Docker Project 1

Creating a container from a pulled Image

1) first pull nginx image by using docker pull nginx command

```
root@PUNELPT0393:/home/einfochips# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
f11c1adaa26e: Pull complete
c6b156574604: Pull complete
ea5d7144c337: Pull complete
1bbcb9df2c93: Pull complete
537a6cfe3404: Pull complete
767bff2cc03e: Pull complete
adc73cb74f25: Pull complete
Digest: sha256:67682bda769fae1ccf5183192b8daf37b64cae99c6c3302650f6f8bf5f0f95df
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

2) run nginx container and verify the container is running by using docker run --name my-nginx -d -p 8080:80 nginx command

```
root@PUNELPT0393:/home/einfochips# docker run --name my-nginx -d -p 8080:80 nginx
77687d763263bc2672038e3ec4a179f7d00edefabb662768ab93a33d33e01870
root@PUNELPT0393:/home/einfochips# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
77687d763263 nginx "/docker-entrypoint..." 25 seconds ago Up 24 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp my-nginx
root@PUNELPT0393:/home/einfochips#
```

3) verify the container is running byr visiting in browser



Modifying the Container and Creating a New Image

Objective: Modify the running Nginx container to serve a custom HTML page and create a new image from this modified container.

1) Access the Running container by using docker exec -it my-nginx /bin/bash command

and create HTML page by using echo "<html><body><h1>Hello from Docker!</h1></body></html>" > /usr/share/nginx/html/index.html command

root@PUNELPT0393:/home/einfochips# docker exec -it my-nginx /bin/bash
root@77687d763263:/# echo "<html><body><h1>Hello from Docker</h1></body></html>" > /usr/share/nginx/html/index.html

2) comit the changes to create new Image by using docker commit my-nginx custom-nginx command

and run the container from new Image by using docker run --name my-custom-nginx -d -p 8081:80 custom-nginx command

```
root@77687d763263:/# exit
exit
root@PUNELPT0393:/home/einfochips# docker commit my-nginx custom-nginx
sha256:36fea635914165b114df80d69ce8712aa53a26e17a54928597a762b62e795a72
root@PUNELPT0393:/home/einfochips# docker run --name my-custom-nginx -d -p 8081:80 custom-nginx
a8379f53233f748a0243eb4adc09f3d79c1b0f82bc75472888544eb319fb188e
root@PUNELPT0393:/home/einfochips#
```

3) verify new container by visiting link in browser



Creating Docker to build and Deploy a web Application

Objective: Write a Dockerfile to create an image for a simple web application and run it as a container.

- 1) create project directory name "my-webapp" and move to that directory by using mkdir my-webapp cd my-webapp
- 2) In my-webapp create "index.html "file to write html code.

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# nano index.html
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# cat index.html
<!DOCTYPE html>
<html>
<body>
    <h1>Hello from My Web App!</h1>
</body>
</html>
```

3) create Dockerfile

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# nano Dockerfile
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# cat Dockerfile
# Use the official Nginx base image
FROM nginx:latest

# Copy the custom HTML file to the appropriate location
COPY index.html /usr/share/nginx/html/

# Expose port 80
EXPOSE 80
```

4) Build the docker Image docker build -t my-webapp-image.

and run the container from the build Image

docker run --name my-webapp-container -d -p 8082:80 my-webapp-image

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker build -t my-webapp-image .

[+] Building 0.2s (7/7) FINISHED docker:default

=> [internal] load build definition from Dockerfile 0.0s

=> => transferring dockerfile: 217B 0.0s

=> [internal] load metadata for docker.io/library/nginx:latest 0.0s

=> [internal] load .dockerignore 0.0s

=> => transferring context: 2B 0.0s

=> [internal] load build context 0.0s

=> => transferring context: 119B 0.0s

=> [1/2] FROM docker.io/library/nginx:latest 0.1s

=> [2/2] COPY index.html /usr/share/nginx/html/ 0.0s

=> exporting to image 0.0s

=> => writing image sha256:e721dd479f8c22cdeb03b8870cd3f787ccb38008bc511abeb 0.0s

=> => naming to docker.io/library/my-webapp-image 0.0s

>> => naming to docker.io/library/my-webapp-image 0.0s
```

5) verify by visiting browser using URL



Remove all reated containers and images to clean up environment

1) stop , remove containers and remove Images

docker stop my-nginx my-custom-nginx my-webapp-container

docker rm my-nginx my-custom-nginx my-webapp-container

docker rmi nginx custom-nginx my-webapp-image

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker stop my-nginx my-custom-nginx my-webapp-container my-nginx

my-custom-nginx

my-webapp-container

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker rm my-nginx my-custom-nginx my-webapp-container my-nginx

my-custom-nginx

my-webapp-container

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# docker rmi nginx custom-nginx my-webapp-image

Untagged: nginx:latest

Untagged: nginx@sha256:67682bda769fae1ccf5183192b8daf37b64cae99c6c3302650f6f8bf5f0f95df

Untagged: custom-nginx:latest

Deleted: sha256:36fea635914165b114df80d69ce8712aa53a26e17a54928597a762b62e795a72 Deleted: sha256:5c170501cbf3aba09344ccc49adeb2e35020784b340d7f72094df376ab89c3b2 Deleted: sha256:fffffc90d343cbcb01a5032edac86db5998c536cd0a366514121a45c6723765c

Untagged: my-webapp-image:latest

Deleted: sha256:e721dd479f8c22cdeb03b8870cd3f787ccb38008bc511abeb7c6d1a581973271

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/my-webapp# |

Docker Project 2

build full-stack application using docker.

Applocation consist of front-end web server (Nginx), back-end application server (node.js) and postgreSQL database.

For database and handle inter-container communication persistent volume is used

1) create project directory name "fullstack-docker-app"

mkdir fullstack-docker-app cd fullstack-docker-app

2) In Directory create subdirectories name "frontend, backend, database" for each service

mkdir frontend backend database

3) create network to allow communication between containers docker network create fullstack-network

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july# mkdir fullstack-docker-app root@PUNELPT0393:/home/einfochips/Desktop/learning/10july# cd fullstack-docker-app/ root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# mkdir frontend backend database root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# ls backend database frontend root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker network create fullstack-network 6323847df045d03efe778495b0177cd834c62594e6f2aae25c4937b5dbce5b0b

4) create volume for PostgreSQL database docker volume create pgdata

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker volume create pgdata pgdata

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app#

set up database with docker

1) In "database" directory create dockerfile for database

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/database# nano Dockerfile root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/database# cat Dockerfile FROM postgres:latest ENV POSTGRES USER=user ENV POSTGRES_PASSWORD=password ENV POSTGRES DB=mydatabase

2) build postgreSQL Image

docker build -t my-postgres-db.

