Lab 2: Docker

Learning objectives

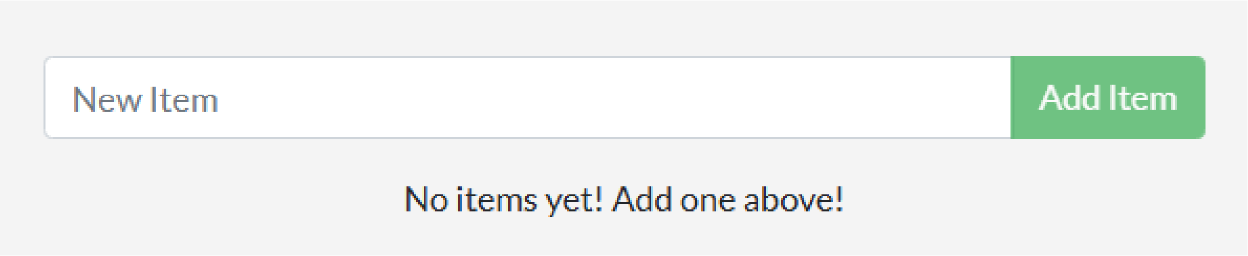
By the end of this lab, students will be able to:

- Understand and explain key Docker concepts: images, containers, and registries.
- Create a Dockerfile to containerize a simple Node.js or Python web app.
- Build and tag Docker images using the CLI.
- Run containers and map ports between host and container.
- Push and pull images to and from Docker Hub.
- Use volumes and bind mounts for persistent storage.
- Define and deploy multi-container applications with Docker Compose.

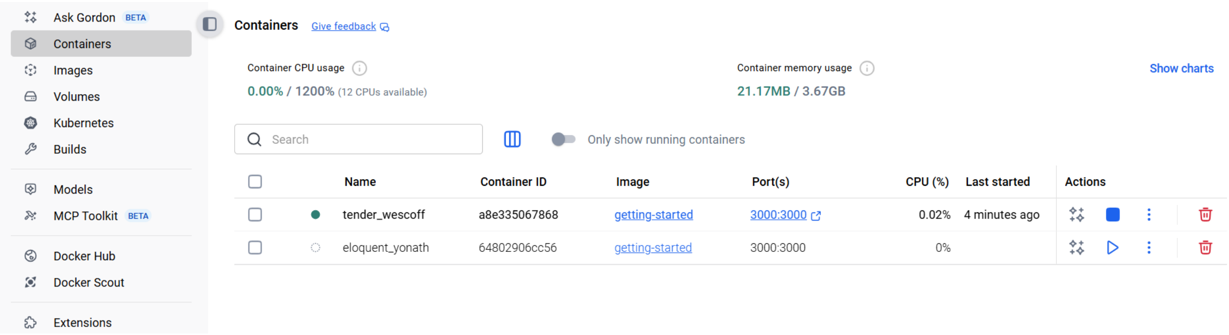
Git Link: https://github.com/Somawatey/Khov-Somawatey-Lab2-docker

Part 1: Containerize an application

I successfully built the Docker image and ran the container. The application worked correctly in the browser on my local machine.



No items yet! Add one above!



Containers

Container CPU usage
0.00% / 1200% (12 CPUs available)

Container memory usage
21.17MB / 3.67GB

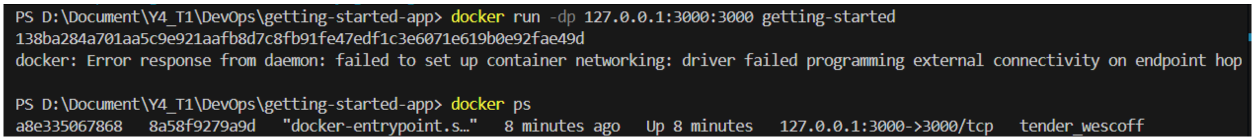
Only show running containers

□		Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
□	●	tender_wescoff	a8e335067868	getting-started	3000:3000	0.02%	4 minutes ago	
□	○	eloquent_yonath	64802906cc56	getting-started	3000:3000	0%		

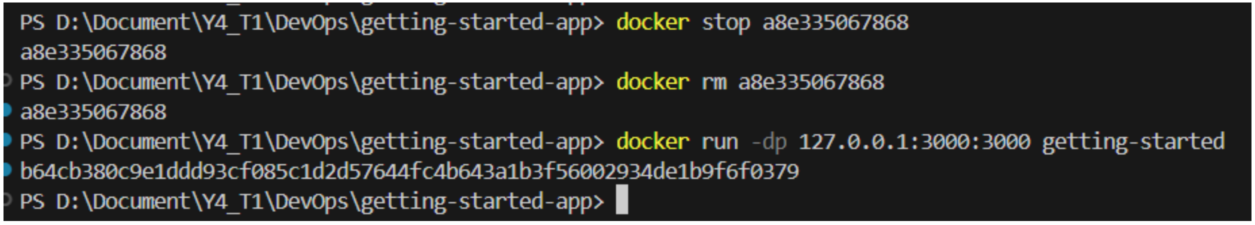
Part2: update application

The previous container with ID **a8e335067868** was removed because it caused an error. I followed the instructions by checking the container ID, stopping it, and then removing it.

After updating, I rebuilt and restarted the new container, and the updated application worked properly.

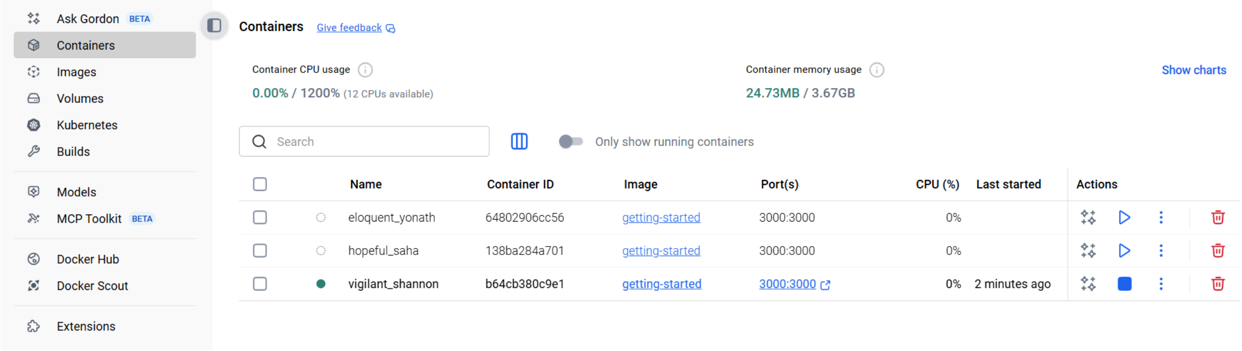


```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker run -dp 127.0.0.1:3000:3000 getting-started
138ba284a701aa5c9e921aafb8d7c8fb91fe47edf1c3e6071e619b0e92fae49d
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint hop

PS D:\Document\Y4_T1\DevOps\getting-started-app> docker ps
a8e335067868   8a58f9279a9d   "docker-entrypoint.s…"   8 minutes ago   Up 8 minutes   127.0.0.1:3000->3000/tcp   tender_wescoff
```



```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker stop a8e335067868
a8e335067868
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker rm a8e335067868
a8e335067868
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker run -dp 127.0.0.1:3000:3000 getting-started
b64cb380c9e1ddd93cf085c1d2d57644fc4b643a1b3f56002934de1b9f6f0379
PS D:\Document\Y4_T1\DevOps\getting-started-app> 
```

After updating the container



Containers Give feedback

Container CPU usage
0.00% / 1200% (12 CPUs available)

