

# Cloud Data Centres:

## Lab 3

**By Sarah Martin**

**Student ID: C00257967**

## Part 2: Containerize An Application

I started off by cloning a git repository from <https://docs.docker.com/get-started/> into a folder called lab 3

```
C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>dir
Volume in drive C is Acer
Volume Serial Number is 94B0-A3A2

Directory of C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app

24/01/2024  14:06    <DIR>        .
24/01/2024  14:06    <DIR>        ..
24/01/2024  14:06                681 package.json
24/01/2024  14:06                273 README.md
24/01/2024  14:06    <DIR>        spec
24/01/2024  14:06    <DIR>        src
24/01/2024  14:06      150,541 yarn.lock
                3 File(s)      151,495 bytes
                4 Dir(s)  19,728,130,048 bytes free

C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>
```

I then looked at the contents of the getting-started-app with this command

```
C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>dir
Volume in drive C is Acer
Volume Serial Number is 94B0-A3A2

Directory of C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app

24/01/2024  14:06    <DIR>        .
24/01/2024  14:06    <DIR>        ..
24/01/2024  14:06                681 package.json
24/01/2024  14:06                273 README.md
24/01/2024  14:06    <DIR>        spec
24/01/2024  14:06    <DIR>        src
24/01/2024  14:06      150,541 yarn.lock
                3 File(s)      151,495 bytes
                4 Dir(s)  19,728,130,048 bytes free

C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>
```

I created an empty Dockerfile and I opened it using Notepad.

```
                3 File(s)      151,495 bytes
                4 Dir(s)  19,728,130,048 bytes free

C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>echo. > Dockerfile
C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>notepad Dockerfile
C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>
```

I added the following content to the Docker file.



```
File Edit Format View Help
# syntax=docker/dockerfile:1
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```

Ln 8, Col 12 100% Windows (CRLF) UTF-8

bar\2nd Semester\Cloud Data Centres\Lab3\getting-started-app> Dockerfile

I then used this command to build the image.

```
C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app> docker build -t getting-started .
2024/01/24 14:11:15 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 23.6s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 191B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> resolve image config for docker.io/docker/dockerfile:1
=> [auth] docker/dockerfile:pull token for registry-1.docker.io
=> CACHED docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcfa89046420e1781752bab202122f8f50032edf31be0021
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:bla0356f7d6b86c958a06949d3db3f7fb27f95f627aa6157cb98bc65c001efa2
=> [internal] load build context
=> => transferring context: 6.50MB
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN yarn install --production
=> exporting to image
=> => exporting layers
=> => writing image sha256:320d6635d66668809266a8ac779063642c63703a71664acf779b7d043a992565
=> => naming to docker.io/library/getting-started

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
```

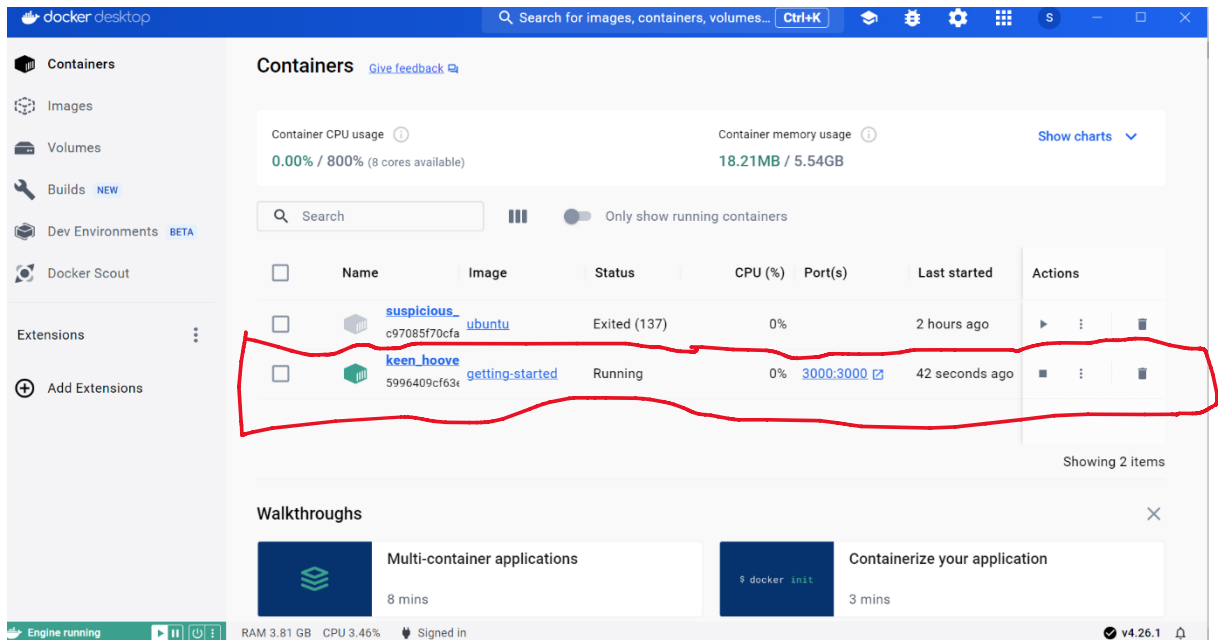
I then ran the container using this command.

