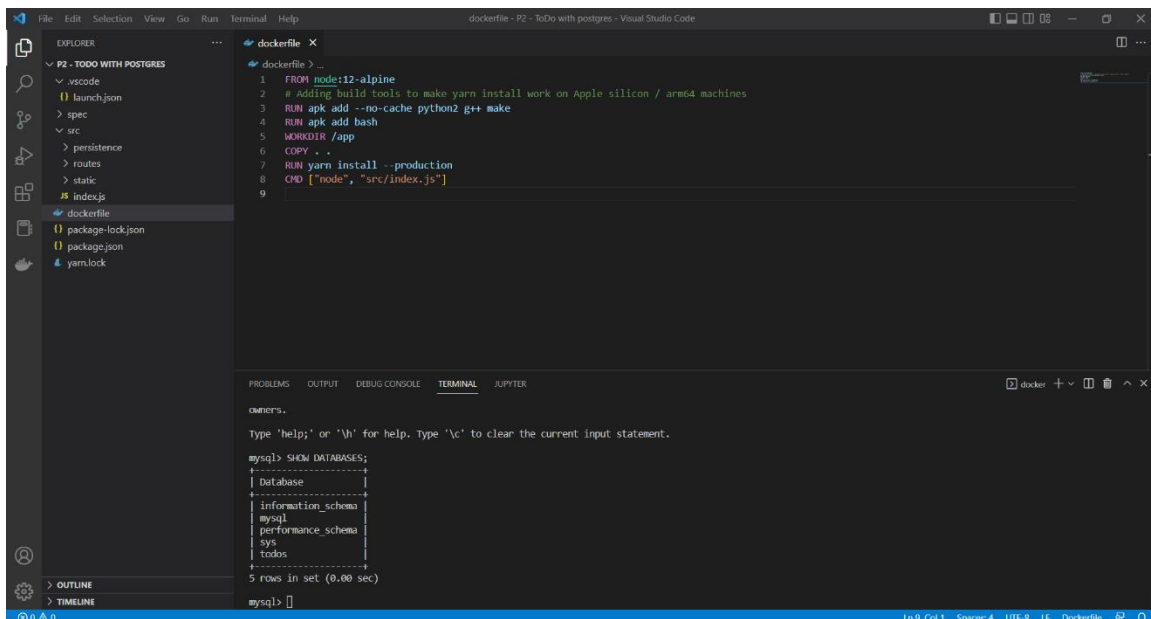
```
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysql/mysql.sock' (2)
PS C:\Users\USER\Desktop\ISE Second Year\Cloud Systems\VM\VP2 - Todo with postgres> docker exec -it 94af8270f680e6274bc529f1d1fc9d2a03da4029603d6b77c8ea709513e445c m
mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



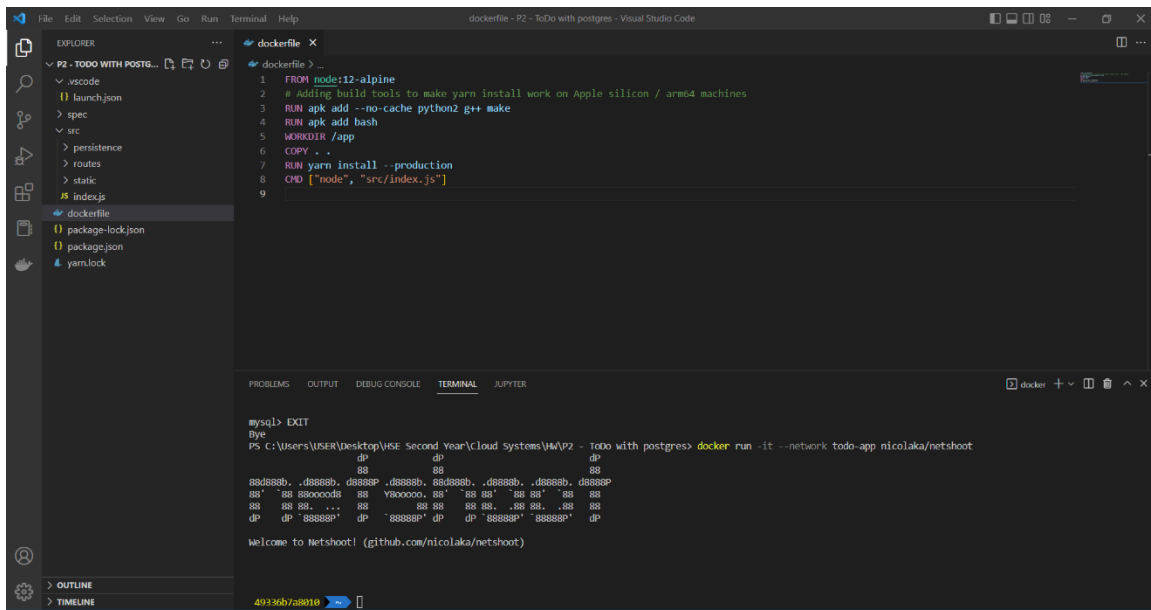
The screenshot shows the same Visual Studio Code editor with the Dockerfile open. The terminal pane now shows the output of a MySQL command to display the database schema. The output lists the databases: information_schema, mysql, performance_schema, sys, and todos. The command was executed successfully, and the output shows 5 rows in set (0.00 sec).

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| todos |
+-----+
5 rows in set (0.00 sec)

mysql>
```