Container memory usage
24.73MB / 3.67GB

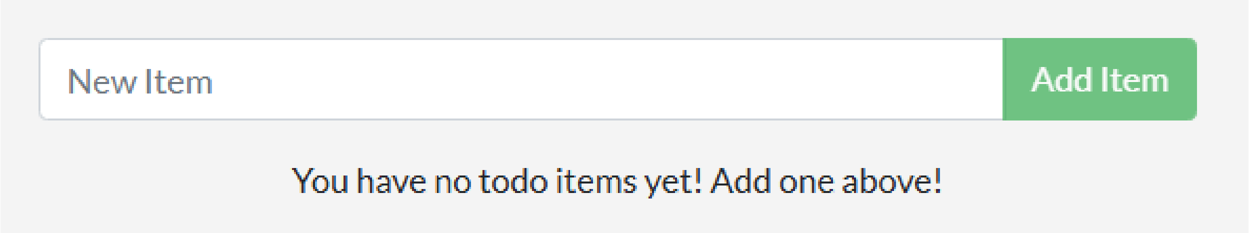
Show charts

Search

Only show running containers

□		Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
□	○	eloquent_yonath	64802906cc56	getting-started	3000:3000	0%		
□	○	hopeful_saha	138ba284a701	getting-started	3000:3000	0%		
□	●	vigilant_shannon	b64cb380c9e1	getting-started	3000:3000	0%	2 minutes ago	

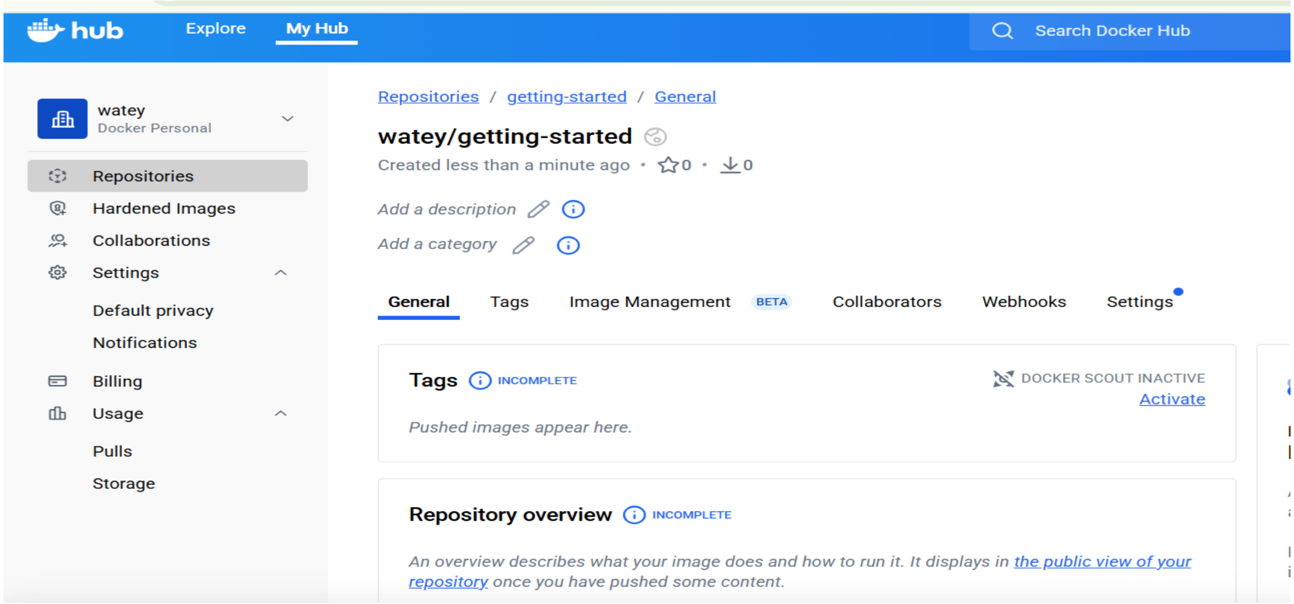
After updating the application, I replaced the old container.



You have no todo items yet! Add one above!

Part3: Share Application

I created a new repository on Docker Hub.



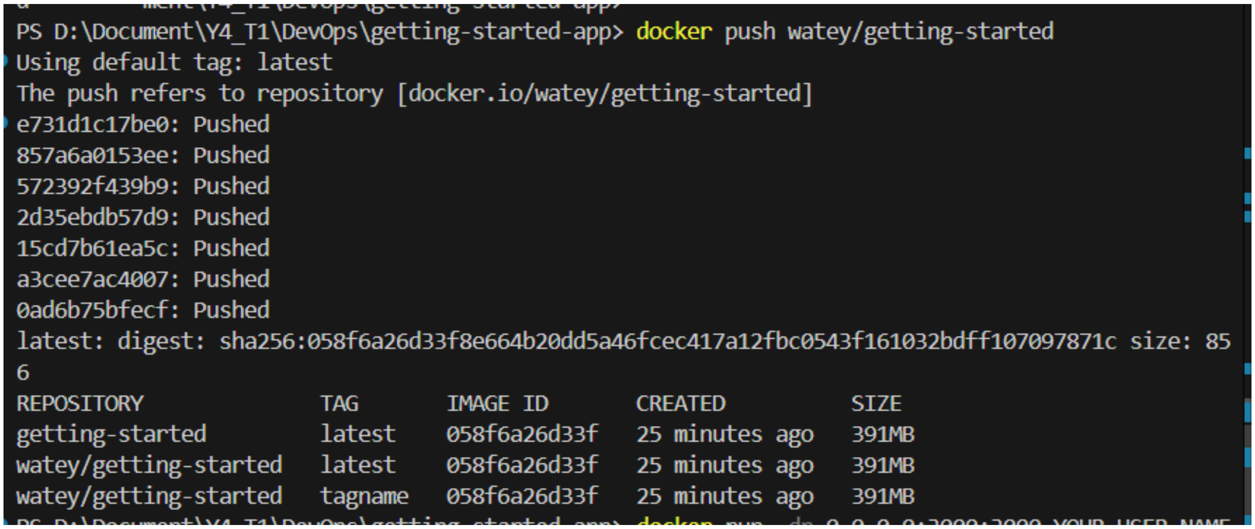
watey/getting-started

Created less than a minute ago • ☆0 • ⤓0

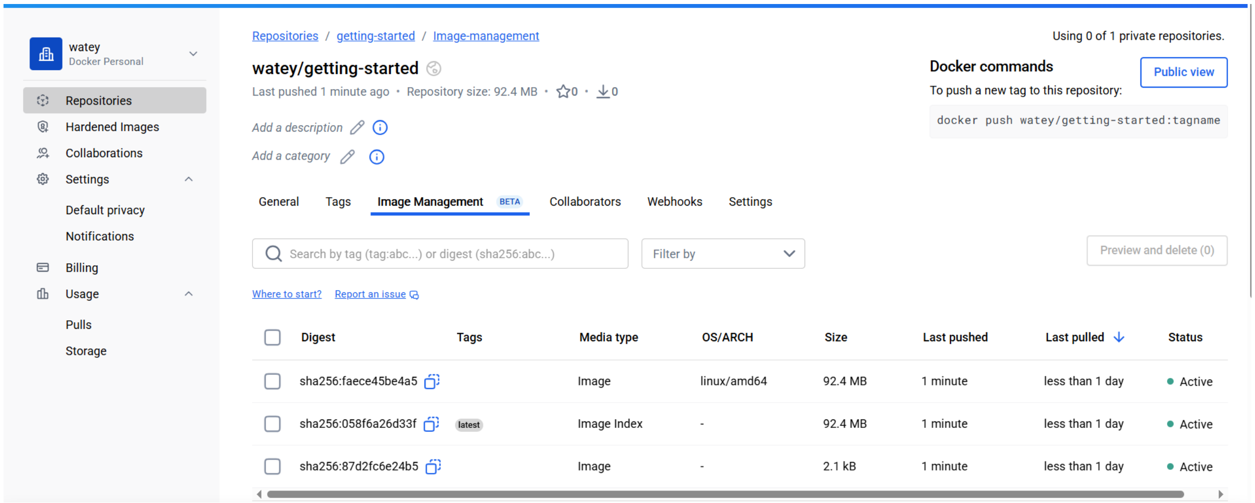
Pushed images appear here.