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/database# docker build -t my-postgres-db
[+] Building 105.3s (3/4)
ault
 => [internal] load build definition from Dockerfile
0.0s
0.0s
 => [1/1] FROM docker.io/library/postgres:latest@sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0
2.1s
 \Rightarrow resolve docker.io/library/postgres:latest@sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0
0.0s
 => => sha256:6b7a1245fe7148b56315cd314fe4764418e6e4ec1095b4e781e3a60ca6990e64 1.45MB / 1.45MB
 => => sha256:0aafd2ae7e6c391f39fb6b7621632d79f54068faebc726caf469e87bd1d301c0 10.26kB / 10.26kB
0.0s
 => => sha256:919ca406a058a0c62788518f39a66257c122ace2007ac6e1425e51846f37e300 4.53MB / 4.53MB
 => => sha256:d13ef786196545cd69aff1945929fc868712196e195bc66581fb1bfe81649eaf 3.63kB / 3.63kB
 => => sha256:76ce212b9153cf6743484a31c2967fd5f284455f8ede2277943f9d92010ffa9e 1.17kB / 1.17kB
 => => sha256:f23dc7cd74bd7693fc164fd829b9a7fa1edf8eaaed488c117312aef2a48cafaa 10.09kB / 10.09kB
 => extracting sha256:76ce212b9153cf6743484a31c2967fd5f284455f8ede2277943f9d92010ffa9e
 => => sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3438077f3409a3e5 8.07MB / 8.07MB
```

Internal Load notatata for docker. Log Losternal Losternal Log	[+] Building 349.7s (5/5) FINISHED => [internal] load build definition from Dockerfile	docker:default
Content Doub docker (propre		
Description		
File		
>> resolve focker. Log (througy)postpress Latest (glasa266-dae/face/e00917976-dae/face/e00917976-dae/face/e009179776-dae/face/e009179776-dae/face/e00917976-dae/face/		
>> shab56/db712/5f7-fl48b5215/cd15f4r64186e4c1295546781828062699645 1.4598		
>>> sha256;0aafdze1ec91f39fb6h721637c09f54068faebc72ccaf499e87bdd9t1c0 10.26la / 18.26l8		
Separate		
⇒⇒⇒⇒⇒±a55c113e7780196495cd69aff1949929f1688711196e1956bc666581fb1bfe81648e8f 3.6388 8.835 ⇒⇒⇒⇒±a526c76c212b9153cf6743484a31c2957f465728445578b2018f18e9 1.1788 1.1788 ⇒⇒⇒⇒±a526c76c7d76dr593f164572844576d78e8ed2827794f39b2018f18e9 8.85 ⇒⇒ extracting sha256c76c212b9133cf6743484a31c2957f46f28e4455f8ede227794f39d2019ffe8e 8.08 ⇒⇒⇒±a526c665509782bc1393736993c0f152g7425c0f1548h20297f72cf2cf343907f139d928e8f8e9 5.9555 ⇒⇒⇒±a526c65597882bc189395844c10e1be35a2c96f15498h2024749f16496f101 1.2089 2.01 ⇒⇒⇒±a526c6597882bc1895844bc18eb35a2c96f15498h2024749f16496f101 1.2089 2.07.55 ⇒⇒⇒±a526c6f82c6f489c186c478585bc2058252c3af563109c26f58e82bc2071 1.808 2.1.58 ⇒⇒⇒±a526c6f82c6f489c4912asac28be407c478c9533c5ebf5097a259f627f649f513 1.4188 3.1488 2.1.15 ⇒⇒⇒±a526c6f82c6f489c49f249412asac28be407a258c5ebf5097a259f627fc49be5097a259f627fc49b667a209fa26c4445e5f369f2099a26c4 0.09 3.09 ⇒⇒⇒±a526c6f82c6f489c49f349f3ac28be607c78b518f36f36be507c7c42ace09fa26c44425e5f369c49f7e7b96 0.25 0.25 ⇒⇒⇒±a526c6f3ace0eb0f97ac6f3ce0effe7f44b55515c3df46f444b8be6efe2ef095b4e7f18eb6667a59f46f7e0b6666 0.05 0.05 ⇒⇒±a526cc9bce6efe8effe7f4b6bc6f5b7a26b667af44b66a66e9f87b6a6efe8efe8efe8efe8efe8efe8efe8efe8efe8efe		
⇒⇒⇒⇒ sha25s:F23dc7cd74bd7693fc164f0829b3a7fa1edf8eaaed488c117312aef2a48cafaa 18.09k8 / 19.09k8 6.8s ⇒⇒⇒ sha25s:67c6c212b1515Cf47449643bc25b7f62f24445f8ede2277449f094016f169e 6.8s ⇒⇒⇒ sha25s:666f7e60c6543978a01373fc49bb1c2957ff2c2f3c343ba97774499a5e 8, 67n8 / 8.07n8 59.555 ⇒⇒ sha25s:6727664970455799993c66f7237312acf8c2daaed89898dd1ccff9c184f59 1168 / 1.20m8 20.1s ⇒⇒ sha25s:672776499704557999993c66f7237312acf8c2daaed898898dd1ccff9c184f59 1168 / 1.20m8 20.7s5 ⇒⇒ sha25s:672776499704557999993c66f72372312caf8c2daaed898898dd1ccff9c184f59 1168 / 1.80m 20.7s5 ⇒⇒ sha25s:6727649704557999993c66f72372472f609332cedbf59077a2956f8e27f0e451 3.14k8 / 3.14k8 21.1s ⇒⇒ sha25s:6786707f16197618696ce0e402648767e2652625af6631006df8e988260270 19.08.09 / 19.0989 30.9s9 ⇒⇒ sha25s:6786267e39f9c4466a988a0c62788518f39a66257c122acc2807ac6e4425e3846f37e390 0.2s ⇒⇒ sha25s:6786267e39f9c446a98a8666587157246f47f31a30f2808605021 9.20k8 / 9.90k8 30.6s5 ⇒⇒ sha25s:678667667e36f573736f57af646474f47f31a30f2806865905021 9.20k8 / 9.90k8 30.6s5 ⇒⇒ sha25s:678667667e36f5744606576567d44f47f31a30f28066050219.20k8 / 9.90k8 51.2s ⇒⇒ sha25s:678666589994f649666585676766586c7444008426764664898238886466666666666666666666666666666666		