2. Connecting to MySQL:

2.1 Starting a new container using the nicolaka/netshoot image.

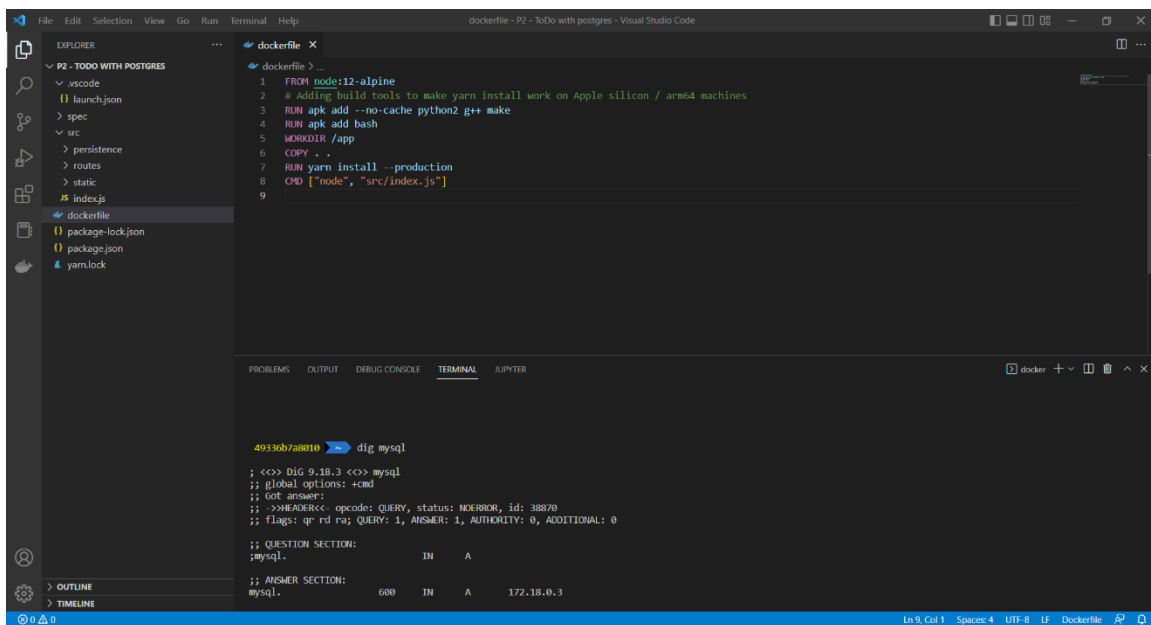


```
1 FROM node:12-alpine
2 # Adding build tools to make yarn install work on Apple silicon / arm64 machines
3 RUN apk add --no-cache python2 g++ make
4 RUN apk add bash
5 WORKDIR /app
6 COPY . .
7 RUN yarn install --production
8 CMD ["node", "src/index.js"]
9
```

```
mysql> EXIT
Bye
PS C:\Users\USER\Desktop\ISE Second Year\Cloud Systems\VM\P2 - ToDo with postgres> docker run -it --network todo-app nicolaka/netshoot
dp      dp      dp
88      88      88
888888b .d8888b .d8888b .d8888b .d8888b .d8888b .d8888b
88" 88 880000da 88 Y000000 88" 88 88" 88 88" 88 88
88 88 88... 88 88 88 88 88 88 88 88 88
dp  dp "888888" dp  dp "888888" "888888" dp

Welcome to Netshoot! (github.com/nicolaka/netshoot)
```