```
What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

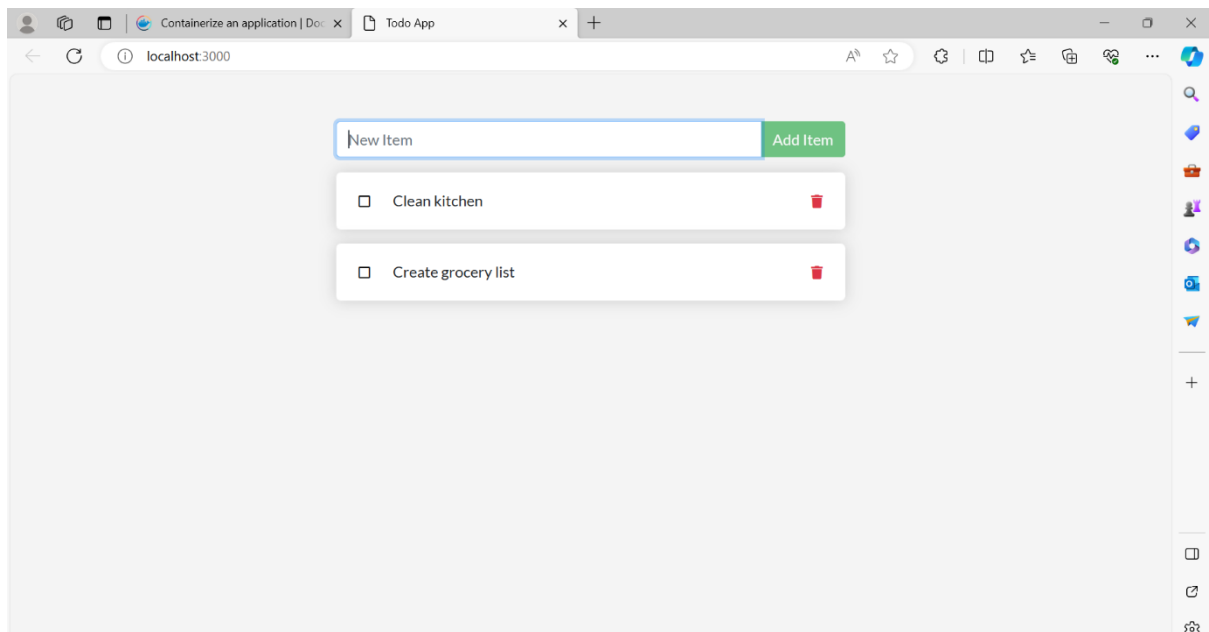
C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app> docker run -dp 127.0.0.1:3000:3000 getting-started
5996409cf63e63ecff0794a4df88d3448f1525e67e918ed7295705d5d741f923

C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>
```

I now have this container in Docker running.



I now have this in my web browser <http://localhost:3000> and I have added some new tasks.



I ran the following command in the terminal to list my containers

```
C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
5996409cf63e   getting-started  "docker-entrypoint.s..."  2 minutes ago  Up 2 minutes  127.0.0.1:3000->3000/tcp  keen_hoover
C:\Users\sarah\4th Year\2nd Semester\Cloud Data Centres\Lab3\getting-started-app>
```

## Part 3: Update The Application

I edited the src/static/js/app.js file to update line 56 to include this.