⇒⇒ extracting sha256:76ce212b013x16f743484a31c29x7fdx78485f8ede2277943f9092e10ff39e 0.8s ⇒⇒ sha256:8064f5e0c5d3072e01373f7d39b16c3229x7ff2ccf5c43907f744093ae5 8.0708 [8.0708 59.555 ⇒⇒ sha256:405c59f2082c085959498a41c01eb32c3c6f15d8b2c2457f419ff0406161 1.2008 [1.2008] 20.1s ⇒⇒ sha256:fef0c26f40492f92a8cc08be407e2fe00333cc6df5807a295bfe77f0e451 3.14k8 [3.14k8] 21.1s ⇒⇒ sha256:fef0c26f40424f92a8cc08be407e2fe0033cc6edf5807a295bfe77f0e451 3.14k8 [3.14k8] 21.1s ⇒⇒ sha256:fef0c26f40424f28c0c25c263af565100ca6f05808bc027b7 10p.308 [10p.308] 340.2s5 ⇒⇒ extracting sha256:0b7a125f7f14b6563f6cd7ac295bfe77f0e451 3.14k8 [9.14k8] 10p.308] ⇒⇒ extracting sha256:0b7a125f7f14b65315cd31ffc4ff48c4418e6eec(1095bfe701e3a6c3690e64] 0.8s ⇒⇒ sha256:1d3e06bf6760bf67537f0bf676bf767bf676bf767bf67bf67bf67bf67bf67	=> => sha256:76ce212b9153cf6743484a31c2967fd5f284455f8ede2277943f9d92010ffa9e 1.17kB / 1.17kB	0.4s
⇒⇒⇒⇒sha256:8064ffe86c65d307820b1373f7d39b16c32929f7f2c2cf3c343807f73409a3e5 8.67N8 8.67N8 59.5SS ⇒⇒⇒sha256:tb5c59f2d8cd365959458a1cd1ebeb35a2c96f1c45ebd2e2x57419cff4066161 1.20W8 1.20W8 20.1s ⇒⇒⇒sha256:te72f7c4b90704557399993c60f23f213caf8cd2adea88988d41ccff9c184f59 1168 1168 21.1s ⇒⇒⇒sha256:te78fc20f4d9249128abc30be4b72cf7c409333csdbf5907a295f8cf7f6eb51 3.14k8 3.14k8 21.1s ⇒⇒⇒sha256:te71fc9d7ff11d95e9c7e54d02048f20e5252263af563100c8df6508820b2707 109.03Wb 109.03Wb 340.2s5 ⇒⇒⇒⇒xtractting sha256:5919ca406c98abc2788518f39a66257c122xe20907ac6c41285e5184f3r2080 0.2s ⇒⇒⇒xtractting sha256:5919ca406c98abc2788518f39a66257c122xe20907ac6c41285e5184f3r2080 0.2s ⇒⇒⇒xtractting sha256:f910c406f4f4r3026cdaebf6e30d6c0404f51df4f7f33a307c280e589624 9.92k8 9.92k8 ⇒⇒⇒xtractting sha256:f910c40f3r245fc7148b53315cd31df6r4ff4f33a307c280e5896924 1.78B 51.6s ⇒⇒ sha256:f23aee4beb07954f6b1c59f7ca9998c5cd33d6x17ab393f44233c9red16312342 1788 1.78B 51.6s ⇒⇒ sha256:f23c26f9509f66e60997c98c33d6x17ab393f44233c9red16312342 1788 1.78B 51.6s ⇒⇒ sha256:f23c26f9509f66e80997c98c33d6x16e7b393f44233c9red16312342 1788 1.78B 51.6s ⇒⇒ sha256:f23c26f9509f66e80997c98c33d6x16e7b395f44233c9red16512344 5.42k8 5.42k8 ⇒	=> => sha256:f23dc7cd74bd7693fc164fd829b9a7fa1edf8eaaed488c117312aef2a48cafaa 10.09kB / 10.09kB	0.0s
⇒⇒ sha256:4b5c9f2d82cd85059458a4Lc01ebe3b5a2c96f1d594b2e24574149ff6406161 1.20MB / 1.20MB 20.1s ⇒⇒ sha256:6f68e2c0f40924192a8ac28be44b27E17ed0933C5eb0f5007a1959f8e7ff0e451 3.14kB / 3.14kB 21.1s ⇒⇒ sha256:6f8e8c20f40924192a8ac28be477e17ed0933C5eb0f5007a195b78e7ff0e451 3.14kB / 3.14kB 21.1s ⇒⇒ sha256:6f8e8c20f40924192a8ac28be477e2b62c263af5c3160eddf695888p0c707 109.03MB / 109.03MB 340.2s5 ⇒⇒ extractting sha256:919ca406a058a0c678818f39a6c157c122ace2007ac6e1425e51846f37e300 0.2s ⇒⇒ sha256:152256f919dd6f447a026cdaabf6e306c0409f516f47f33a39f78e80956b971 9.92kB / 9.92kB 50.655 ⇒⇒ sha256:15256f919ddf4f47a026cdaabf6e306c0409f5447f3a3a30f28e8056b064 0.8s ⇒⇒ sha256:15256d917afc6bbcc96e10e7lbb80c7b3f82c46c388a80e4b2ef7347h0c0bc 128B / 128B 51.2s ⇒⇒ sha256:1526cd3aee49eb07954f0cb1c957ca0990e5c635b0617ab593f4c433ac97ed16312342 170B / 170B 51.6s ⇒⇒ sha256:1285cd3a99194f439ba266b753710f59da6s3be9715774c6249e3823ae8711c7 5.42kB / 5.44kB 52.1s ⇒⇒ sha256:1285cd2aef9f0c2df66e80992c90c53c9ec7b59eccf24d00abf52820eaf 184B / 184B 52.4s ⇒⇒ extracting sha256:4b5c59f2d82cd855945bbc1c424d00a5c820ea 184B / 184B 52.4s ⇒⇒ extracting sha256:6f68e2c054942412aaca2beabf2aef24d64aef8be7aef8b07aef8b07aef8b07aef9b07aef6b0665493419aaaca2beabf2aef3c26af3aef3aef8b07a	=> => extracting sha256:76ce212b9153cf6743484a31c2967fd5f284455f8ede2277943f9d92010ffa9e	0.0s
⇒ ⇒ sha256:fe727649907045579e9993c60f23f3213caf82d2ade388988d41ccff9c184f59 116B / 116B 20.7s5 ⇒ ⇒ sha256:ef82cdf40241238ac20betb72e17e0933c5cbdr5907a29516e77fee451 3.14kB / 3.14kB 21.1s ⇒ ⇒ sha256:ef192cdf40241238ac20betb72e17e0933c5cbdr5907a29516e77fee451 3.14kB / 3.14kB 340.2s5 ⇒ ⇒ sha256:fe91f1e90ff111095e9c7e54002048f20e5c26263af563100c8df698080b20737 109.00NB / 109.03NB 340.2s5 ⇒ ⇒ sha256:f3225d69190dd8f447a026cdaebfe03d662de5bf53f6567c122ace2087ac6e1425e5184f37e300 0.2s ⇒ ⇒ sha256:f3225d69190dd8f447a026cdaebfe03d662de5bf51df4f7f3a3a9f2088056b921 9.92kB / 9.92kB 50.6s5 ⇒ ⇒ sha256:f3225d69190dd8f447a026cdaebfe03d662de5bf51df4f6451e6e426e7347b02be 128B / 128B 51.2s ⇒ ⇒ sha256:f2bf90d17afc8b5ec96e1ber1ib080c7b53f626bc6388080b320ef7347b02be 128B / 128B 51.2s ⇒ ⇒ sha256:f2bf90d17afc8b5ec96e1ber1ib080c7b53f62d6c63880a96b21737f2d62d9a88230a8211c7 5.4kB / 5.4kB 51.5s ⇒ ⇒ sha256:e1e8566589194f439ba266b753f10f59da63de9173f7dc94a88230a8211c7 5.4kB / 5.4kB 52.1s ⇒ ⇒ sha256:e1e8566589194f4666e80992c98c33cec7b5e8ccf24d908526b54bc287da9a8823a88d1ccff9c144966161 0.1s ⇒ ⇒ extracting sha256:f3bc16bf5972d2acd8b593945a41c01ebe3b5acc96f1d584b2e447f449ff406161 0.1s ⇒ ⇒ extracting sha256:f3bc16f2bf0f46924f1923aacdbe4b72ef1ed6935cedbf5907a295f6e7f6e451 0.0s ⇒ ⇒ extracting sha256:f3bc2f3bc3bc4bf2ef1f47f33a387c8088985b921	=> => sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3438077f3409a3e5 8.07MB / 8.07MB	59.5s5