2.2 using dig command.



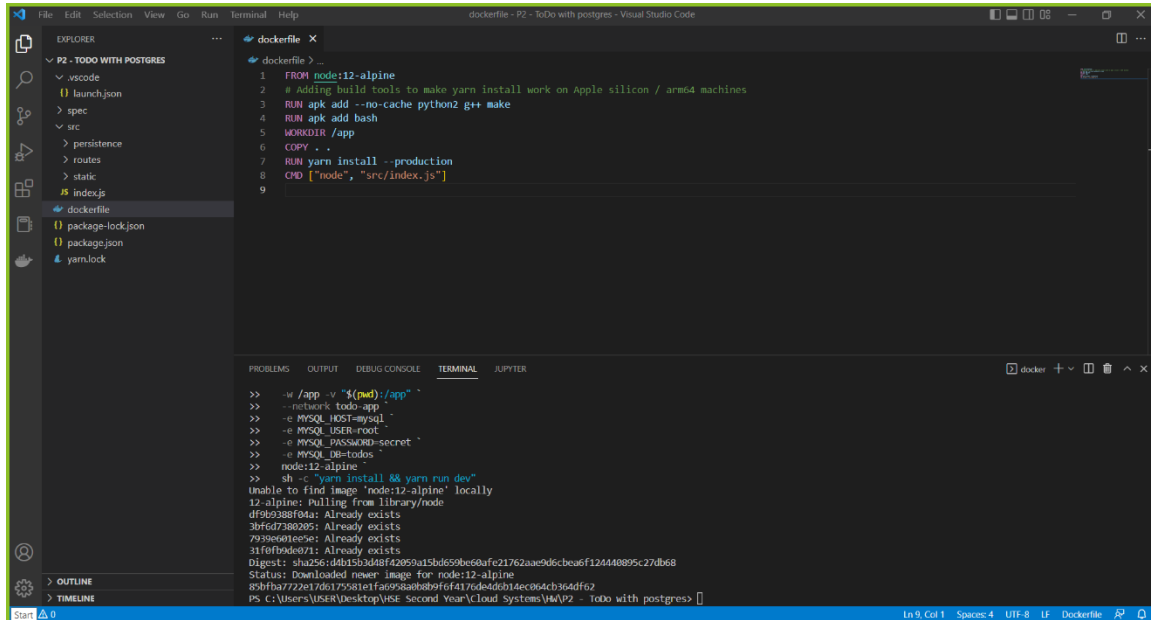
```
1 FROM node:12-alpine
2 # Adding build tools to make yarn install work on Apple silicon / arm64 machines
3 RUN apk add --no-cache python2 g++ make
4 RUN apk add bash
5 WORKDIR /app
6 COPY . .
7 RUN yarn install --production
8 CMD ["node", "src/index.js"]
9
```

```
49136b7a0010 > dig mysql
;<<> DiG 9.18.3 <<> mysql
;; global options: +cmd
;; Got answer:
;;->HEADER: opcode: QUERY, status: NOERROR, id: 38870
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;mysql.
;; ANSWER SECTION:
mysql. 600 IN A 172.18.0.3
```

3. Running my app with MySQL:

3.1 specifying each of the environment variables, connecting the container to my app network.



