

- 1) Some part before
- 2) Convert into mysql
- 3) **mkdir todobackendmysql**

ls

```
azureuser@vmdocker:~$ mkdir todobackendmysql
azureuser@vmdocker:~$ ls
PyTodoBackendMonolith  ReactTodoUIMonolith  snap  todobackendmysql
```

- 4) **touch app.py**

ls

```
azureuser@vmdocker:~/todobackendmysql$ touch app.py
azureuser@vmdocker:~/todobackendmysql$ ls
app.py
```

- 5) chatgpt - my application works on sql server convert it into mysql
- 6) **nano app.py** – paste the chatgpt code

Update below details

root@vmdocker: /home/azureuser/todobackendmysql

GNU nano 7.2

app.py

```
import mysql.connector
import uvicorn
import os
from fastapi import FastAPI
from pydantic import BaseModel
from fastapi.middleware.cors import CORSMiddleware
from dotenv import load_dotenv
```

```
# Load environment variables from .env file
load_dotenv()
```

```
# Get connection details from environment
```

```
db_config = {
    'host': "74.226.140.248",
    'user': "azureuser",
    'password': "Mommy7Daddy!",
    'database': "mydatabase"
}
```

```
app = FastAPI()
```

```
# Configure CORS
```

```
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

```
# Define the Task model
```

```
class Task(BaseModel):
    title: str
    description: str
```

```
# Create the Tasks table
```

```
@app.get("/api")
def create_tasks_table():
    try:
        conn = mysql.connector.connect(**db_config)
        cursor = conn.cursor()
        cursor.execute("""
            CREATE TABLE IF NOT EXISTS Tasks (
```

[Read 54 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line	M-F Redo


7) SEARCH – mysql image on docker hub

SQL Server to MySQL Conversio x | mysql image on docker hub - G x | mysql - Official Image | Docker x | +

hub.docker.com/_/mysql

dockerhub Search Docker Hub

Explore / Official Images / mysql

 **mysql** Docker Official Image · ↓ 1B+ · ☆ 10K+
MySQL is a widely used, open-source relational database management system (RDBMS).
DATABASES & STORAGE

Overview Tags

8) **docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=Mommy7Daddy! -e MYSQL_DATABASE=mydatabase -e MYSQL_USER=azureuser -d mysql:latest** = command for mysql db

```
root@vmdocker:/home/azureuser/todobackendmysql# docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=Mommy7Daddy! -e MYSQL_DATABASE=mydatabase -e MYSQL_USER=azureuser -d mysql:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
ceal172a6e83b: Pull complete
daac2c594bdd: Pull complete
cb8acbf2440c: Pull complete
fae51f7de1fb: Pull complete
b2ead3e96e6b: Pull complete
769c3ac51f88: Pull complete
79f239a40e62: Pull complete
c11056354384: Pull complete
49978e7ccddf: Pull complete
548990e33276: Pull complete
Digest: sha256:0596fa224cdf3b3355ce3ddbdf7ce77be27ec9e51841dfc5d2e1c8b1eea69d2
Status: Downloaded newer image for mysql:latest
079b0f8cba9630f46d2828d3046b87aec9babbecf5c4d99e3c28e246713f25fc
```

9) **docker ps**

```
root@vmdocker:/home/azureuser/todobackendmysql# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
079b0f8cba96   mysql:latest   "docker-entrypoint.s..."  30 seconds ago   Up 29 seconds   3306/tcp, 33060/tcp      some-mysql
root@vmdocker:/home/azureuser/todobackendmysql#
```

Port - 3306

10) Update as below

```
# Get connection details from environment
db_config = {
    'host': "74.226.140.248:3306",
    'user': "azureuser",
    'password': "Mommy7Daddy!",
    'database': "mydatabase"
}
```

11) touch Dockerfile

Nano Dockerfile

root@vmdocker: /home/azureuser/todobackendmysql

GNU nano 7.2

Dockerfile *

```
# Use the official Python image as the base image
FROM python:3.9.20

# Set the working directory in the container
WORKDIR /backendapp/

# Copy the application files into the container
COPY . /backendapp/

# Install necessary packages
RUN apt-get update && apt-get install -y unixodbc unixodbc-dev
RUN curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
RUN curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.list
RUN apt-get update
RUN ACCEPT_EULA=Y apt-get install -y msodbcsql17

RUN pip install -r requirements.txt

# Start the FastAPI application
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]
```

12) Copy Readme.md and cp requirements.txt files and paste in todobackendmysql directory.

cp Readme.md /home/azureuser/todobackendmysql/

cp requirements.txt /home/azureuser/todobackendmysql/

13) **docker build -t todobackendmysql . = create docker image**

docker images

```
root@vmdocker:/home/azureuser/todobackendmysql# docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
todobackendmysql    latest      c992b369a1e9  18 seconds ago  1.11GB
pytodo               latest      8badf51efb08  17 hours ago   1.11GB
```

14) Now running pytodo image container of backend one

cd ..

ls

cd PyTodoBackendMonolith/

ls