An overview describes what your image does and how to run it. It displays in the public view of your repository once you have pushed some content.

Then I pushed my Docker image to Docker Hub successfully.



```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker push watey/getting-started
Using default tag: latest
The push refers to repository [docker.io/watey/getting-started]
e731d1c17be0: Pushed
857a6a0153ee: Pushed
572392f439b9: Pushed
2d35ebdb57d9: Pushed
15cd7b61ea5c: Pushed
a3cee7ac4007: Pushed
0ad6b75bfecf: Pushed
latest: digest: sha256:058f6a26d33f8e664b20dd5a46fcec417a12fbc0543f161032bdff107097871c size: 85
6
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
getting-started         latest    058f6a26d33f   25 minutes ago   391MB
watey/getting-started   latest    058f6a26d33f   25 minutes ago   391MB
watey/getting-started   tagname   058f6a26d33f   25 minutes ago   391MB
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker run -dp 0.0.0.0:3000:3000 YOUR-USER-NAME
```



Using 0 of 1 private repositories.

watey/getting-started

Last pushed 1 minute ago • Repository size: 92.4 MB • ☆0 • ⤓0

Add a description

Add a category

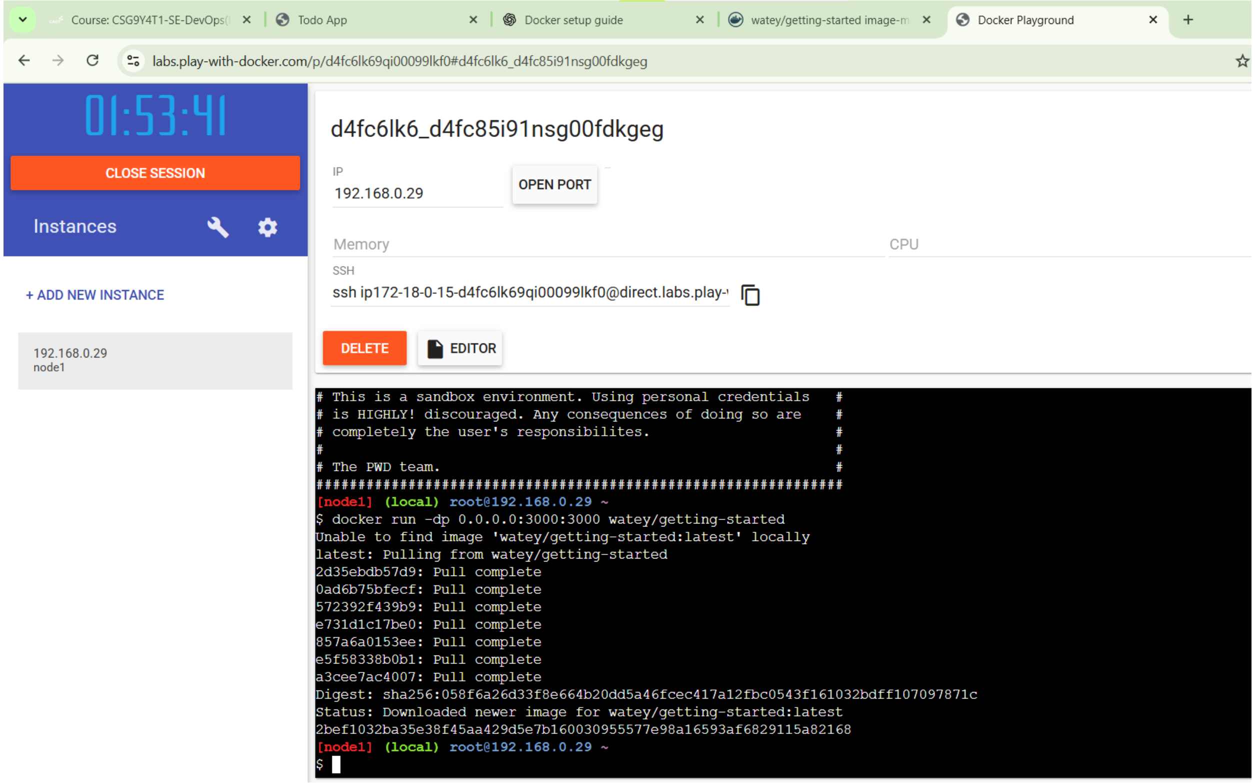
Docker commands

To push a new tag to this repository:

```
docker push watey/getting-started:tagname
```

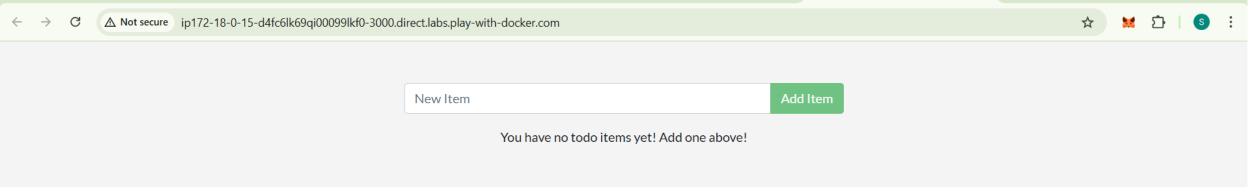
□	Digest	Tags	Media type	OS/ARCH	Size	Last pushed	Last pulled ↓	Status
□	sha256:faece45be4a5		Image	linux/amd64	92.4 MB	1 minute	less than 1 day	● Active
□	sha256:058f6a26d33f	latest	Image Index	-	92.4 MB	1 minute	less than 1 day	● Active
□	sha256:87d2fc6e24b5		Image	-	2.1 kB	1 minute	less than 1 day	● Active

After that, I tested the image using **Play With Docker**.


Course: CSG9Y4T1-SE-DevOps
Todo App
Docker setup guide
watey/getting-started image-m
Docker Playground
labs.play-with-docker.com/p/d4fc6lk69qi00099lkf0#d4fc6lk6_d4fc85i91nsg00fdkgeg
01:53:41
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.29
node1
d4fc6lk6_d4fc85i91nsg00fdkgeg
IP
192.168.0.29
OPEN PORT
Memory
CPU
SSH
ssh ip172-18-0-15-d4fc6lk69qi00099lkf0@direct.labs.play-
DELETE
EDITOR

```
# This is a sandbox environment. Using personal credentials   #
# is HIGHLY! discouraged. Any consequences of doing so are    #
# completely the user's responsibilites.                      #
#                                                             #
# The PWD team.                                               #
###############################################################
[node1] (local) root@192.168.0.29 ~
$ docker run -dp 0.0.0.0:3000:3000 watey/getting-started
Unable to find image 'watey/getting-started:latest' locally
latest: Pulling from watey/getting-started
2d35ebdb57d9: Pull complete
0ad6b75bfecf: Pull complete
572392f439b9: Pull complete
e731d1c17be0: Pull complete
857a6a0153ee: Pull complete
e5f58338b0b1: Pull complete
a3cee7ac4007: Pull complete
Digest: sha256:058f6a26d33f8e664b20dd5a46fcec417a12fbc0543f161032bdff107097871c
Status: Downloaded newer image for watey/getting-started:latest
2bef1032ba35e38f45aa429d5e7b160030955577e98a16593af6829115a82168
[node1] (local) root@192.168.0.29 ~
$ 
```