⇒ ⇒ sha256:6ef8e2c6f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e77f0e451 3.14kB / 3.14kB 21.1s ⇒ ⇒ sha256:re1f6effff1d95e9c7e54d020428720e5262253a7553100c3df6888820b2701 109.03WB / 109.03WB 340.2s5 ⇒ ⇒ extracting sha256:919ca406a6808a0c6278815l8739a66275t122ace2007ac661425e184673re300 0.2s ⇒ ⇒ sha256:f3226d59190c406a658a0c6278815l8f39a66257t122ace2007ac661425e1846f37e300 50.6s5 ⇒ ⇒ extracting sha256:6b7a1245fe7148b56315cd314fe4764418e6e4ec1095b4e781e3a66ca6999e64 0.0s ⇒ ⇒ sha256:cbF90d17afc8b5ec96e10e71bb08c75b3f82c46c388a804b3e9de73437b02be 128B / 128B 51.2s ⇒ ⇒ sha256:d3aec49eb07954f0cb1c9975ca69998e5c635de6f3ab593f4c433c97ed651214c1 170B / 170B 51.6s ⇒ ⇒ sha256:d3aec49eb07954f0cb1c9975ca6999e5c635c9ec7b5e8ccf244d08a5c85e46bbc2828ae 184B / 154B 52.4s ⇒ ⇒ sha256:e18266659194f39ba266b753710f59da6a5de997147ec6249a88230a8211c7 5 .42kB / 5 .42kB 52.4s ⇒ ⇒ extracting sha256:e962d84f6e06c5d3d782b313f3f7d39b16c32929f7f2c2cf3c343807f3409a3e5 0.9s ⇒ ⇒ extracting sha256:e962d84f6e06c5d3d782b353f3af1c61be0b5a2c9ef1d594k2c45f7449f1640e6161 0.1s ⇒ ⇒ extracting sha256:efe8c26f4d924192aBac2bbed7b72af1cf6d93a5c2edbf580f2a89bf76e7f0e16f96 0.8s ⇒ ⇒ extracting sha256:efe8c26f4d924192aBac2bbed7b72af2e6f4c38a8988db3e9de73437b02be 0.0s ⇒ ⇒ extracting sha256:efe8c26f4d924192aBac2bbed7573d06c18de6f593f44664938230a89211c7 0.0s	=> => sha256:4b5c59f2d82cd85059458a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 1.20MB / 1.20MB	20.1s
⇒ ⇒ sha256:e71Fe9d7ff11d95e9c7e54d020428f20e5262263af563100c0df6508820b2707 109.03WB / 109.03WB 340.255 ⇒ ⇒ extracting sha256:919ca4666383a0cC278851Bf39a66237c122ace2007ac6e1425e51346f37e30B 0.2s ⇒ ⇒ sha256:f3225d09190ddBf447a026cdaebfe03d062d04bf51df47f33a30f20880569919 9.92kB / 9.92kB 50.655 ⇒ ⇒ sha256:f3225d69199ddBf447605bcc96c10e71bb867315d34fc4764418e6e4c1095b4e781e3a60ca6990e64 0.8s ⇒ ⇒ sha256:d3aee49eb07954f0cb1c957ca8999e5c635d0617ab593f42433c97ed16312342 170B / 170B 51.6s ⇒ ⇒ sha256:d3aee49eb07954f0cb1c957ca8999e5c635d0617ab593f42433c97ed16312342 170B / 170B 51.6s ⇒ ⇒ sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a80230a8211c7 5.42kB / 5.42kB 52.1s ⇒ ⇒ sha256:95e2c2ce9f762d7666e89992c9c3c3c3ecef7bse8ccf244d08a5c38e46bbc2828ae 184B / 184B 52.4s ⇒ ⇒ extracting sha255:8064ffe06c5d30782b1373f7d39b16c32295f7f2c2cf3c343807f34093a5e5 0.9s ⇒ ⇒ extracting sha256:f62f3d82c08f999599458a41c01eeb3ba2c9ff1d584b2c4574149ff6406161 0.1s ⇒ ⇒ extracting sha256:f692c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e7f7f0e451 0.0s ⇒ ⇒ extracting sha256:e1990d17afc8b5ec96e10e71bb08c75bf36264c38888808b5e9de734737b02be 0.0s ⇒ ⇒ extracting sha256:e1990d17afc8b5ec96e10e71bb08c75bg8ec635d0e71b5997f24453c9fed65312242 0.0s ⇒ ⇒ extracting sha256:e1990d17afc8b5ec96e10e71bb08c75bf8e6c63d0e97157442453c9fed6532242	=> => sha256:fe72764b907045579a9993c60f23f3213caf82d2ade388988d41ccff9c184f59 116B / 116B	20.7s5
⇒ ⇒ extracting sha256:919ca406a058a0cc0788518f39a66257c122ace2007ac6e1425e51846f37e300 0.2s ⇒ ⇒ sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 9.92k8 / 9.92k8 50.6s5 ⇒ ⇒ extracting sha256:fb124f5f2748b55315cd31df4a76418c64ec1095b4e78le3a66ca9990ec4 0.0s ⇒ ⇒ sha256:2bf90d17afc0b5ec96e10e71b08c75b5f8264c388a0884b3e9de73437b02be 128B / 128B 51.2s ⇒ ⇒ sha256:2bf90d17afc0b5ec96e10e71b08c75b5f8264c38a8a084b3e9de73437b02be 128B / 170B 51.6s ⇒ ⇒ sha256:95c2cef9f02d7666e80992c98c53c9ec75b5e8ccf244d00a5c85e46bbc282dae 134B / 184B 52.1s ⇒ ⇒ sha256:95c2c2ef9f02d7666e80992c98c3c9ec7b5e8ccf244d00a5c85e46bbc282dae 134B / 184B 52.4s ⇒ ⇒ extracting sha256:66559f2d82cd8565948a41c81be3b5a2c96f1d584b2e24574149ff0466161 0.1s ⇒ ⇒ extracting sha256:fe72f4d90f94573pa9993c66f23f3213caf82d2ade38989804Lcff9c45f3f4 0.0s ⇒ ⇒ extracting sha256:fe71fe9d7ff11d95e0f2e54d082d2e3e26e26a3af563100c8df6508820b2707 5.7s ⇒ ⇒ extracting sha256:e71fe9d7ff11d95e0f2e54d02a82d2e3e262c3af563100c8df6508820b2707 5.7s ⇒ ⇒ extracting sha256:e71fe9d7ff11d95e0f2e54d02e3d2e3e5262c3af563100c8df6508820b2707 0.0s ⇒ ⇒ extracting sha256:e71fe9d7ff11d95e0f2e54d062a2d2e3e526a3af563100c8df6508820b2707 0.0s ⇒ ⇒ extracting sha256:e18566589194f43yaba266b753710f53da65de9715724c6249a8823a8211c7 0.0s ⇒ extracting sha256:e128566589194f43yaba266b7537	=> => sha256:6ef8e2c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e77f0e451 3.14kB / 3.14kB	21.1s