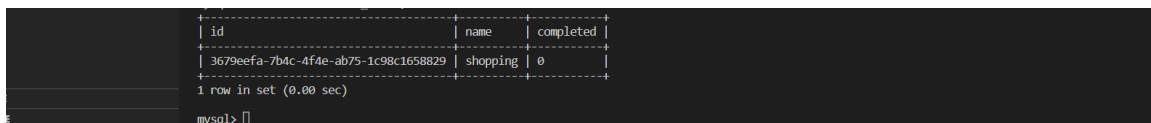
The screenshot shows a Visual Studio Code editor with a Dockerfile open. The Dockerfile contains the following instructions:

```
1 FROM node:12-alpine
2 # Adding build tools to make yarn install work on Apple silicon / arm64 machines
3 RUN apk add --no-cache python2 g++ make
4 RUN apk add bash
5 WORKDIR /app
6 COPY . .
7 RUN yarn install --production
8 CMD ["node", "src/index.js"]
9
```

The terminal output shows the command to run the container with environment variables and network settings:

```
>> ./app -v "${PWD}/app"
>> --network todo-app
>> -e MYSQL_HOST=mysql
>> -e MYSQL_USER=root
>> -e MYSQL_PASSWORD=secret
>> -e MYSQL_DB=todos
>> node:12-alpine
>> sh -c 'yarn install && yarn run dev'
unable to find image 'node:12-alpine' locally
12-alpine: pulling from library/node
df569388f04a: Already exists
30f6d7200205: Already exists
793ee60ee9e: Already exists
31f0fb0de071: Already exists
Digest: sha256:d4b15b3d48f42059a15bd690e60afe21762aae9dcbea6f124440895c27db68
Status: Downloaded newer image for node:12-alpine
85bfba7722e17d6175581e1fa998a08899ef4176d4ddb14ec064cb364df62
PS C:\Users\USER\Desktop\ISE Second Year\Cloud Systems\HW\P2 - Todo with postgres>
```

3.2 Connecting to the mysql database and proving that the items are being written to the database, after writing one row (shopping).



The screenshot shows a MySQL terminal with the following query and result:

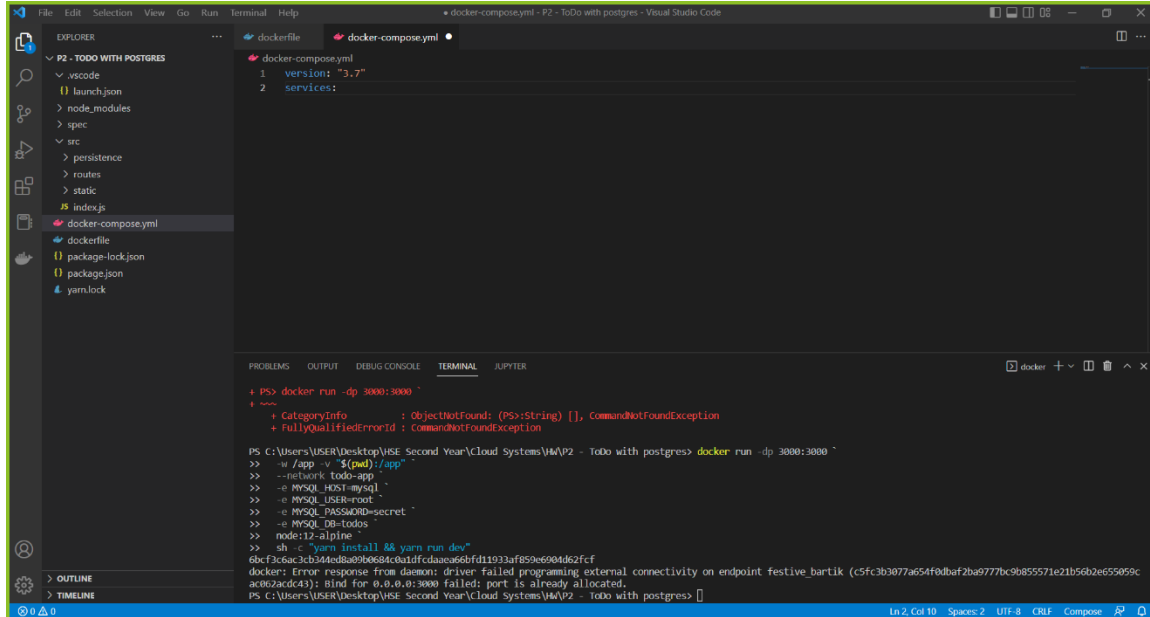
```
mysql> SELECT * FROM shopping;
```

id	name	completed
3679eefa-7b4c-4f4e-ab75-1c98c1658829	shopping	0

1 row in set (0.00 sec)

B. Part 8:

1. Defining the app service after creating the compose file and installing Docker Compose:



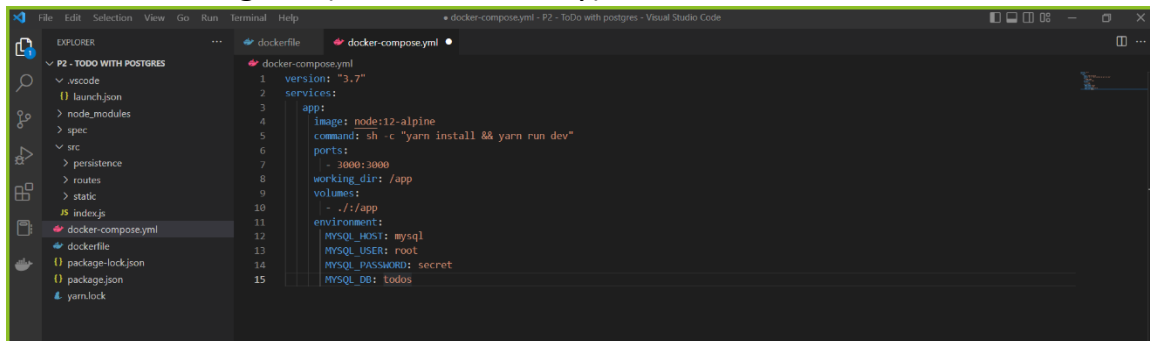
The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the file structure of a project named 'P2 - TODO WITH POSTGRES'. The main editor area shows the 'docker-compose.yml' file with the following content:

```
1 version: "3.7"
2 services:
```

The terminal window at the bottom shows the command 'docker run -dp 3000:3000' being executed. The output indicates that the command failed due to a 'CommandNotFoundException'.

```
PS C:\Users\USER\Desktop\ISE Second Year\Cloud Systems\HW\P2 - Todo with postgres> docker run -dp 3000:3000
docker: Error response from daemon: driver failed programming external connectivity on endpoint festive_bartik (c5fc3b3077a654f0baf2ba9777bc9b855571e21b56b2e65959c
ac0c2acdc43): bind for 0.0.0.0:3000 failed: port is already allocated.
PS C:\Users\USER\Desktop\ISE Second Year\Cloud Systems\HW\P2 - Todo with postgres>
```