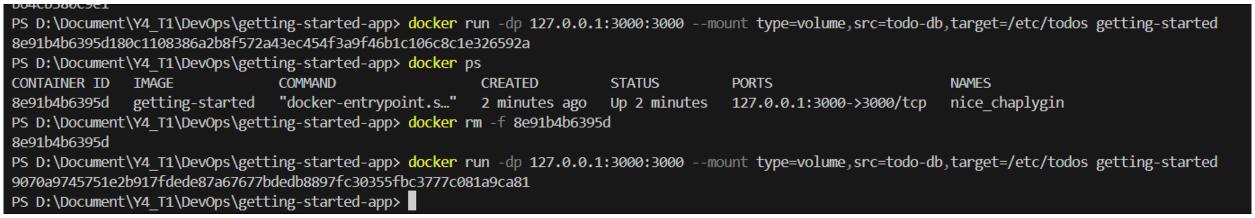
I ran the container and opened port **3000**, and the web app worked correctly in the online environment.


Not secure
ip172-18-0-15-d4fc6lk69qi00099lkf0-3000.direct.labs.play-with-docker.com
New Item
Add Item
You have no todo items yet! Add one above!

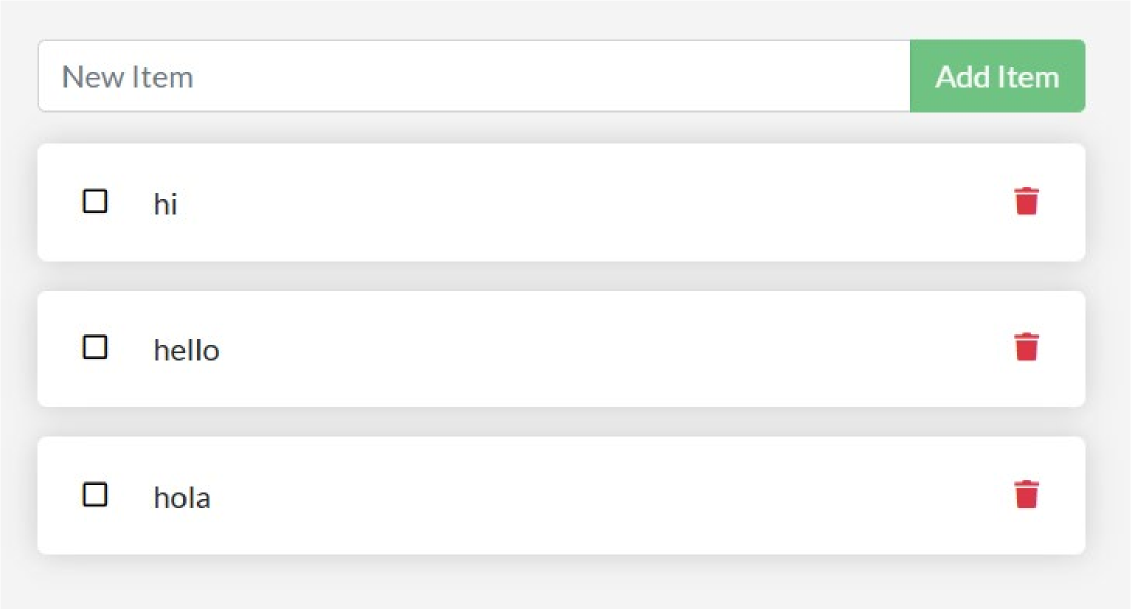
Part 4: Persist the DB

I ran a new container and created items in the list on port **3000**.

Then I removed the container and started a new one. The data still existed, meaning the volume worked correctly and the data was persistent.



```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker run -dp 127.0.0.1:3000:3000 --mount type=volume,src=todo-db,target=/etc/todos getting-started
8e91b4b6395d180c1108386a2b8f572a43ec454f3a9f46b1c106c8c1e326592a
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED         STATUS         PORTS                      NAMES
8e91b4b6395d   getting-started   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   127.0.0.1:3000->3000/tcp   nice_chaplygin
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker rm -f 8e91b4b6395d
8e91b4b6395d
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker run -dp 127.0.0.1:3000:3000 --mount type=volume,src=todo-db,target=/etc/todos getting-started
9070a9745751e2b917fdede87a67677bdedb8897fc30355fbc3777c081a9ca81
PS D:\Document\Y4_T1\DevOps\getting-started-app> 
```

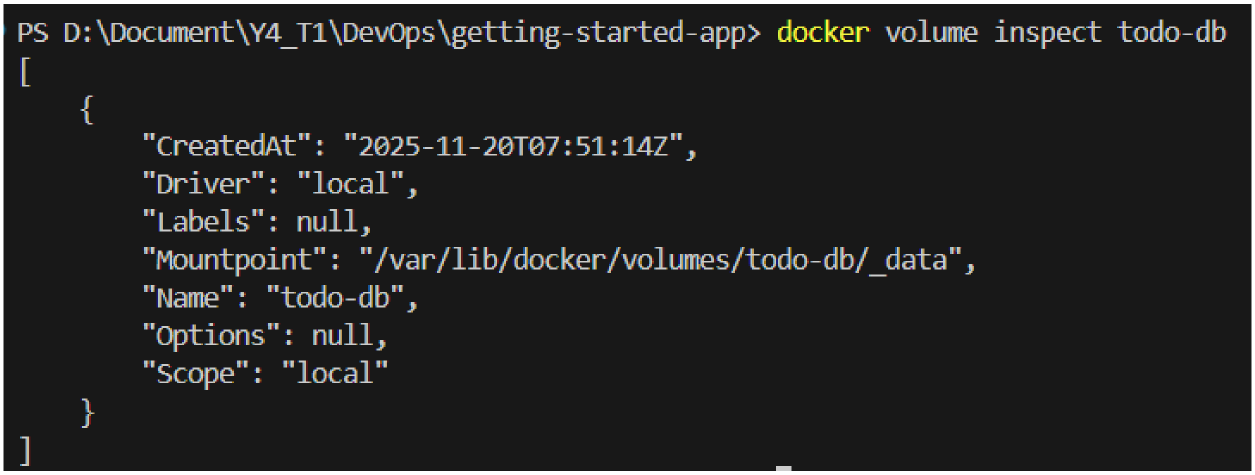


□ hi

□ hello

□ hola

I also checked where Docker stores the data.
The **Mountpoint** shows the actual location of the stored data on the host machine.