⇒ ⇒ sha256:f3225d69190dd8f447a026cdaebfe03d662d04bf51df47f33a30f2080856b921 9.92kB / 9.92kB 50.655 ⇒ ⇒ extracting sha256:6b7a1245fe7148b56315cd314fe4764418e6e4ec1095b4e781e3a60ca6990e64 0.05 ⇒ ⇒ sha256:2bf90d17afc8b5ec96e10e71b080c75b378264c38838084b2e9dc73437b02be 128B / 128B 51.25 ⇒ ⇒ sha256:2bf90d17afc8b5ec96e10e7b0b0c75c3979e95e5c553b0617ab593f2433c97cd16312342 170B / 170B 51.65 ⇒ ⇒ sha256:9b2c2cf9f0d27666889992c98c553c9ec7b58eccf244d08a5c85e4dbbc2820ae 184B / 184B 52.45 ⇒ ⇒ sha256:9b2c2cf9f0d27666889992c98c55c9ec7b58eccf244d08a5c85e4dbbc2820ae 184B / 184B 52.45 ⇒ ⇒ extracting sha256:9b2c2cf9f0d27666889992c98c55c9ec7b58eccf24d408a5c85e4dbbc2820ae 184B / 184B 52.45 ⇒ ⇒ extracting sha256:9b2f2d2cd85059458a41c01ebe3b5a2c96f1d58db2e24574149ff0406161 0.1s ⇒ ⇒ extracting sha256:fe727649907045799993c60f23f21213ccf82d2ade388988d41ccff9c184f59 0.0s ⇒ ⇒ extracting sha256:re18e20f40g24192aaceb850e3c226af5037a05256db760760880802b02707 5.7s ⇒ ⇒ extracting sha256:re18e20f40g24192aacebebd72e17ed0933c5edbf5080760588820b2707 5.7s ⇒ ⇒ extracting sha256:re18e20f40g24192aacebebd72e17e3a98625dc26a765308060f058880808020b2707 0.0s ⇒ ⇒ extracting sha256:re18e366589194f4439ba266db553f264dc388a8084b3e9de73437b02be 0.0s ⇒ ⇒ extracting sha256:re18e366589194f439ba266db573710f59da63de9715774c6249a802308211c7 0.0s ⇒ ⇒ extracting sha2	=> => sha256:e71fe9d7ff11d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 109.03MB / 109.03MB	340.2s5
⇒ ⇒ extracting sha256:6b7a1245fe7148b56315cd314fe4764418e6e4ec1095b4e781e3a60ca6990e64 0.0s ⇒ ⇒ sha256:2bf90d17afc8b5ec96e10e71bb08c75b5f82646c388a8084b3e9de73437b02be 1288 / 1288 51.2s ⇒ ⇒ sha256:d3aee49eb67954f0cb1c95rca6998e5c3350d617ab593f42433c97ed16312342 1708 / 1708 51.6s ⇒ ⇒ sha256:e1e85656389194f439ba266b753710f59da6a5de9715724c6249a80230a8211c7 5.42k8 / 5.42k8 52.1s ⇒ ⇒ sha256:955c2c2ef9f0cd7666e80992c98c53c9ec75be8cc7244d00a5c85e46bbc282dae 1848 / 1848 1848 ⇒ ⇒ extracting sha256:8b64ffe06c65d30782db1373f7d39b16c32929f7f2c2cf3c343807f3409a3e5 0.9s ⇒ ⇒ extracting sha256:4b5c59f2d82cd88595948a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 0.1s ⇒ ⇒ extracting sha256:6fe82c6f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e7f70e451 0.0s ⇒ ⇒ extracting sha256:ef98c64d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e7f70e451 0.0s ⇒ ⇒ extracting sha256:ef98c64d924492a8c20abc262d263af563100c8df05008020b2707 5.7s ⇒ ⇒ extracting sha256:1cf80f064d64f47a026cdaebfe03d062d04bf51df4f7f3a30f2080856b921 0.0s ⇒ ⇒ extracting sha256:1cf80f0647bc1cf97ca0998c5c635d0617ab593f42433.C9Pcd16312342 0.0s ⇒ ⇒ extracting sha256:1cf80f067ab6e80992c98c53.c9ec7b5e8ccf244d00a5c85e46bbc2820ae 0.0s ⇒ extracting sha256:1cf80f66889194f439ba266b733710f59da6a5de9715724c6c49a88230a8211c7 0.0s ⇒ extra		
⇒ > sha256:2bF90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3e9de73437b02be 128B 51.2s ⇒ > sha256:d3aee49eb87954f0cb1c97ca0998e5c635d0617ab593f42433c97ed16312342 170B 51.6s ⇒ > sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 5.42kB 5.42kB ⇒ > sha256:95c2c2ef9f02d7666e8992c98c55c9ec7b5e8ccf244d0a5c8sb46bbc2820ae 184B 194B ⇒ > extracting sha256:8b65695d86ffe0c65d30782b01373f7d39b16c922929f762c2cf3c343807f7409a3e5 0.9s ⇒ > extracting sha256:4b5c59f2d82cd8b5059458a41c01be3b5a2c96f1d584b2e24574149ff0406161 0.1s ⇒ ⇒ extracting sha256:fe72764b907045579a9993c66723f3213caf82d2ade388988d41ccff9c184f59 0.0s ⇒ ⇒ extracting sha256:fe72f64b97045579a9993c660f23f3213caf82d2ade388988d41ccff9c184f59 0.0s ⇒ ⇒ extracting sha256:e1ge2cdf4d924192a8ac28be4b7ze17ed9335c8ebf5007a295bf8e77f0e451 0.0s ⇒ ⇒ extracting sha256:e7jze3d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 0.0s ⇒ ⇒ extracting sha256:19b60d7afc6be8092c7e54d020428f20e5262263af563100c8d60508820b2707 5.7s ⇒ ⇒ extracting sha256:19b60d7afc6be80992c98c53c9ec7b5e8c63d6d7ab593f42433c97ed16312342 0.0s ⇒ ⇒ extracting sha256:19b60d7afc6be80992c98c53c9ec7b5e8c6724d08a5285e46bbc2820ae 0.0s ⇒ extracting sha256:19b60d7afc6be80992c98c53c9ec7b5e8ccf24dd08a5c85e46bbc2820ae		
⇒ > sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 170B / 170B 51.6s ⇒ > sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 5.42kB / 5.42kB 52.1s ⇒ > sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 184B / 184B 52.4s ⇒ > extracting sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3438077f3409a3e5 0.9s ⇒ > extracting sha256:4b5c59f2d82cd85059458a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 0.1s ⇒ > extracting sha256:fe727e4b997045579a9993c60f23f3213caf82d2ade38898041ccff9c184f59 0.0s ⇒ > extracting sha256:fe726d74924192a8ac20be407ze17ed0933c5edbf50807a295bf8e77f0e451 0.0s ⇒ > extracting sha256:e7fe9d0ff1f1d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 5.7s ⇒ > extracting sha256:93225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f20e808650921 0.0s ⇒ > extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b38f82646c388a8084b3e9de73437b02be 0.0s ⇒ > extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a80230a8211c7 0.0s ⇒ > extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a80230a8211c7 0.0s ⇒ > exporting to image 0.0s ⇒ > exporting to image 0.0s ⇒ > mriting image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf40406570b920262d21e57ca2275 0.0s		