```
root@vmdocker:/home/azureuser# cd PyTodoBackendMonolith/
root@vmdocker:/home/azureuser/PyTodoBackendMonolith# ls
Dockerfile  Readme.md  app.py  requirements.txt
root@vmdocker:/home/azureuser/PyTodoBackendMonolith# docker run
```

15) **docker run -d -p 8000:8000 pytodo**

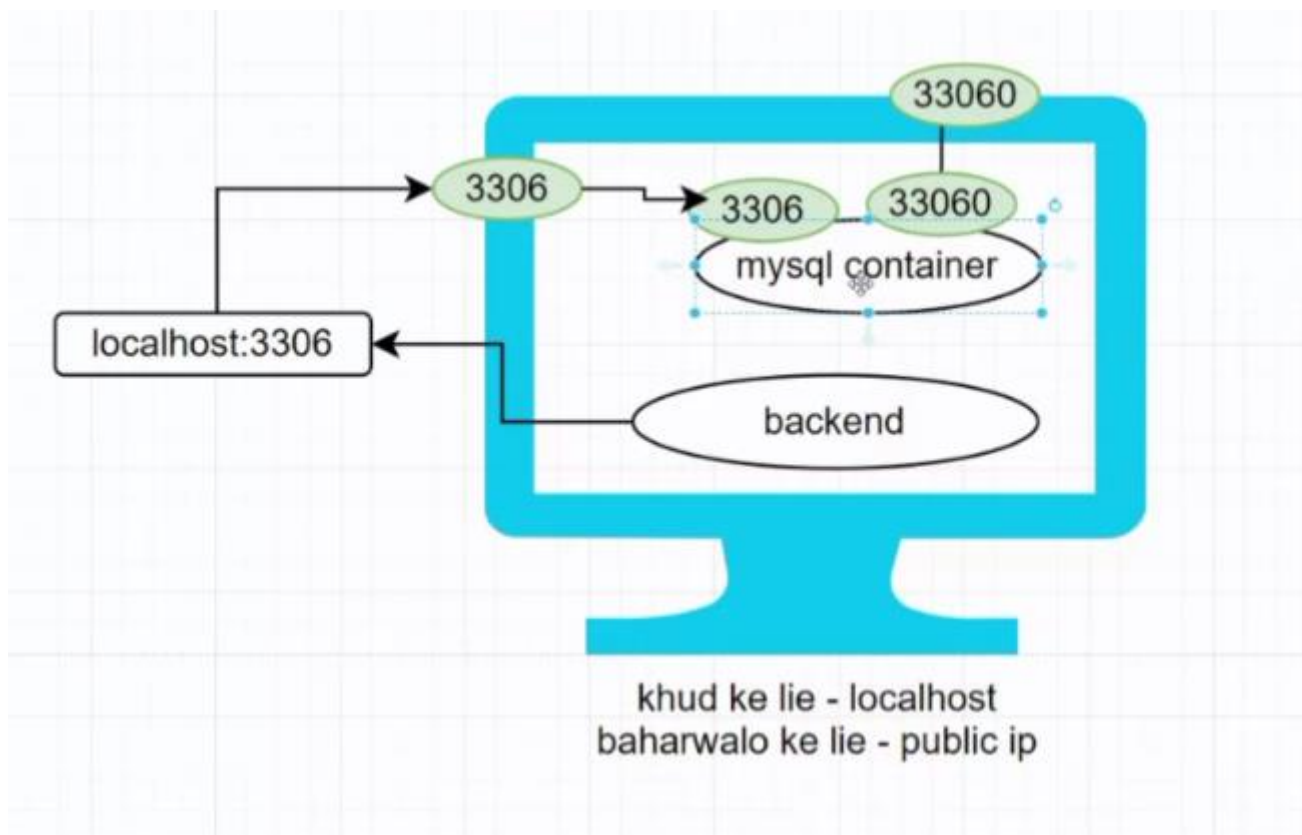
```
root@vmdocker: /home/azureuser/PyToDoBackendMonolith
```

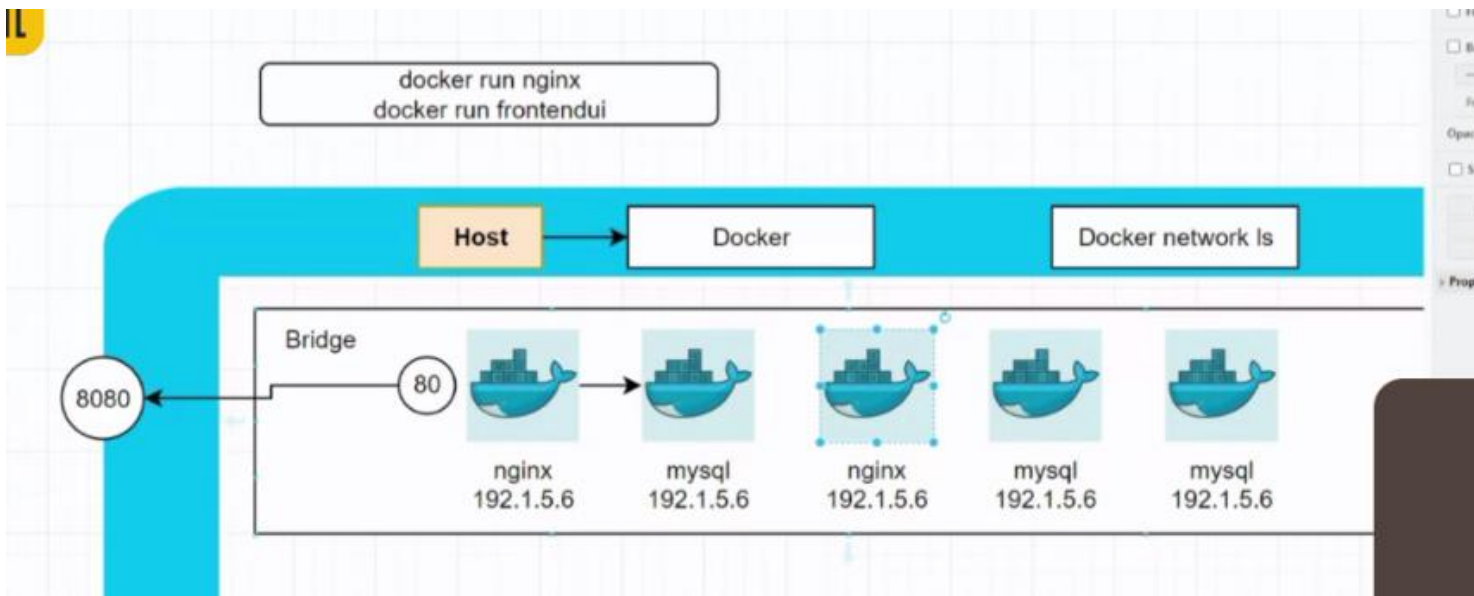
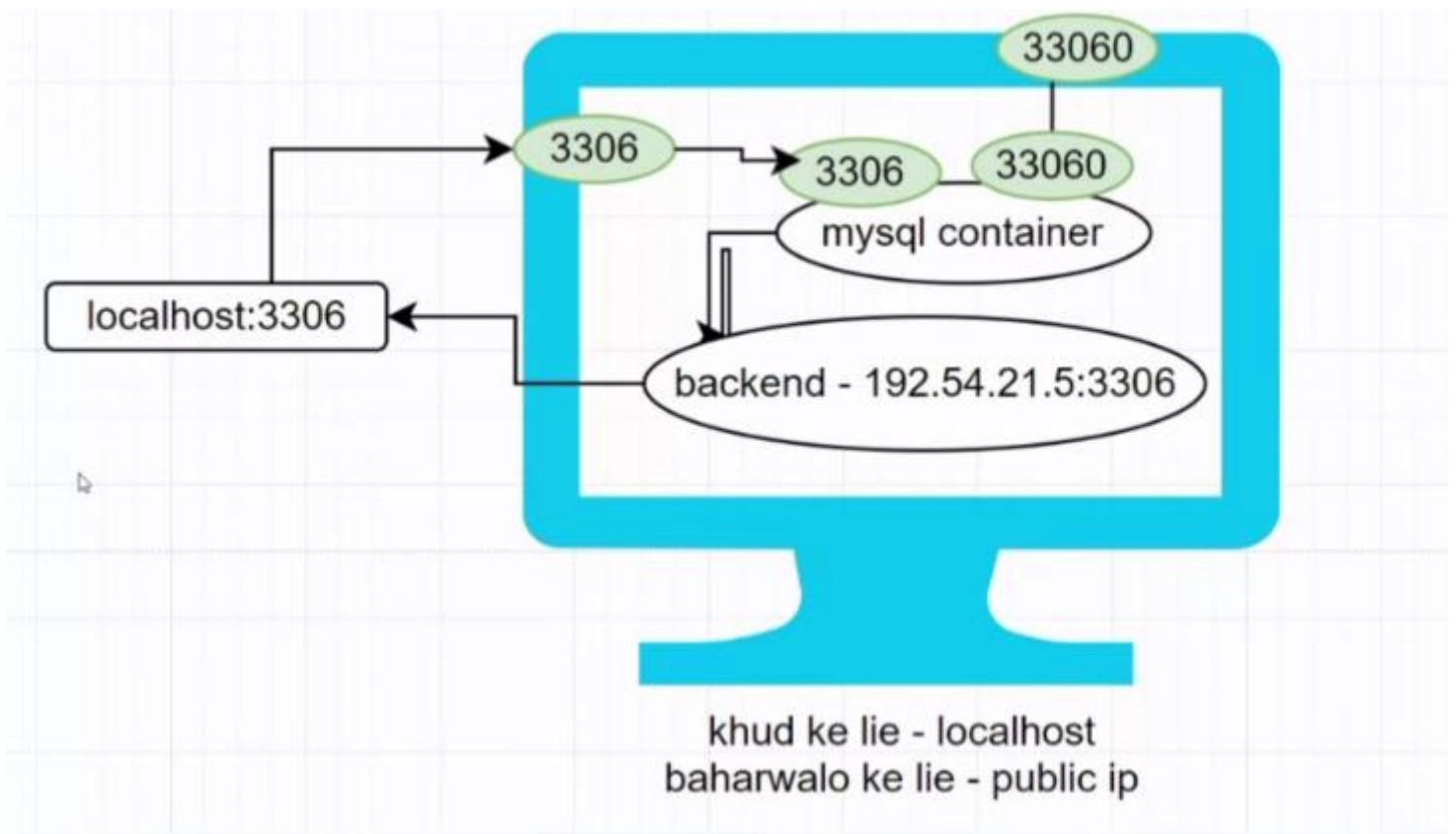
```
root@vmdocker:/home/azureuser/todobackendmysql# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
todobackendmysql	latest	c992b369a1e9	36 minutes ago	1.11GB
pytodo	latest	8badf51cfb98	17 hours ago	1.11GB
bahutpatliimage	latest	d639e855c309	46 hours ago	50.4MB
patliimage	latest	94df29409897	46 hours ago	195MB
todoui	latest	8cfbe43bc4ca	2 days ago	1.34GB
<none>	<none>	fea1e545debc	2 days ago	1.34GB
<none>	<none>	4d09465a1759	2 days ago	1.34GB
<none>	<none>	f1ebcad628d3	4 days ago	1.34GB
mcr.microsoft.com/mssql/server	2022-latest	2b41d0be8283	5 weeks ago	1.65GB
mysql	latest	567107cb6971	2 months ago	797MB

```
root@vmdocker:/home/azureuser/PyToDoBackendMonolith# docker run -d -p 8000:8000 pytodo
592598c4a12e10a3b664e7164354704cacdd64b20d2d4c3c1be2983f8ee15a97
root@vmdocker:/home/azureuser/PyToDoBackendMonolith#
```