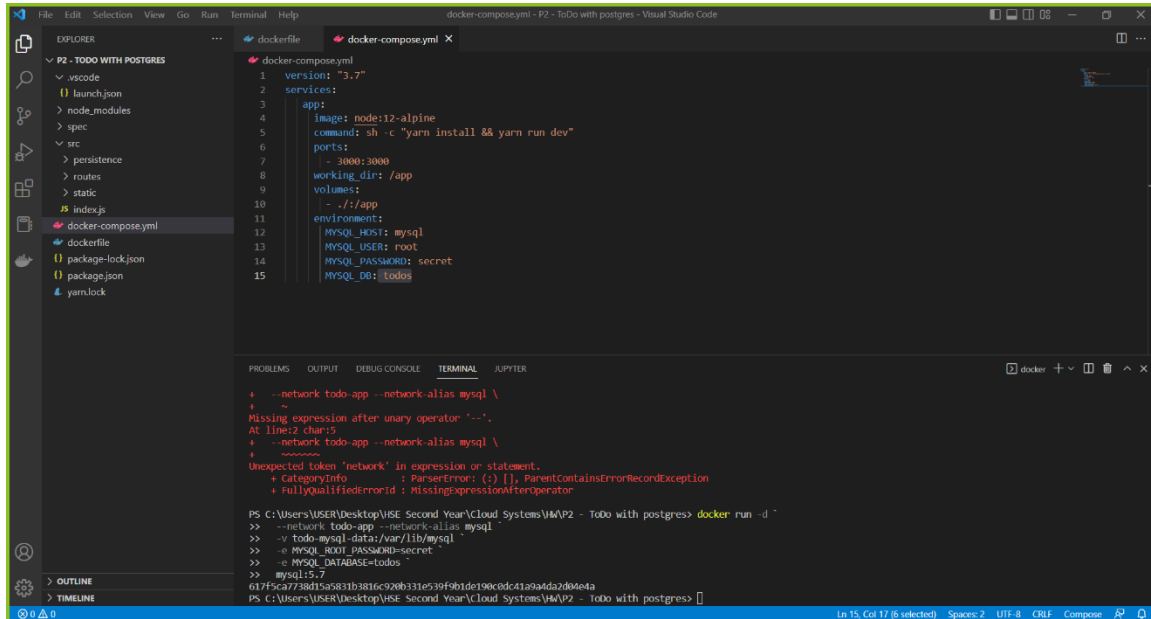
2. defining the service entry and the image for the container and migrating both the working directory, also migrating the environment variable definitions using the (environment key).



The screenshot shows the Visual Studio Code interface with the 'docker-compose.yml' file updated. The file now includes the following content:

```
1 version: "3.7"
2 services:
3   app:
4     image: node:12-alpine
5     command: sh -c "yarn install && yarn run dev"
6     ports:
7       - 3000:3000
8     working_dir: /app
9     volumes:
10      - ./:/app
11     environment:
12       MYSQL_HOST: mysql
13       MYSQL_USER: root
14       MYSQL_PASSWORD: secret
15       MYSQL_DB: todos
```

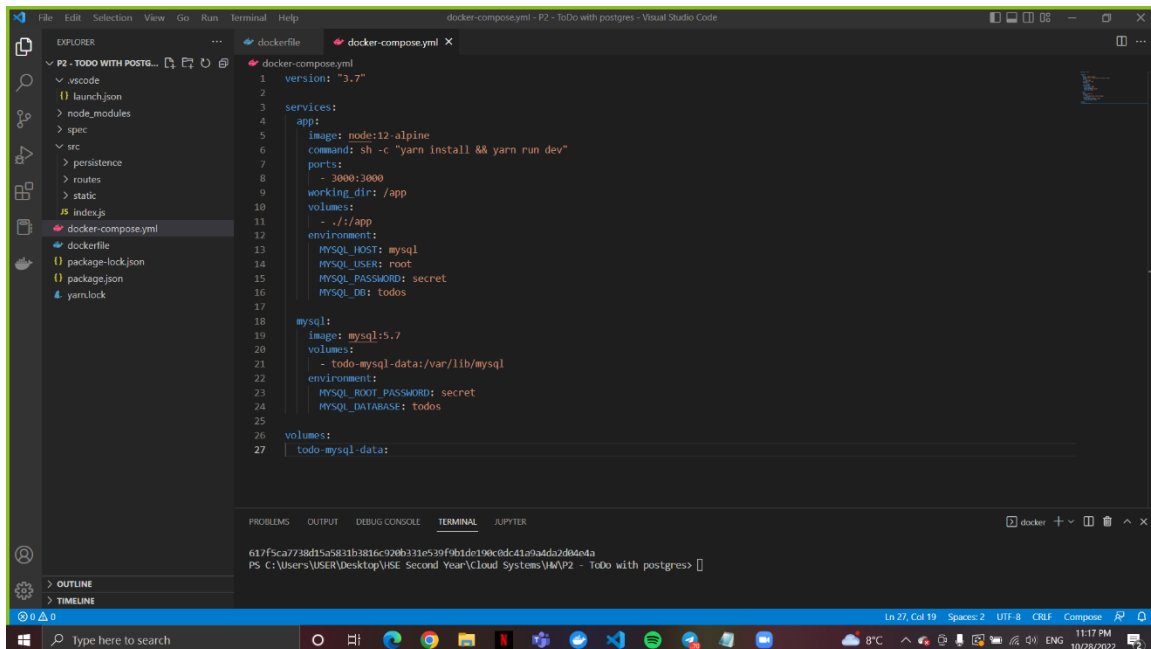
3. Defining the MySQL service.