```
return (
  <React.Fragment>
    <AddItemForm onNewItem={onNewItem} />
    {items.length === 0 && (
      <p className="text-center">You have no todo items yet! Add one above!</p>
    )}
    {items.map(item => (
      <ItemDisplay
        item={item}
        key={item.id}
        onItemUpdate={onItemUpdate}
        onItemRemoval={onItemRemoval}
      />
    ))}
  )}
```

I then built the updated version of the image using the following command

```
Command Prompt
C:\Users\sarah\getting-started-app\src\static>cd ..
C:\Users\sarah\getting-started-app\src>cd ..
C:\Users\sarah\getting-started-app>docker build -t getting-started .
[+] Building 24.7s (13/13) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 191B                                0.0s
=> [internal] load .dockerignore                                    0.0s
=> => transferring context: 2B                                       0.0s
=> resolve image config for docker.io/docker/dockerfile:1        2.3s
=> [auth] docker/dockerfile:pull token for registry-1.docker.io   0.0s
=> CACHED docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f50 0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine  0.9s
=> [auth] library/node:pull token for registry-1.docker.io        0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:b1a0356f7d6b86c958a06949d3db3f7fb27f95f627aa6157cb98bc65c8 0.0s
=> [internal] load build context                                    0.8s
=> => transferring context: 6.50MB                                    0.8s
=> CACHED [2/4] WORKDIR /app                                        0.0s
=> [3/4] COPY . .                                                  0.2s
=> [4/4] RUN yarn install --production                             18.9s
=> exporting to image                                              1.3s
=> => exporting layers                                              1.3s
=> => writing image sha256:46e75777ddac88f8a72b3b8ab79882cadf156d1fc7cd75ad076bf8add626b3bd 0.0s
=> => naming to docker.io/library/getting-started                  0.0s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
C:\Users\sarah\getting-started-app>
```

I then started a new container using the updated code and got the following error.

```
> -> naming to docker.io/library/getting-started 0.0.0
What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

C:\Users\sarah\getting-started-app>docker run -dp 127.0.0.1:3000:3000 getting-started
64c4ee7e433bcd4aa5eeb9e4a75d4b1da978fb292a8c947986353ab32043a25c
docker: Error response from daemon: driver failed programming external connectivity on endpoint beautiful_swirles (924f0b0feee15a3359a3c44b0810bc7c8092d995d2cbc24e33fada4c3bfd221): Bind for 127.0.0.1:3000 failed: port is already allocated

C:\Users\sarah\getting-started-app>
```

In order to fix the error I need to remove the old container.

First I got the ID of the container

```
C:\Users\sarah\getting-started-app>docker stop 5996409cf63e
5996409cf63e
```

And then I stopped it from running.

```
C:\Users\sarah\getting-started-app>docker stop 5996409cf63e
5996409cf63e
```

I then removed the container.

```
C:\Users\sarah\getting-started-app>docker rm 5996409cf63e
5996409cf63e

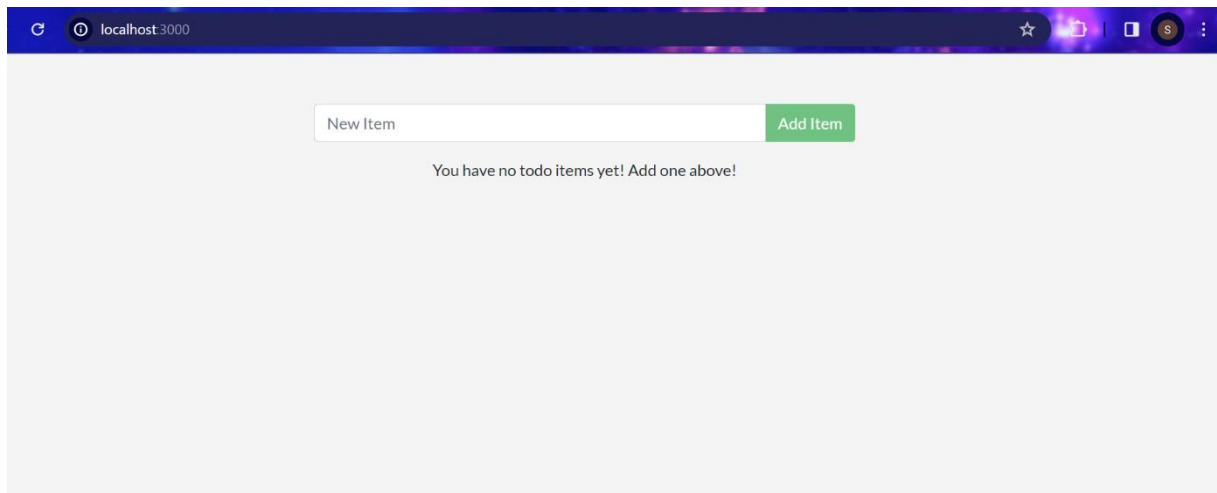
C:\Users\sarah\getting-started-app>
```

I ran this command to then start my updated app.