>> sha256:e1e8566599194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 5.42k8 / 5.42k8 5.42k8 / 5.42k8 >> sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 1848 / 1848 52.4s >> extracting sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3438077f3409a3e5 0.9s >> extracting sha256:4b5c59f2d82cd85059458a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 0.1s >> extracting sha256:fe72764b907045579a9993c60f23f3213caf82d2ade388988d41ccff9c18df59 0.0s >> extracting sha256:fef8e2c0f4d924192a8ac20be4b7Ze17ed0933c5edbf5007a295bf8e77f0e451 0.0s >> extracting sha256:f3225d69190dd8f447a026cdaebfe030d62d04bf51df47f33a30f2088056b921 0.0s >> extracting sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 0.0s >> extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 0.0s >> extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 0.0s >> exporting to image 0.1s >> exporting layers 0.1s >> mining image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 0.0s >> naming to docker.io/library/my-postgres-db 0.0s		
>> sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 184B / 184B 52.4s >> extracting sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3438077f3409a3e5 0.9s >> extracting sha256:4b5c59f2d82cd85059458a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 0.1s >> extracting sha256:fe72764b907045579a9993c60f23f3213caf82d2ade388988d41ccff9c184f59 0.0s >> extracting sha256:6ef8e2c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e77f0e451 0.0s >> extracting sha256:71fe9d7ff11d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 5.7s >> extracting sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 0.0s >> extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b5f82646c388a8084b3e9de73437b02be 0.0s >> extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 0.0s >> extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a80230a8211c7 0.0s >> exporting to image 0.1s >> exporting layers 0.0s >> mining image sha256:1cfe6ff872zf48bf2889ab46a7z7e0e5ed0bf4046b570b920262d21e57ca2275 0.0s >> naming to docker.io/library/my-postgres-db 0.0s		
⇒ > extracting sha256:8064ffe06c65d307820b1373f7d39b16c32929f7f2c2cf3c3d38077f3409a3e5 0.9s ⇒ > extracting sha256:4b5c59f2d82cd85059458a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 0.1s ⇒ > extracting sha256:fe72764b907045579a9993c60f23f3213caf82d2ade388988d41ccff9c184f59 0.0s ⇒ > extracting sha256:6ef8e2c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e7f76e451 0.0s ⇒ > extracting sha256:e71fe9d7ff11d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 5.7s ⇒ > extracting sha256:f3225d69190dd8f447a026cdaebfe03d062dd4bf51df47f33a30f2088056b921 0.0s ⇒ > extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3ede73437b02be 0.0s ⇒ > extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 0.0s ⇒ > extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 0.0s ⇒ > extracting sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 0.0s ⇒ > exporting to image 0.1s ⇒ > exporting layers 0.0s ⇒ > writing image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 0.0s ⇒ > naming to docker.io/library/my-postgres-db 0.0s		
=> extracting sha256:4b5c59f2d82cd85059458a41c01ebe3b5a2c96f1d584b2e24574149ff0406161 => => extracting sha256:fe72764b907045579a9993c60f23f3213caf82dade388988d41ccff9c184f59 => extracting sha256:6ef8e2c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e77f0e451 => => extracting sha256:e71fe9d7ff11d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 => => extracting sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 => => extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f826d46c388a80804b3e9de73437b02be => => extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 => => extracting sha256:e18e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 => => extracting sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae => exporting to image => => exporting layers => exporting image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 => => naming to docker.io/library/my-postgres-db 0.1s => => naming to docker.io/library/my-postgres-db		