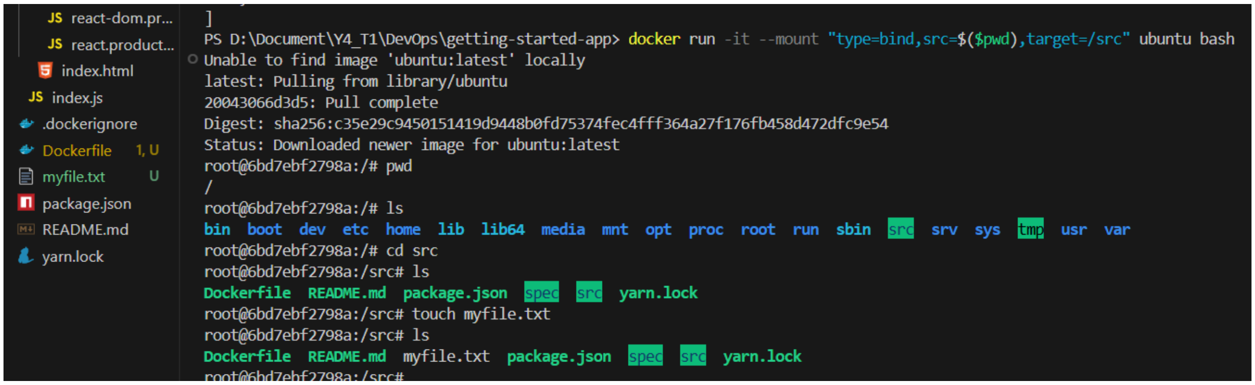


```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker volume inspect todo-db
[
    {
        "CreatedAt": "2025-11-20T07:51:14Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/todo-db/_data",
        "Name": "todo-db",
        "Options": null,
        "Scope": "local"
    }
]
```

The **Mountpoint** shows the actual location of the stored data on the host machine.

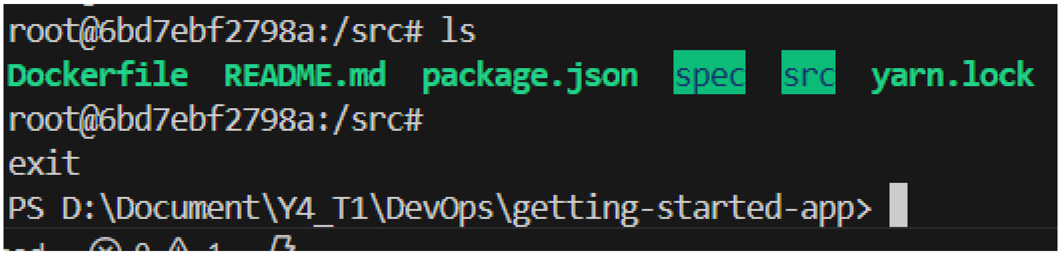
Part 5: Use bind mounts

I created a myfile.txt file inside the src directory using the container. The file appeared both inside the container and on the host system.



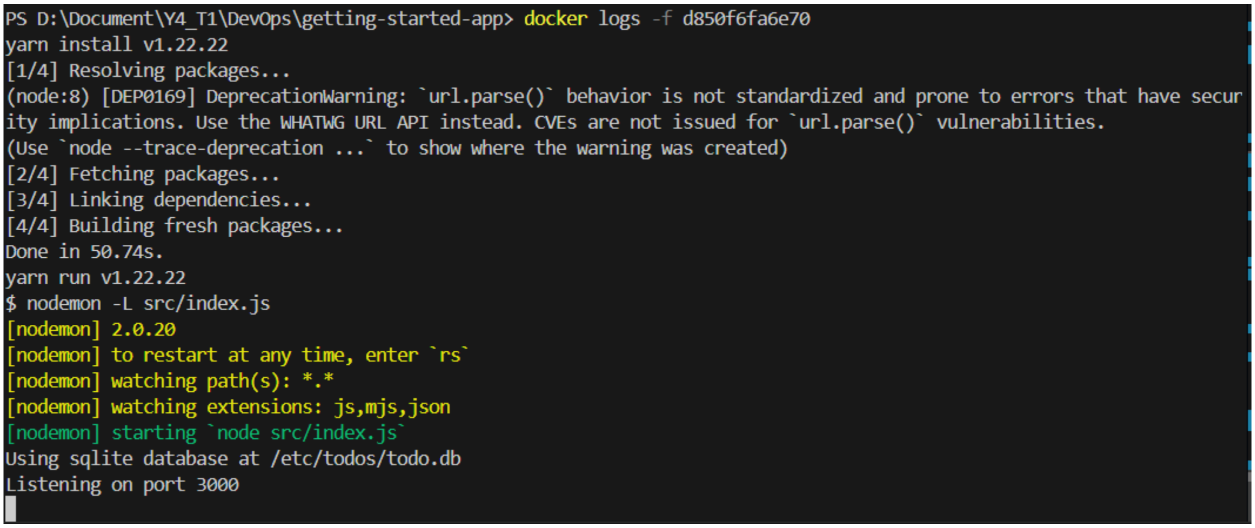
```
]
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker run -it --mount "type=bind,src=$($pwd),target=/src" ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
20043066d3d5: Pull complete
Digest: sha256:c35e29c9450151419d9448b0fd75374fec4fff364a27f176fb458d472dfc9e54
Status: Downloaded newer image for ubuntu:latest
root@6bd7ebf2798a:/# pwd
/
root@6bd7ebf2798a:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  src  srv  sys  tmp  usr  var
root@6bd7ebf2798a:/# cd src
root@6bd7ebf2798a:/src# ls
Dockerfile  README.md  package.json  spec  src  yarn.lock
root@6bd7ebf2798a:/src# touch myfile.txt
root@6bd7ebf2798a:/src# ls
Dockerfile  README.md  myfile.txt  package.json  spec  src  yarn.lock
root@6bd7ebf2798a:/src#
```

After deleting the file from the host, it was also removed from the container.



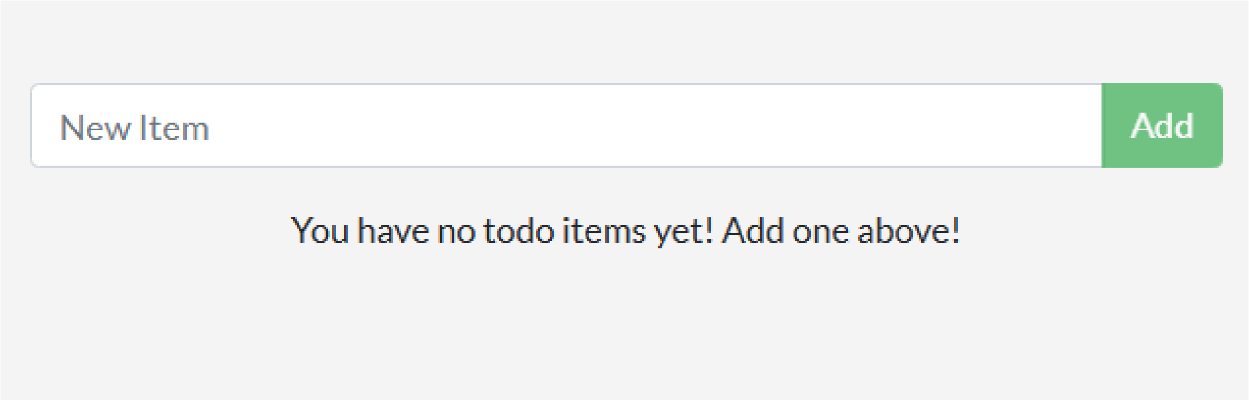
```
root@6bd7ebf2798a:/src# ls
Dockerfile  README.md  package.json  spec  src  yarn.lock
root@6bd7ebf2798a:/src#
exit
PS D:\Document\Y4_T1\DevOps\getting-started-app>
```

This proved the bind mount synchronization between host and container.



```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker logs -f d850f6fa6e70
yarn install v1.22.22
[1/4] Resolving packages...
(node:8) [DEP0169] DeprecationWarning: `url.parse()` behavior is not standardized and prone to errors that have secur
ity implications. Use the WHATWG URL API instead. CVEs are not issued for `url.parse()` vulnerabilities.
(Use `node --trace-deprecation ...` to show where the warning was created)
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
Done in 50.74s.
yarn run v1.22.22
$ nodemon -L src/index.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
Using sqlite database at /etc/todos/todo.db
Listening on port 3000
```