16) Causing error





17) **docker run -d -p 8080:80 nginx** = command to run nginx container

```
root@vmdocker:/home/azureuser/PyToDoBackendMonolith# docker run -d -p 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
8a628cdd7ccc: Already exists
75b642592991: Already exists
553c8756fd66: Already exists
10fe6d2248e3: Already exists
3b6e18ae4ce6: Already exists
3dce86e3b082: Already exists
e81a6b82cf64: Already exists
Digest: sha256:09369da6b10306312cd908661320086bf87fbae1b6b0c49a1f50ba531fef2eab
Status: Downloaded newer image for nginx:latest
7362d1274426cbc551b275b85dda0a81204ef7c280ef2fc7bbc2d954829321fe
root@vmdocker:/home/azureuser/PyToDoBackendMonolith# docker ps
```

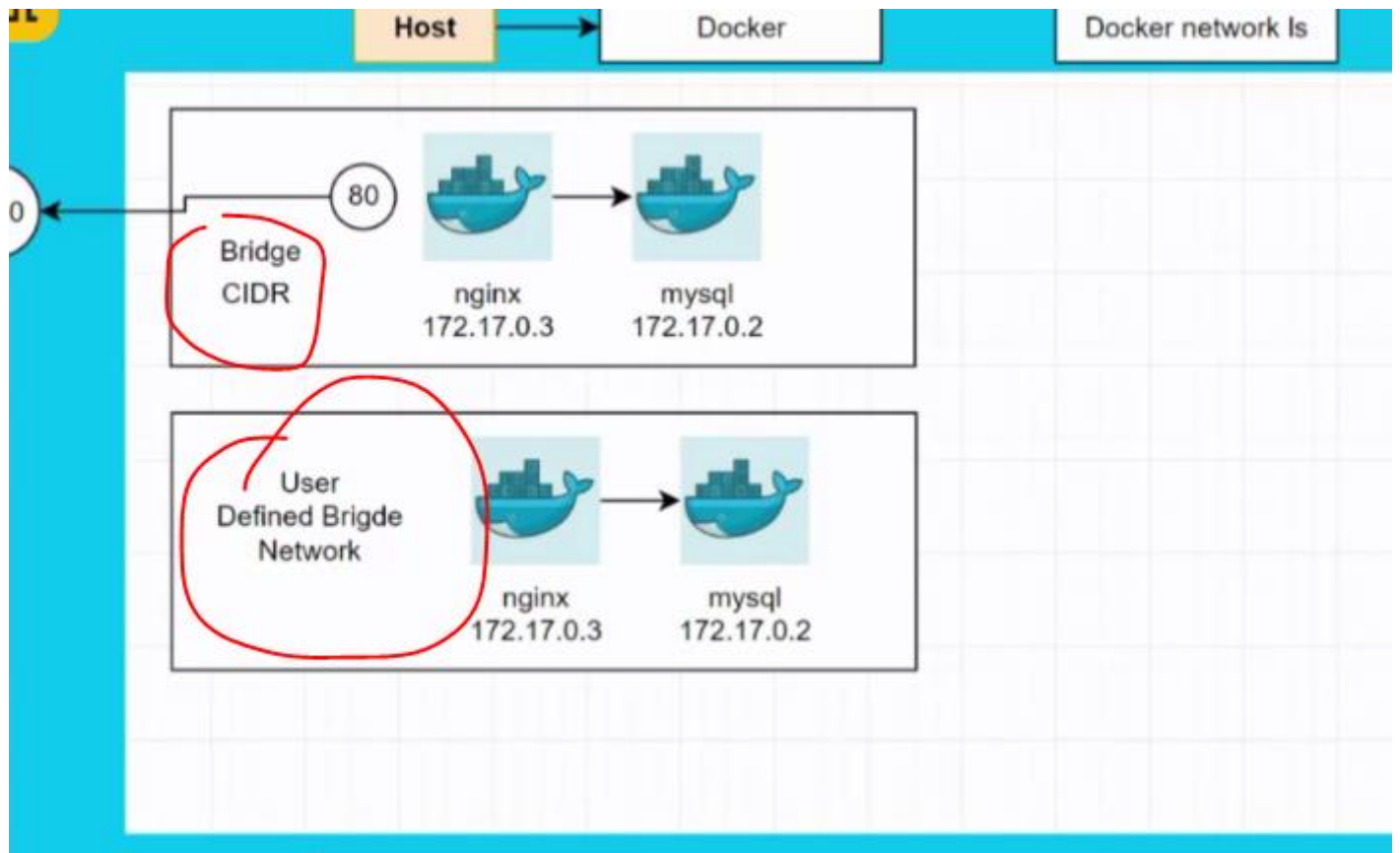
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7362d1274426	nginx	"/docker-entrypoint..."	4 seconds ago	Up 3 seconds	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	epic_dubinsky
079b0f8cba96	mysql:latest	"docker-entrypoint.s..."	4 hours ago	Up 4 hours	3306/tcp, 33060/tcp	some-mysql

```
root@vmdocker:/home/azureuser/PyToDoBackendMonolith#
```