=> extracting sha256:fe72764b907045579a9993c60f23f3213caf82d2ade388988d41ccff9c184f59 0.0s => extracting sha256:6ef8e2c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e77f0e451 0.0s => extracting sha256:e7f9d7ff11d95e9c7e54d020d28f20e5262263af563100c8df0508820b2707 5.7s => extracting sha256:f3225d69190dd8f4d7a026cdaebfe03d062d04bf51df47f33a30f2088056b921 0.0s => extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3e9de73437b02be 0.0s => extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 0.0s => extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 0.0s => exporting to image 0.1s => exporting layers 0.0s => writing image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 0.0s => naming to docker.io/library/my-postgres-db 0.0s		
=> extracting sha256:6ef8e2c0f4d924192a8ac20be4b72e17ed0933c5edbf5007a295bf8e77f0e451 0.0s => extracting sha256:e71fe9d7ff11d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 5.7s => extracting sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 0.0s => extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3e9de73437b02be 0.0s => extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 0.0s => extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 0.0s => exporting sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 0.0s => exporting lo image 0.1s => exporting lo image 0.0s => exporting		
=> => extracting sha256:e71fe9d7ff11d95e9c7e54d020428f20e5262263af563100c8df0508820b2707 5.7s => => extracting sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 0.0s => => extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3e9de73437b02be 0.0s => => extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 0.0s => => extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 0.0s => => extracting sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 0.0s => exporting to image 0.1s => => exporting layers 0.0s => => writing image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 0.0s => => naming to docker.io/library/my-postgres-db 0.0s		
=> => extracting sha256:f3225d69190dd8f447a026cdaebfe03d062d04bf51df47f33a30f2088056b921 0.0s => => extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3e9de73437b02be 0.0s => => extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342 0.0s => => extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7 0.0s => => extracting sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 0.0s => exporting to image 0.1s => => exporting layers 0.0s => => writing image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 0.0s => => naming to docker.io/library/my-postgres-db 0.0s		
=> => extracting sha256:2bf90d17afc8b5ec96e10e71bb08c75b3f82646c388a8084b3e9de73437b02be		
=> => extracting sha256:d3aee49eb07954f0cb1c957ca0998e5c635d0617ab593f42433c97ed16312342		
=> => extracting sha256:e1e8566589194f439ba266b753710f59da6a5de9715724c6249a88230a8211c7		
=> => extracting sha256:95c2c2ef9f02d7666e80992c98c53c9ec7b5e8ccf244d00a5c85e46bbc2820ae 0.0s => exporting to image 0.1s => => exporting layers 0.0s => => writing image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 0.0s => => naming to docker.io/library/my-postgres-db 0.0s		
=> exporting to image		
=> => exporting layers		
=> => writing image sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275 0.0s => => naming to docker.io/library/my-postgres-db 0.0s		
=> => naming to docker.io/library/my-postgres-db 0.0s		
	<pre>=> naming to docker.lo/library/my-postgres-db root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/database#</pre>	0.0s

3) Run postgreSQL container

docker run --name postgres-container --network fullstack-network -v pgdata:/var/lib/postgresql/data -d my-postgres-db

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/database# docker run --name postgres-container --network fullstack-network -v pgdata:/var/lib/postgresql/data -d my-postgres-db
66102c0b6d073ceec650cbe77ea8c6fcc5391f8159f5027ab20e61d6358eeed0

create node.js application with Express and set it up with Docker

1)Initialize the node.js Application. Before that we checke version of node and npm

npm init -y

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# node --version
v10.19.0
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# npm --version
6.14.4
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# npm init -y
Wrote to /home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend/package.json:
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  "keywords": [],
  "author": "",
  "license": "ISC"
```