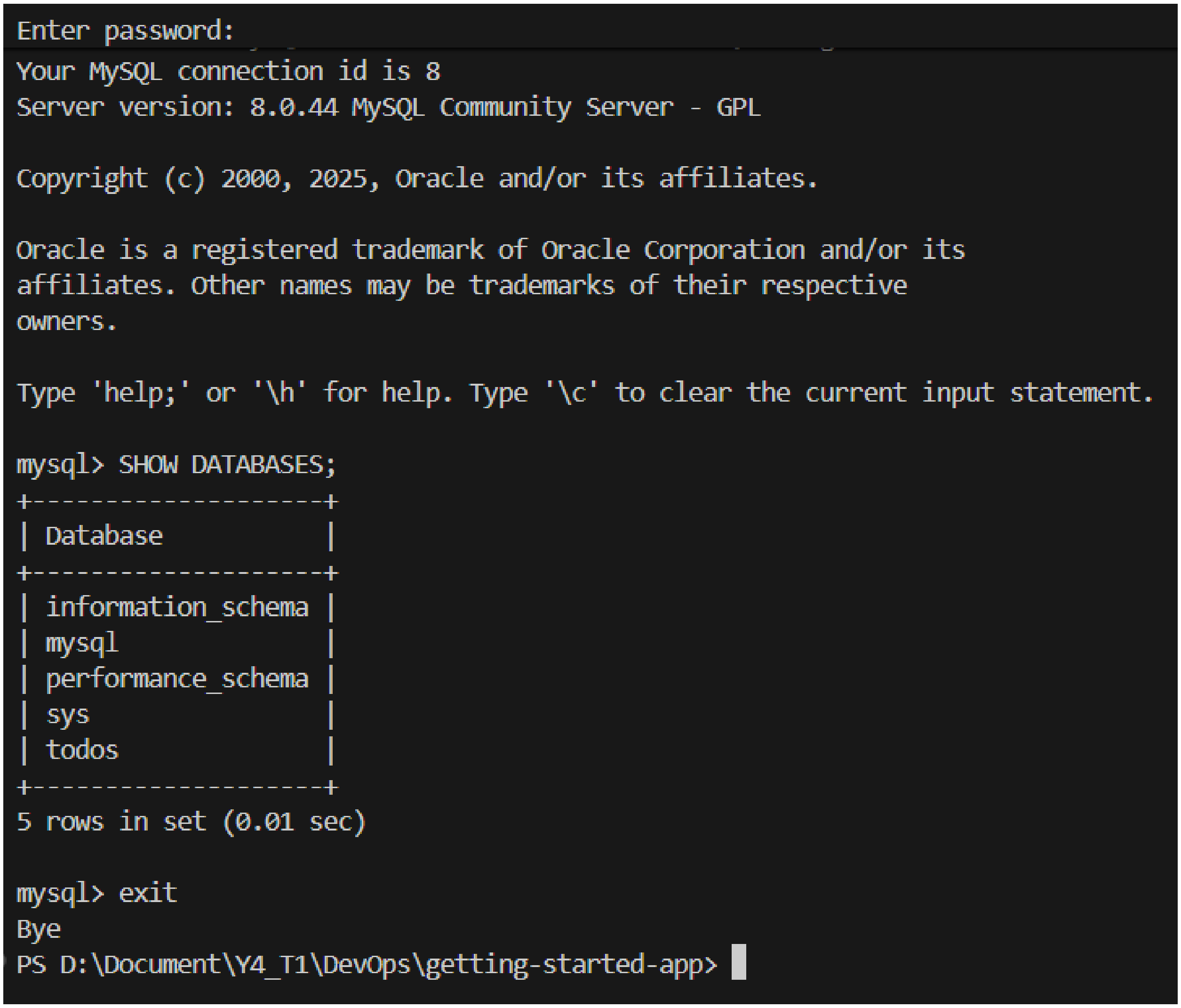
I also updated the UI text from **“Add Item”** to **“Add ”**, and the changes were visible immediately.



You have no todo items yet! Add one above!

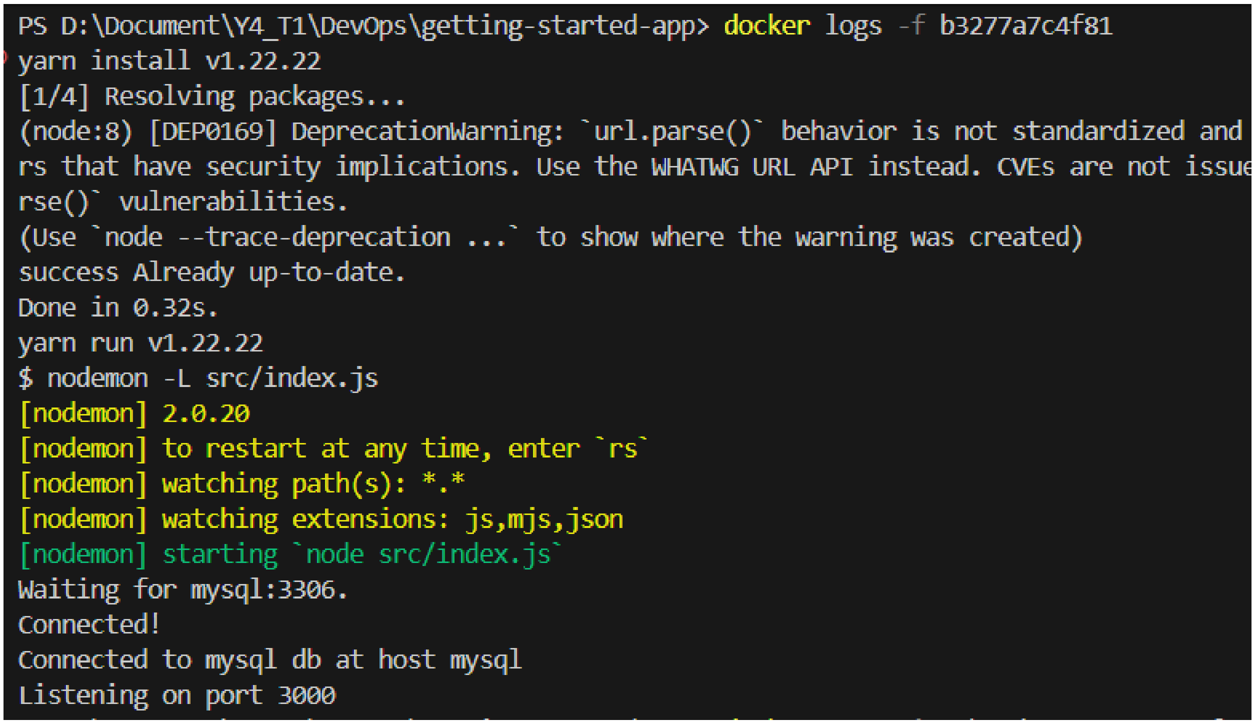
Part6: Multi-container apps

I created a new database named **todos**.