The screenshot shows the Visual Studio Code editor with a Docker Compose file named `docker-compose.yml` open. The file defines a service named `app` using the `node:12-alpine` image. The terminal window at the bottom displays an error message: "Missing expression after unary operator '~'. At line:2 char:5". The error is related to the `--network` flag in the `docker run` command. The file content is as follows:

```
1 version: "3.7"
2 services:
3   app:
4     image: node:12-alpine
5     command: sh -c "yarn install && yarn run dev"
6     ports:
7       - 3000:3000
8     working_dir: /app
9     volumes:
10      - ./:/app
11     environment:
12       MYSQL_HOST: mysql
13       MYSQL_USER: root
14       MYSQL_PASSWORD: secret
15       MYSQL_DB: todos
```

The terminal output shows the command `docker run -d --network todo-app --network-alias mysql --image mysql:5.7 --volume todo-mysql-data:/var/lib/mysql --env MYSQL_ROOT_PASSWORD=secret --env MYSQL_DATABASE=todos mysql:5.7` and the error message.








The screenshot shows the Visual Studio Code editor with the same Docker Compose file, but now it includes a `mysql` service. The terminal window is empty, indicating that the file is now valid. The file content is as follows:


```
1 version: "3.7"
2 services:
3   app:
4     image: node:12-alpine
5     command: sh -c "yarn install && yarn run dev"
6     ports:
7       - 3000:3000
8     working_dir: /app
9     volumes:
10      - ./:/app
11     environment:
12       MYSQL_HOST: mysql
13       MYSQL_USER: root
14       MYSQL_PASSWORD: secret
15       MYSQL_DB: todos
16
17   mysql:
18     image: mysql:5.7
19     volumes:
20       - todo-mysql-data:/var/lib/mysql
21     environment:
22       MYSQL_ROOT_PASSWORD: secret
23       MYSQL_DATABASE: todos
24
25 volumes:
26   todo-mysql-data:
```

3. Running the application stack.

```
9977ea919cbe1b): Bind for 0.0.0.0:3000 failed: port is already allocated
PS C:\Users\USER\Desktop\HSE Second Year\Cloud Systems\HW\p2 - ToDo with postgres> docker compose up -d
[+] Running 2/2
 - Container p2-todowithpostgres-mysql-1 Running 0.0s
 - Container p2-todowithpostgres-app-1 Started 13.7s
PS C:\Users\USER\Desktop\HSE Second Year\Cloud Systems\HW\p2 - ToDo with postgres> 
```

4. checking the app stack in Docker Dashboard.

<input type="checkbox"/>		p2-todowithpostgres 2 containers	-	Running (2/2)	-	<input type="checkbox"/>
<input type="checkbox"/>		app-1 25969ec66988 	node:12-alpine	Running	3000	6 minutes ago <input type="checkbox"/>
<input type="checkbox"/>		mysql-1 3bb462bc7315 	mysql:5.7	Running	-	8 minutes ago <input type="checkbox"/>

RAM 5.46GB CPU 0.00%  Connected to Hub

Regards,