```
C:\Users\sarah\getting-started-app>docker run -dp 127.0.0.1:3000:3000 getting-started
160e0b99441fc6e25ed87afbc1cb98b84fc9e8062900ed33acd8a8d1575743ccb

C:\Users\sarah\getting-started-app>
```

My web browser <http://localhost:3000> is now updated to this.



And this is my now running container.

<input type="checkbox"/>	 <b>beautiful_sl</b> 64c4ee7e433 <a href="#">getting-started</a>	Created (128)	0%	3000:3000		
<input type="checkbox"/>	 <b>pensive_col</b> 60e0b99441f <a href="#">getting-started</a>	Running	0%	<a href="#">3000:3000</a>		

Showing 3 items

## Part 5: Persist the DB

I started a ubuntu container that created a file name and a number between 1 and 1000

```
C:\Users\sarah\getting-started-app>docker run -d ubuntu bash -c "shuf -i 1-10000 -n 1 -o /data.txt && tail -f /dev/null"
97e19b03a111115e27510109f949972b926c9d9cd459f7bad7159832da346ee4
C:\Users\sarah\getting-started-app>
```

Used the docker exec command to access the container

```
C:\Users\sarah\getting-started-app>docker exec 97e19b03a111115e27510109f949972b926c9d9cd459f7bad7159832da346ee4 cat /data.txt
1571
C:\Users\sarah\getting-started-app>
```

Started a new ubuntu container and I don't have the same file. I used this command to get the following content.

```
C:\Users\sarah\getting-started-app>docker run -it ubuntu ls /  
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var  
C:\Users\sarah\getting-started-app>
```

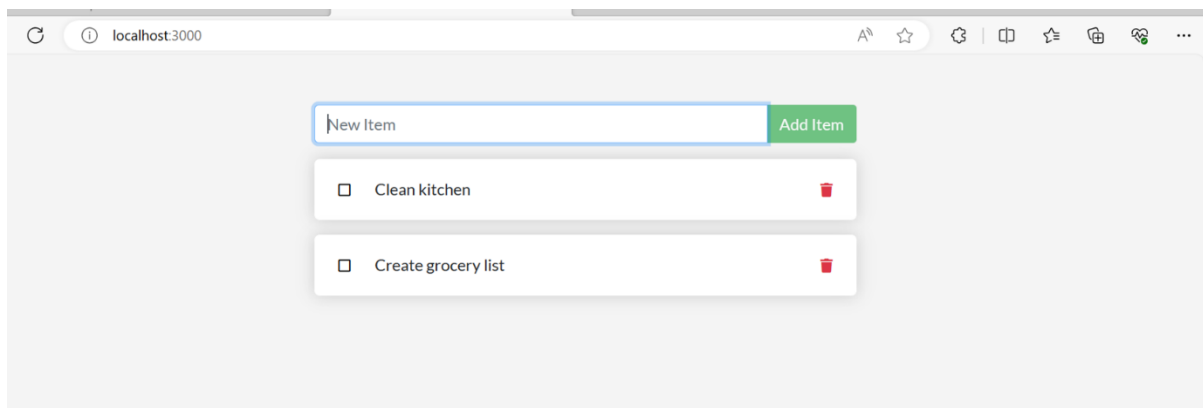
Created a volume.