2) install Express and pg. This are postgreSQL client for Node.js

npm install express pg

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# npm install express pg
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN backend@1.0.0 No description
npm WARN backend@1.0.0 No repository field.

+ express@4.19.2
+ pg@8.12.0
added 78 packages from 49 contributors and audited 78 packages in 7.038s

12 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# ls
node_modules package.json package-lock.json
```

3) In backend directory create file name "index.js"

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# nano index.js
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# code index.js
You are trying to start Visual Studio Code as a super user which isn't recommended. If this was intended, please add the argument `--no-sandbox` and specify an alternate user data dir
ectory using the `--user-data-dir` argument.
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# cat index.js
const express = require('express');
const { Pool } = require('pg');
const app = express();
const port = 3000;
const pool = new Pool({
   user: 'user',
    host: 'postgres-container',
    database: 'mydatabase',
   password: 'password',
    port: 5432,
});
app.get('/', (req, res) => {
    res.send('Hello from Node.js and Docker!');
app.get('/data', async (req, res) => {
    const client = await pool.connect();
    const result = await client.query('SELECT NOW()');
    client.release();
    res.send(result.rows);
});
app.listen(port, () => {
    console.log(`App running on http://localhost:${port}`);
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend#
```

4) create "dockerfile" in "backend" directory

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# nano dockerfile root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# cat dockerfile FROM node:latest

WORKDIR /usr/src/app

COPY package*.json ./
RUN npm install

COPY . .

EXPOSE 3000

CMD ["node", "index.js"]
```

5) build backend image

docker build -t my-node-app.

6)run backend container

docker run --name backend-container --network fullstack-network -d my-node-app

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/backend# docker run --name backend-container --network fullstack-network -d my-node-app c52b28aa5ccc7a5b412d6169ab6b2c972558cd3c8c45d0b91b2b6ad55d284576

create static front-end and set it up with docker

1) in "frontend" directory create "index.html "file

2) in "frontend" directory create "dockerfile"

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# nano dockerfile
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cat dockerfile
FROM nginx:latest
COPY index.html /usr/share/nginx/html/index.html

3) build frontend Image and Run frontend conatiner

docker build -t my-nginx-app .

cd ..

docker run --name frontend-container --network fullstack-network -p 8080:80 -d my-nginx-app

Connecting the Backend and Database

Backend can communicate with database and handle data requests

1) Access backend container

docker exec -it backend-container /bin/bash

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker exec -it backend-container /bin/bash

2) Test the connection to the database using psql apt-get update && apt-get install -y postgresgl-client

psql -h postgres-container -U user -d mydatabase -c "SELECT NOW();"

```
root@c52b28aa5ccc:/usr/src/app# apt-get update && apt-get install -y postgresql-client
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm-updates InRelease
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  postgresgl-client-15 postgresgl-client-common
Suggested packages:
  postgresql-15 postgresql-doc-15
The following NEW packages will be installed:
  postgresql-client postgresql-client-15 postgresql-client-common
0 upgraded, 3 newly installed, 0 to remove and 10 not upgraded.
Need to get 1744 kB of archives.
After this operation, 8244 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 postgresgl-client-common all 248 [35.1 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 postgresgl-client-15 amd64 15.7-0+deb12u1 [1699 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 postgresql-client all 15+248 [10.1 kB]
Fetched 1744 kB in 5s (363 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package postgresql-client-common.
(Reading database ... 23243 files and directories currently installed.)
Preparing to unpack .../postgresql-client-common_248_all.deb ...
Unpacking postgresql-client-common (248) ...
Selecting previously unselected package postgresql-client-15.
Preparing to unpack .../postgresql-client-15_15.7-0+deb12u1_amd64.deb ...
Unpacking postgresql-client-15 (15.7-0+deb12u1) ...
Selecting previously unselected package postgresql-client.
Preparing to unpack .../postgresql-client_15+248_all.deb ...
Unpacking postgresql-client (15+248) ...
Setting up postgresql-client-common (248) ...
Setting up postgresql-client-15 (15.7-0+deb12u1) ...
update-alternatives: using /usr/share/postgresql/15/man/man1/psql.1.gz to provide /usr/share/man/man1/psql.1.gz (psql.1.gz) in auto mode
Setting up postgresql-client (15+248) ...
```

Final Integration and Testing

1) visit the browser use localhost:8080 to se the Nginx welcome page



2) On "frontend" branch update "index.html" file to include backend link. Using this we will verify full Integration

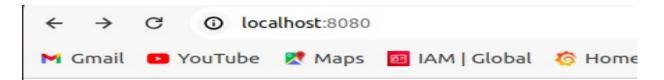
```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# cd frontend/
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# ls
dockerfile index.html
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# nano index.html
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app/frontend# cat index.html
<!DOCTYPE html>
<html>
<br/>
<html>
This is a simple static front-end served by Nginx.
<a href="http://localhost:3000/data">Nginx.
</body>
</html>
```

3) Rebuild and run the updated Frontend Container

docker build -t my-nginx-app .

docker run --name frontend-container --network fullstack-network -p 8080:80 -d mynginx-app

4) to verify visit browser by using localhost:8080



Hello from Nginx and Docker!

This is a simple static front-end served by Nginx.

Fetch Data from Backend

Remove all created containers, Images , networks and volumes to cleanup environment

1) stop and remove the container

docker stop frontend-container backend-container postgres-container docker rm frontend-container backend-container postgres-container

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker stop frontend-container backend-container postgres-container backend-container backend-container backend-container postgres-container postgres-container root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker rm frontend-container backend-container postgres-container frontend-container

backend-container

postgres-container

2) Remove the images:

docker rmi my-nginx-app my-node-app my-postgres-db

root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker rmi my-nginx-app my-node-app my-postgres-db

Untagged: my-nginx-app:latest

Deleted: sha256:b0619b2ccb0653a4ceb2606bf2bfb9662fe0bb824a8cab5109eda9056bab6613

Untagged: my-node-app:latest

Deleted: sha256:b47e31cbcbf247b69718a6dafb02c9bc4e734daf6590a1de0ade305ed7301e4f

Untagged: my-postgres-db:latest

Deleted: sha256:1cfe6ff8722f48bf2889ab46a727e0e5ed0bf4046b570b920262d21e57ca2275

3) remove the network and volume

docker network rm fullstack-network docker volume rm pgdata

```
root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker network rm fullstack-network root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# docker volume rm pgdata pgdata root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# cd.. cd.: command not found root@PUNELPT0393:/home/einfochips/Desktop/learning/10july/fullstack-docker-app# cd .. root@PUNELPT0393:/home/einfochips/Desktop/learning/10july# ls 10july 'Docker' fullstack-docker-app my-webapp root@PUNELPT0393:/home/einfochips/Desktop/learning/10july# []
```