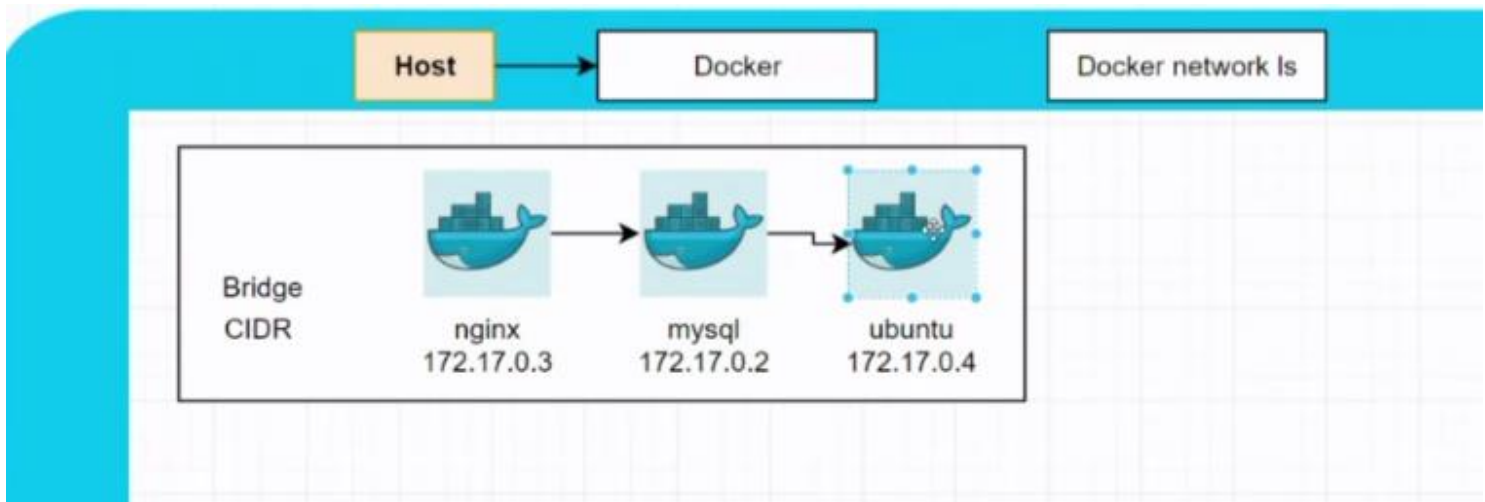
18) mysql = 172.17.0.2

nginx - 172.17.0.3

19) Network – Bridge, User defined



20) Creating Ubuntu container as well to check the mutual connectivity between mysql, nginx and Ubuntu container



docker run -it ubuntu bash = command to create and run nginx container and going inside it using bash

```
root@vmdocker:/home/azureuser/todobackendmysql# docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
2726e237d1a3: Pull complete
Digest: sha256:1e622c5f073b4f6bfad6632f2616c7f59ef256e96fe78bf6a595d1dc4376ac02
Status: Downloaded newer image for ubuntu:latest
```

ls

```
root@afd80163b0c4:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@afd80163b0c4:/#
```