```
C:\Users\sarah\getting-started-app>docker volume create todo-db  
todo-db  
  
C:\Users\sarah\getting-started-app>
```

Started the todo app container and specified a volume amount.

```
C:\Users\sarah\getting-started-app>docker run -dp 127.0.0.1:3000:3000 --mount type=volume,src=todo-db,target=/etc/todos getting-started  
a8ff6a45635569d72f45e75b000cf69edf6e64654033c34d89ff8e6846f39bd5  
C:\Users\sarah\getting-started-app>
```

Added some items to the todo list.





I used docker ps and then removed the container for the todo app.

```
C:\Users\sarah\getting-started-app>docker rm -f a8ff6a45635569d72f45e75b000cf69edf6e64654033c34d89ff8e6846f39bd5
a8ff6a45635569d72f45e75b000cf69edf6e64654033c34d89ff8e6846f39bd5

C:\Users\sarah\getting-started-app>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES

C:\Users\sarah\getting-started-app>docker run -dp 127.0.0.1:3000:3000 --mount type=volume,src=todo-db,target=/etc/todos getting-started
438f958043eb7df7bbb06efb690e7a9ec3d3d2351500f368935022ecd4e86c37

C:\Users\sarah\getting-started-app>docker rm -f 438f958043eb7df7bbb06efb690e7a9ec3d3d2351500f368935022ecd4e86c37
438f958043eb7df7bbb06efb690e7a9ec3d3d2351500f368935022ecd4e86c37

C:\Users\sarah\getting-started-app>
```

I inspected the docker data

```
C:\Users\sarah\getting-started-app>docker volume inspect todo-db
[
  {
    "CreatedAt": "2024-01-29T19:49:32Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/todo-db/_data",
    "Name": "todo-db",
    "Options": null,
    "Scope": "local"
  }
]

C:\Users\sarah\getting-started-app>
```

## Part 6: Use Bind Mounts

Ran this command to start bash in an Ubuntu container with a bind amount. I then did an ls to list the containers file systems. I then changed to the src directory and created a file called “myfile.txt” here.

```
C:\Users\sarah\Downloads\getting-started-app>docker run -it --mount "type=bind,src=%cd%,target=/src" ubuntu bash
root@10623c3ce897:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  src  sys  usr
boot  etc  lib  lib64  media  opt  root  sbin  srv  tmp  var
root@10623c3ce897:/# cd src
root@10623c3ce897:/src# ls
Dockerfile  README.md  package.json  spec  src  yarn.lock
root@10623c3ce897:/src# touch myfile.txt
root@10623c3ce897:/src#
```

I then observed that the myfile.txt was created in the getting-started-app directory.

This PC > Downloads > getting-started-app	
Name	Date modified
spec	31/01/2024 10:22
src	31/01/2024 10:22
Dockerfile	29/01/2024 18:00
myfile.txt	31/01/2024 10:30
package.json	29/01/2024 17:55
README.md	29/01/2024 17:55
yarn.lock	29/01/2024 17:55







And then I proceeded to delete the myfile.txt file.

Share

view

> This PC > Downloads > getting-started-app

^

Name	Date modified	Type	Size
 spec	31/01/2024 10:22	File folder	
 src	31/01/2024 10:22	File folder	
 Dockerfile	29/01/2024 18:00	File	1 KB
 package.json	29/01/2024 17:55	JSON File	1 KB
 README.md	29/01/2024 17:55	Markdown Source File	1 KB
 yarn.lock	29/01/2024 17:55	LOCK File	148 KB

I then checked to confirm the file was deleted in the app directory.

```
root@10623c3ce897:/src# touch myfile.txt
root@10623c3ce897:/src# ls
Dockerfile  README.md  package.json  spec  src  yarn.lock
root@10623c3ce897:/src#
```

I think stopped the container session with ctrl + D.

I ran this command in Powershell.