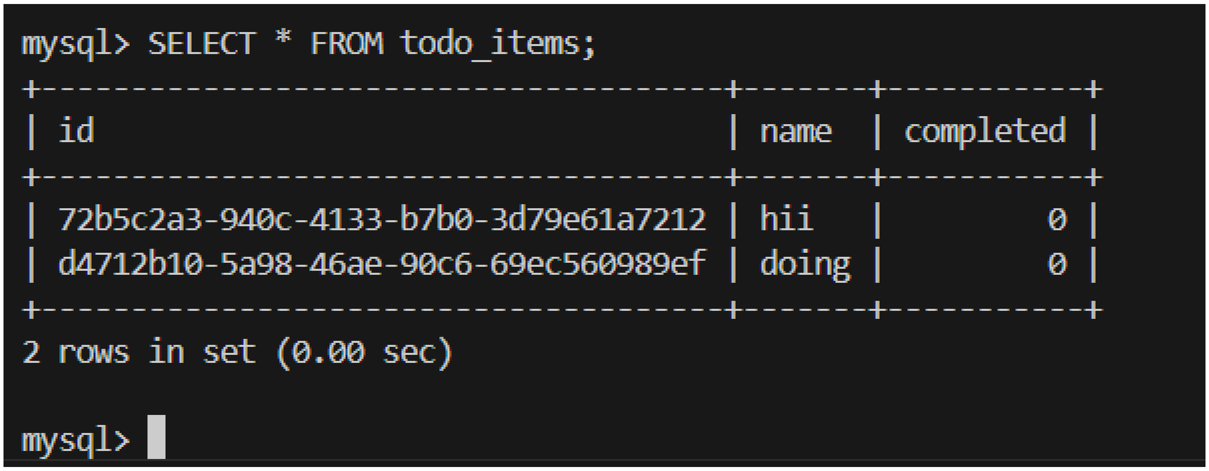
```
Enter password:
Your MySQL connection id is 8
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, 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> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| todos              |
+--------------------+
5 rows in set (0.01 sec)

mysql> exit
Bye
PS D:\Document\Y4_T1\DevOps\getting-started-app>
```

By checking the Docker logs, I confirmed that the app successfully connected to the MySQL database running on the host named mysql.



```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker logs -f b3277a7c4f81
yarn install v1.22.22
[1/4] Resolving packages...
(node:8) [DEP0169] DeprecationWarning: `url.parse()` behavior is not standardized and
rs that have security implications. Use the WHATWG URL API instead. CVEs are not issue
rse()` vulnerabilities.
(Use `node --trace-deprecation ...` to show where the warning was created)
success Already up-to-date.
Done in 0.32s.
yarn run v1.22.22
$ nodemon -L src/index.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
Waiting for mysql:3306.
Connected!
Connected to mysql db at host mysql
Listening on port 3000
```

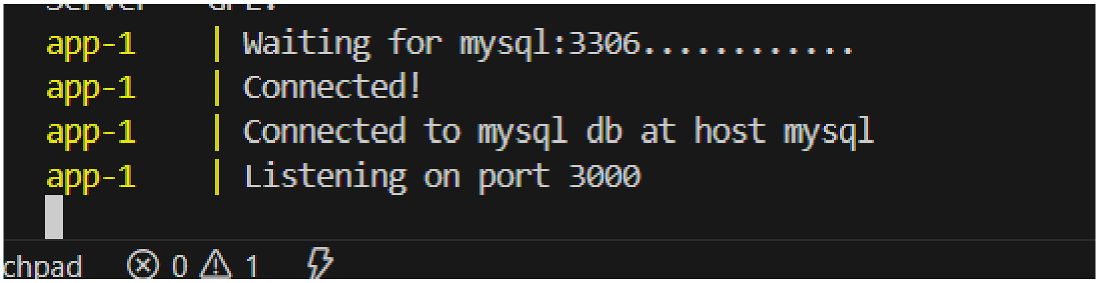
I also checked the MySQL container to verify that the data added through the app was correctly stored in the database.



```
mysql> SELECT * FROM todo_items;
+--------------------------------------+-------+-----------+
| id                                   | name  | completed |
+--------------------------------------+-------+-----------+
| 72b5c2a3-940c-4133-b7b0-3d79e61a7212 | hii   |         0 |
| d4712b10-5a98-46ae-90c6-69ec560989ef | doing |         0 |
+--------------------------------------+-------+-----------+
2 rows in set (0.00 sec)

mysql>
```

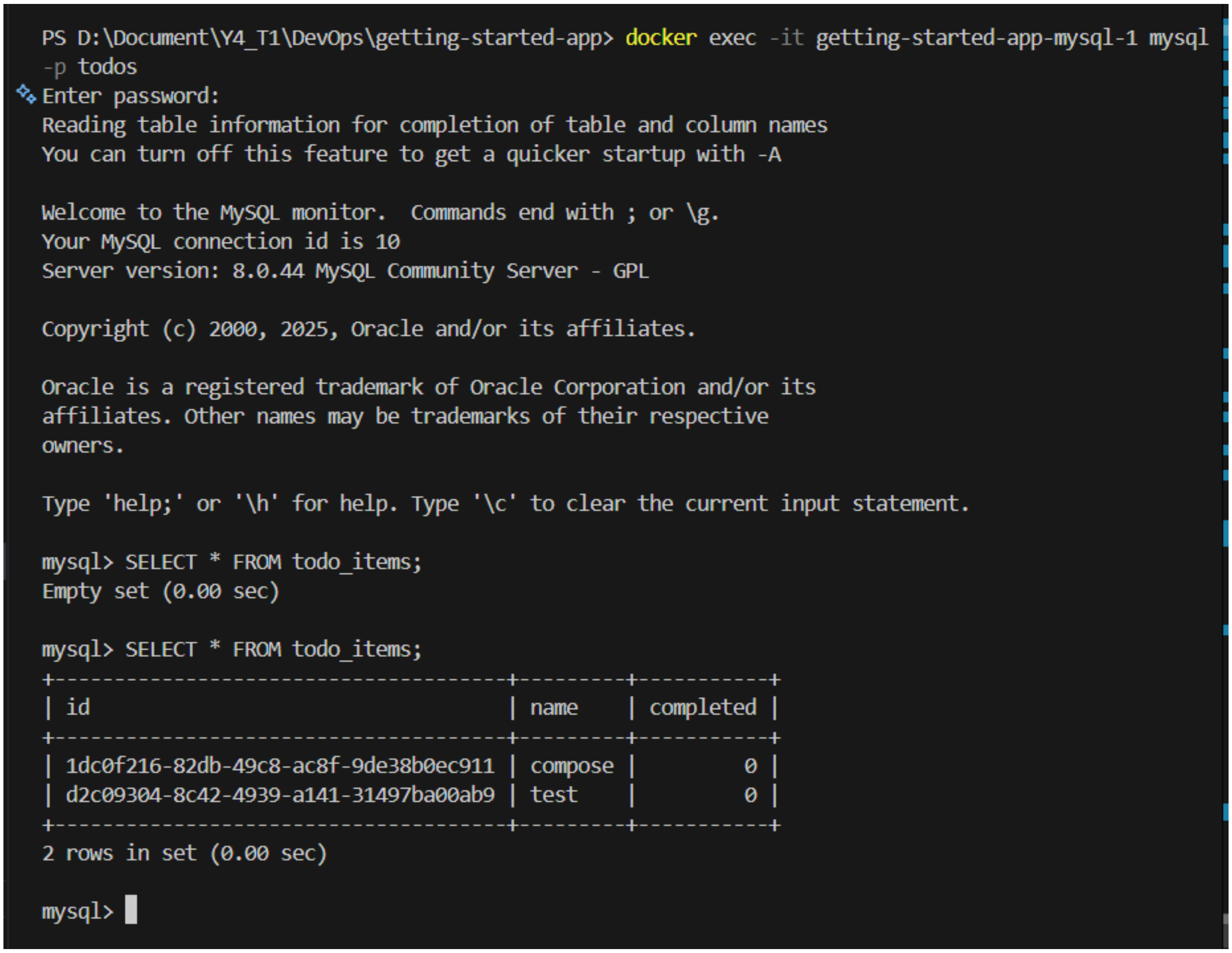
Part 7: Use Docker Compose

With Docker Compose, the application now connects to MySQL automatically. Running docker compose logs -f allowed me to see the real-time logs for both services.



```
app-1     | Waiting for mysql:3306............
app-1     | Connected!
app-1     | Connected to mysql db at host mysql
app-1     | Listening on port 3000
```

I also checked to verify that the data added through the app was correctly stored in the database.