Docker ps

```
root@vmdocker:/home/azureuser# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
f2b3c8d1d2ff	ubuntu	"bash"	4 minutes ago	Up 4 minutes		festive_elbakyan
7362d1274426	nginx	"/docker-entrypoint..."	2 hours ago	Up 2 hours	0.0.0.0:8080->80/tcp, [::]:8080->80/tcp	epic_dubinsky
079b0f8c8a96	mysql:latest	"docker-entrypoint.s..."	6 hours ago	Up 6 hours	3306/tcp, 33060/tcp	some-mysql

21) To run curl

apt update kro


```

root@f2b3c8d1d2ff:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1088 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [934 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [21.9 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1071 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1258 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [26.4 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1132 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1366 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [48.0 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [30.9 kB]
Fetched 29.2 MB in 5s (6459 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@f2b3c8d1d2ff:/#

```

22) apt install curl

Ip address – ubuntu - 172.17.0.4

mysql = 172.17.0.2

nginx - 172.17.0.3

23) Now giving ip address of nginx to check connectivity or access

curl 172.17.0.3

```

root@f2b3c8d1d2ff:/# curl 172.17.0.3
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>

```

24) Now we can set our backend also of our application in bridge network of docker

mysql = 172.17.0.2

cd todobackendmysql/

ls

```
root@vmdocker:/home/azureuser# cd todobackendmysql/
root@vmdocker:/home/azureuser/todobackendmysql# ls
Dockerfile  Readme.md  app.py  requirements.txt
```

nano app.py = put ip of mysql container in it

```
# Get connection details from environment
db_config = {
    'host': "172.17.0.2",
    'user': "azureuser",
    'password': "Mommy7Daddy!",
    'database': "mydatabase"
}
```

25) **docker build -t todobackendmysql .** = building image

```
root@vmdocker:/home/azureuser/todobackendmysql# docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
todobackendmysql    latest       52493efe1c07     14 seconds ago   1.11GB
```

26) Update docker file as below

nano Dockerfile

```
root@vmdocker:/home/azureuser/todobackendmysql
GNU nano 7.2 Dockerfile
# Use the official Python image as the base image
FROM python:3.9.20

# Set the working directory in the container
WORKDIR /backendapp/

# Copy the application files into the container
COPY . /backendapp/

# Install necessary packages
RUN apt-get update && apt-get install -y unixodbc unixodbc-dev
RUN curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
RUN curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.list
RUN apt-get update
RUN ACCEPT_EULA=Y apt-get install -y msodbcsql17

# Install Python dependencies
RUN pip install --upgrade pip
RUN pip install -r requirements.txt

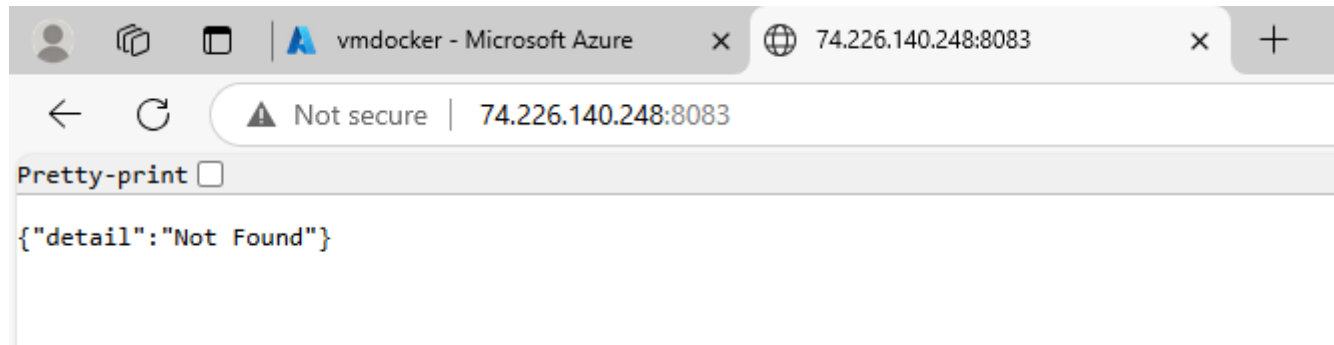
# Ensure mysql-connector-python is installed
RUN pip install mysql-connector-python

# Start the FastAPI application
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]
```

docker build -t todobackendmysql .

docker run -p 8083:8000 todobackendmysql

```
root@vmdocker:/home/azureuser/todobackendmysql# docker run -p 8083:8000 todobackendmysql
INFO:      Started server process [1]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
INFO:      Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:      106.205.85.55:53754 - "GET / HTTP/1.1" 404 Not Found
INFO:      106.205.85.55:53754 - "GET /favicon.ico HTTP/1.1" 404 Not Found
```



AGENDA – CONNECTING FRONTEND ALSO

1) **cd ReactTodoUIMonolith/**

ls