```
root@10623c3ce897:/src# exit
exit

C:\Users\sarah\Downloads\getting-started-app>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\sarah\Downloads\getting-started-app> docker run -dp 127.0.0.1:3000:3000 `
>> -w /app --mount "type=bind,src=$pwd,target=/app" `
>> node:18-alpine `
>> sh -c "yarn install && yarn run dev"
Unable to find image 'node:18-alpine' locally
18-alpine: Pulling from library/node
4abcf2066143: Already exists
eb6c7c29ba4d: Already exists
3d4a65156edf: Already exists
5bdb6c27eb32: Already exists
Digest: sha256:0085670310d2879621f96a4216c893f92e2ded827e9e6ef8437672e1bd72f437
Status: Downloaded newer image for node:18-alpine
994c71efaeae6e858f8f418a04a92e9f21a09552d78689f9c7d6bccf8edc8369
PS C:\Users\sarah\Downloads\getting-started-app>
```

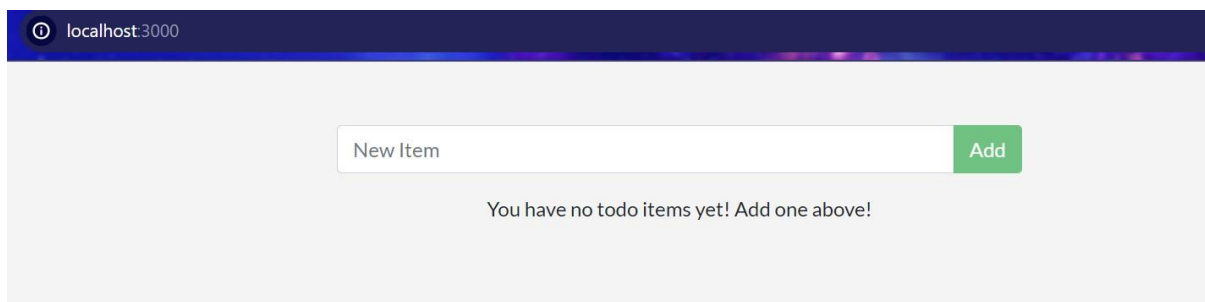
I then checked the logs.

```
PS C:\Users\sarah\Downloads\getting-started-app> docker logs -f 994c71efaeae6e858f8f418a04a92e9f21a09552d78689f9c7d6bccf8edc8369
yarn install v1.22.19
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
Done in 69.08s.
yarn run v1.22.19
$ 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
```

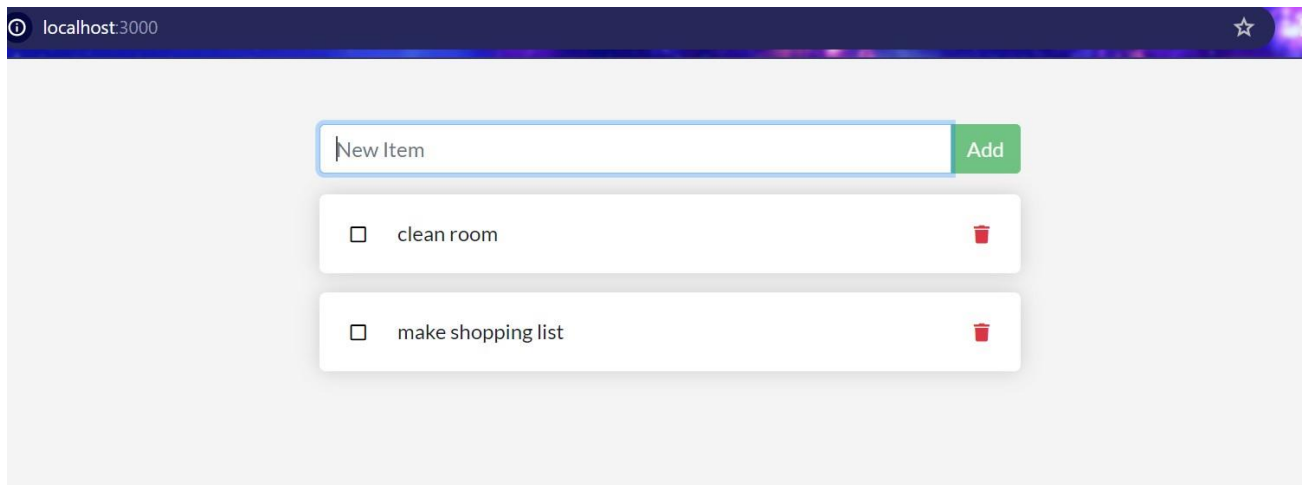
I then changed the line in src/static/js/app.js on line 109. It changes the add item button to just a simple add.