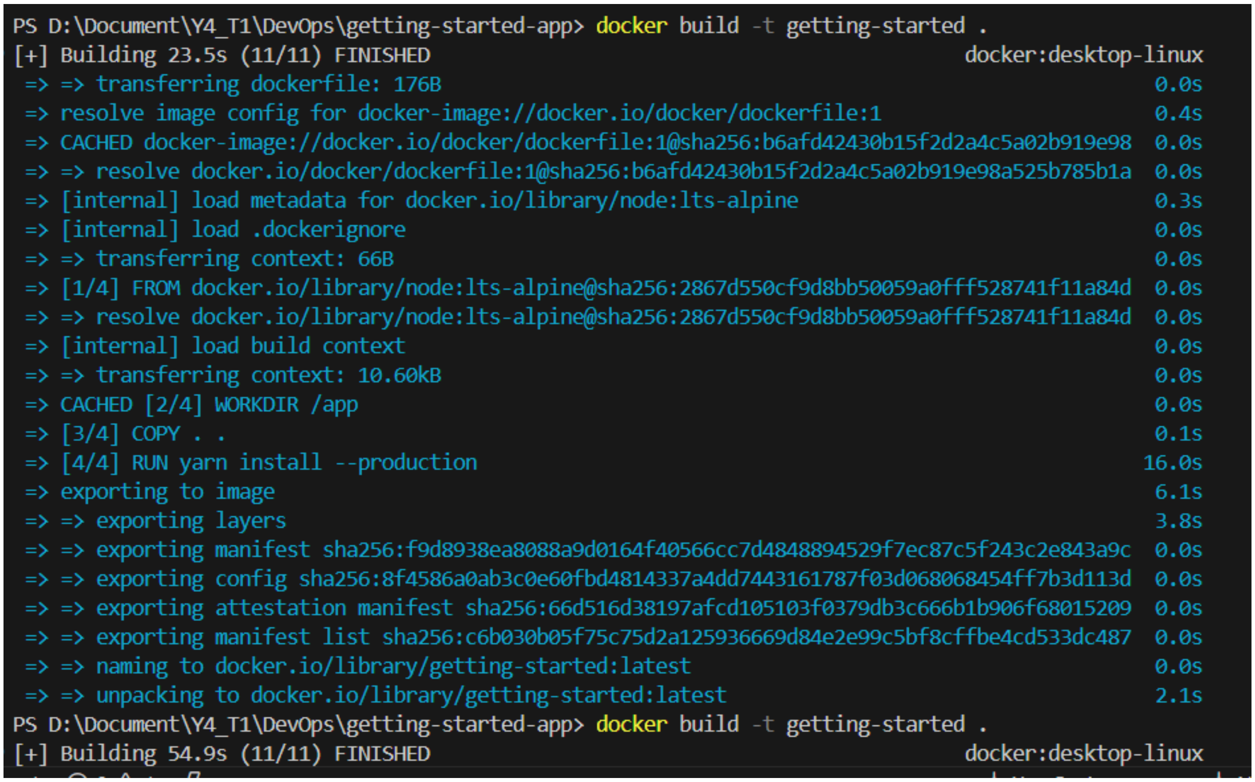
```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker exec -it getting-started-app-mysql-1 mysql
-p todos
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, 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> SELECT * FROM todo_items;
Empty set (0.00 sec)

mysql> SELECT * FROM todo_items;
+--------------------------------------+---------+-----------+
| id                                   | name    | completed |
+--------------------------------------+---------+-----------+
| 1dc0f216-82db-49c8-ac8f-9de38b0ec911 | compose |         0 |
| d2c09304-8c42-4939-a141-31497ba00ab9 | test    |         0 |
+--------------------------------------+---------+-----------+
2 rows in set (0.00 sec)

mysql> 
```

Docker Compose

- Single command to start/stop entire application
- Version controlled configuration
- Automatic network creation
- Easy to share with team members

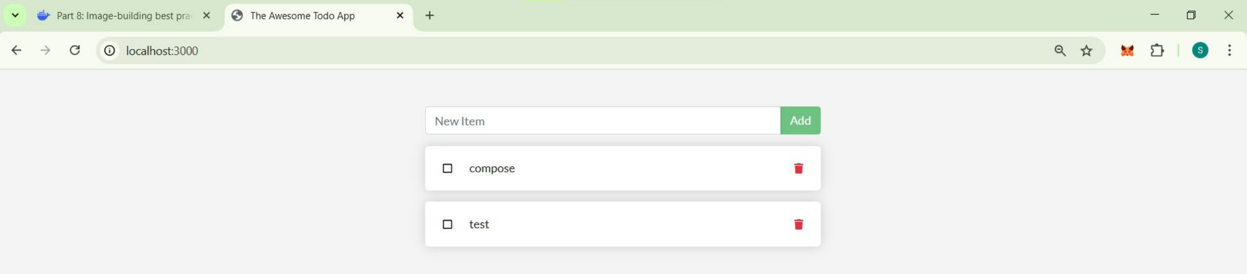
Part 8: Image-building best practices

After clearing the cache and rebuilding the application, the system ran faster and the updated changes were applied correctly.



```
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker build -t getting-started .
[+] Building 23.5s (11/11) FINISHED                                                 docker:desktop-linux
 => => transferring dockerfile: 176B                                                                0.0s
 => resolve image config for docker-image://docker.io/docker/dockerfile:1                           0.4s
 => CACHED docker-image://docker.io/docker/dockerfile:1@sha256:b6afd42430b15f2d2a4c5a02b919e98     0.0s
 => => resolve docker.io/docker/dockerfile:1@sha256:b6afd42430b15f2d2a4c5a02b919e98a525b785b1a      0.0s
 => [internal] load metadata for docker.io/library/node:lts-alpine                                 0.3s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 66B                                                                    0.0s
 => [1/4] FROM docker.io/library/node:lts-alpine@sha256:2867d550cf9d8bb50059a0fff528741f11a84d      0.0s
 => => resolve docker.io/library/node:lts-alpine@sha256:2867d550cf9d8bb50059a0fff528741f11a84d      0.0s
 => [internal] load build context                                                                   0.0s
 => => transferring context: 10.60kB                                                                0.0s
 => CACHED [2/4] WORKDIR /app                                                                       0.0s
 => [3/4] COPY . .                                                                                  0.1s
 => [4/4] RUN yarn install --production                                                            16.0s
 => exporting to image                                                                              6.1s
 => => exporting layers                                                                             3.8s
 => => exporting manifest sha256:f9d8938ea8088a9d0164f40566cc7d4848894529f7ec87c5f243c2e843a9c      0.0s
 => => exporting config sha256:8f4586a0ab3c0e60fbd4814337a4dd7443161787f03d068068454ff7b3d113d      0.0s
 => => exporting attestation manifest sha256:66d516d38197afcd105103f0379db3c666b1b906f68015209      0.0s
 => => exporting manifest list sha256:c6b030b05f75c75d2a125936669d84e2e99c5bf8cffbe4cd533dc487      0.0s
 => => naming to docker.io/library/getting-started:latest                                           0.0s
 => => unpacking to docker.io/library/getting-started:latest                                        2.1s
PS D:\Document\Y4_T1\DevOps\getting-started-app> docker build -t getting-started .
[+] Building 54.9s (11/11) FINISHED                                                 docker:desktop-linux
```

Rebuild and check what change.



□ compose

□ test