```
root@vmdocker:/home/azureuser# cd ReactTodoUIMonolith/
root@vmdocker:/home/azureuser/ReactTodoUIMonolith# ls
Dockerfile  README.md  package-lock.json  package.json  public  src
root@vmdocker:/home/azureuser/ReactTodoUIMonolith# ls -a
.  ..  .git  .gitignore  Dockerfile  README.md  package-lock.json  package.json  pub
root@vmdocker:/home/azureuser/ReactTodoUIMonolith#
```

2) **cd src**

ls

```
root@vmdocker:/home/azureuser/ReactTodoUIMonolith# cd src
root@vmdocker:/home/azureuser/ReactTodoUIMonolith/src# ls
App.css  App.js  App.test.js  TodoApp.js  index.css  index.js  logo.svg  reportWebVitals.js  setupTests.js  theme.js
root@vmdocker:/home/azureuser/ReactTodoUIMonolith/src#
```

3) **nano TodoApp.js**

root@vmdocker: /home/azureuser/ReactTodoUIMonolith/src

GNU nano 7.2

TodoApp.js

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import { Button, TextField, Container, Typography, Grid, Card, CardContent, IconButton } from '@mui/material';
import { Delete } from '@mui/icons-material';
import { Box } from '@mui/material';

const API_BASE_URL = 'http://52.146.33.135:8000';

const backgroundImage = process.env.PUBLIC_URL + '/background.jpg';

function TodoApp() {
  const [tasks, setTasks] = useState([]);
  const [newTask, setNewTask] = useState({ title: '', description: '' });

  const fetchTasks = async () => {
    try {
      const response = await axios.get(`${API_BASE_URL}/tasks`);
      setTasks(response.data);
    } catch (error) {
      console.error('Error fetching tasks', error);
    }
  };
};
```

4) In frontend, give ip address of backend

root@vmdocker: /home/azureuser/ReactTodoUIMonolith/src

GNU nano 7.2

TodoApp.js *

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import { Button, TextField, Container, Typography, Grid, Card, CardContent, IconButton } from '@mui/material';
import { Delete } from '@mui/icons-material';
import { Box } from '@mui/material';

const API_BASE_URL = 'http://172.17.0.5:8000';

const backgroundImage = process.env.PUBLIC_URL + '/background.jpg';

function TodoApp() {
  const [tasks, setTasks] = useState([]);
  const [newTask, setNewTask] = useState({ title: '', description: '' });

  const fetchTasks = async () => {
    try {
      const response = await axios.get(`${API_BASE_URL}/tasks`);
      setTasks(response.data);
    } catch (error) {
      console.error('Error fetching tasks', error);
    }
  };
};
```

5) **docker build -t reacttodoui .** = build docker image

Docker images

```
root@vmdocker:/home/azureuser/ReactTodoUIMonolith# docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
reacttodoui         latest      d80161923d1e     About a minute ago 50.4MB
todobackendmysql    latest      7f4b45a34b2d     56 minutes ago   1.25GB
```

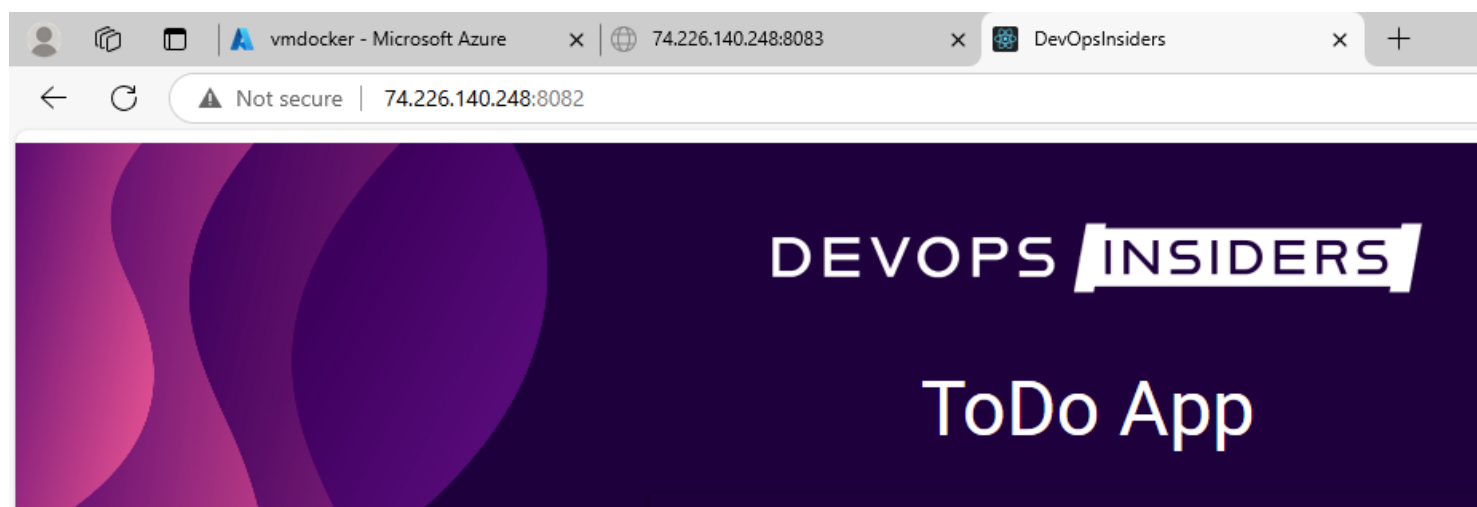
6) **docker run -p 8082:80 reacttodoui** = run container

```

root@vmdocker:/home/azureuser/ReactTodoUIMonolith# docker run -p 8082:80 reacttodoui
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/04/13 16:59:08 [notice] 1#1: using the "epoll" event method
2025/04/13 16:59:08 [notice] 1#1: nginx/1.27.4
2025/04/13 16:59:08 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
2025/04/13 16:59:08 [notice] 1#1: OS: Linux 6.11.0-1012-azure
2025/04/13 16:59:08 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1024:524288
2025/04/13 16:59:08 [notice] 1#1: start worker processes
2025/04/13 16:59:08 [notice] 1#1: start worker process 29
2025/04/13 16:59:08 [notice] 1#1: start worker process 30

```

74.226.140.248:8082



But task not added as frontend not connected to backend

7) docker ps

```

root@vmdocker:/home/azureuser# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
df3e4c0caa6a   reacttodoui   "/docker-entrypoint...." About an hour ago Up About an hour   0.0.0.0:8082->80/tcp, [::]:8082->80/tcp   quirky_galois
38b89284ff66   todobackendmysql "uvicorn app:app --h..." 2 hours ago   Up 2 hours       0.0.0.0:8083->8000/tcp, [::]:8083->8000/tcp   dazzling_cartwright
7362d1274426   nginx         "/docker-entrypoint...." 5 hours ago   Up 5 hours       0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   epic_dubinsky
079b0f8c8a96   mysql:latest  "docker-entrypoint.s..." 9 hours ago   Up 9 hours       3306/tcp, 33060/tcp                 some-mysql

```

Ip of backend - 172.17.0.4"

8) cd ReactTodoUIMonolith

docker build -t reacttodoui .

docker images

```

root@vmdocker:/home/azureuser/ReactTodoUIMonolith# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
reacttodoui   latest    b51671a6670a   About a minute ago   50.4MB
<none>        <none>    d80161923d1e   About an hour ago   50.4MB
todobackendmysql latest    7f4b45a34b2d   2 hours ago       1.25GB
<none>        <none>    52493efe1c07   3 hours ago       1.11GB
<none>        <none>    c992b369a1e9   7 hours ago       1.11GB
pytodo        latest    8badf51cfb98   24 hours ago      1.11GB

```

docker run reacttodoui = noneed to give -p as we are in the same network so no port mapping required

```

root@vmdocker:/home/azureuser/ReactToDoUIMonolith# docker run reacttodoui
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/04/13 18:21:15 [notice] 1#1: using the "epoll" event method
2025/04/13 18:21:15 [notice] 1#1: nginx/1.27.4
2025/04/13 18:21:15 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
2025/04/13 18:21:15 [notice] 1#1: OS: Linux 6.11.0-1012-azure
2025/04/13 18:21:15 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1024:524288
2025/04/13 18:21:15 [notice] 1#1: start worker processes
2025/04/13 18:21:15 [notice] 1#1: start worker process 29
2025/04/13 18:21:15 [notice] 1#1: start worker process 30

```

docker ps

```

root@vmdocker:/home/azureuser# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
db88f2843172   reacttodoui                         "/docker-entrypoint..." 4 minutes ago  Up 4 minutes  80/tcp
df3e4c0ca66   d80161925d1e                       "/docker-entrypoint..." About an hour ago Up About an hour 0.0.0.0:8082->80/tcp, [::]:8082->80/tcp
38b89284ff66   todobackendmysql                  "uvicorn app:app --h..." 2 hours ago   Up 2 hours    0.0.0.0:8083->8000/tcp, [::]:8083->8000/tcp
7362d1274426   nginx                              "/docker-entrypoint..." 5 hours ago   Up 5 hours    0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
079b0f8cba96   mysql:latest                       "docker-entrypoint.s..." 9 hours ago   Up 9 hours    3306/tcp, 33060/tcp
root@vmdocker:/home/azureuser#

```

docker inspect db88f2843172

```

    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "MacAddress": "02:42:ac:11:00:06",
    "DriverOpts": null,
    "NetworkID": "367e9c964a30ed64da50a9383971a32b855570cd2d6ce6db053735e876f1afdc",
    "EndpointID": "dc79dc92092a6b37ba35f82f3d948601c88cb64dd9abc1c245e545c24c0bebb6",
    "Gateway": "172.17.0.1",
    "IPAddress": "172.17.0.6",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "DNSNames": null
  }
}

```

9) thoda baad me