```
107         className={submitting ? 'disabled' : ''}
108       >
109       {submitting ? 'Adding...' : 'Add'}
110     </Button>
```

The new changes due to the bind amount.



I added in some new items to the list by adding two new items.



I stopped the container.

```
C:\Users\sarah\Downloads\getting-started-app>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
994c71efaeae   node:18-alpine "docker-entrypoint.s..." About an hour ago Up About an hour 127.0.0.1:3000->3000/tcp
busy_gates

C:\Users\sarah\Downloads\getting-started-app>docker stop 994c71efaeae
994c71efaeae

C:\Users\sarah\Downloads\getting-started-app>
```

And then built my new image.

```
C:\Users\sarah\Downloads\getting-started-app>docker build -t getting-started .
[+] Building 4.7s (8/11)                                docker:default
=> [internal] load build definition from Dockerfile      0.1s
=> => transferring dockerfile: 188B                     0.0s
=> [internal] load .dockerignore                        0.0s
=> => transferring context: 2B                           0.0s
=> resolve image config for docker.io/docker/dockerfile:1 4.2s
=> [auth] docker/dockerfile:pull token for registry-1.docker.io 0.0s
=> CACHED docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f50 0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine 0.0s
=> [1/4] FROM docker.io/library/node:18-alpine          0.0s
=> CANCELED [internal] load build context              0.1s
=> => transferring context: 5.16kB                      0.0s
ERROR: failed to solve: Canceled: context canceled

View build details: docker-desktop://dashboard/build/default/default/931zzgi9fiev1lu0lut24lbt5

C:\Users\sarah\Downloads\getting-started-app>
```

## Part 7: Multi Container Apps

Created the network.

```
C:\Users\sarah\Downloads\getting-started-app>docker network create todo-app
796255b2ec62dd7aa73c9f015e9c2503846842fc84bcb656bd1a9fed36bf88d0

C:\Users\sarah\Downloads\getting-started-app>
```



I started a SQL container and attached it to the network.

```
C:\Users\sarah\Downloads\getting-started-app>docker run -d ^
More? --network todo-app --network-alias mysql ^
More? -v todo-mysql-data:/var/lib/mysql ^
More? -e MYSQL_ROOT_PASSWORD=secret ^
More? -e MYSQL_DATABASE=todos ^
More? mysql:8.0
Unable to find image 'mysql:8.0' locally
8.0: Pulling from library/mysql
558b7d69a2e5: Pull complete
c7e714ea5470: Pull complete
508ada0981f6: Pull complete
1ae11d1b9d69: Pull complete
3c9be74ddc84: Pull complete
d4a59c718252: Pull complete
ac42afee6c9c: Pull complete
dcf1f586b2e2: Pull complete
210e58b61fa3: Pull complete
d9719ad703de: Pull complete
1ef92b20c8ec: Pull complete
Digest: sha256:3f75dccc64fffa40a06a4a9256206280a5ddc3e26dea3f1ab0df35b2cc12f472
Status: Downloaded newer image for mysql:8.0
e271aa1c8434454e01a46f8295d163782799b676bfee424de6c476093c6a91fa
C:\Users\sarah\Downloads\getting-started-app>
```

I verified that my database was up and running.

```
C:\Users\sarah\Downloads\getting-started-app>docker exec -it e271aa1c8434454e01a46f8295d163782799b676bfee424de6c476093c6a91fa mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, 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>
```

I used the show databases command.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| todos |
+-----+
5 rows in set (0.01 sec)

mysql>
```

And the exited SQL.

```
8a49fdb3b6a5: Downloading [=====>] 593.1kB/3.397MB
Bye
```

I then started a new container using the nicolaka/netshoot image. I then used dig mysql to look up the IP address for the hostname mysql.

```
Welcome to Netshoot! (github.com/nicolaka/netshoot)
Version: 0.11

7d366e1bd8b5 [root@~]# dig mysql

; <<>> DiG 9.18.13 <<>> mysql
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50309
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;mysql.                IN      A

;; ANSWER SECTION:
mysql.                600     IN      A      172.18.0.2

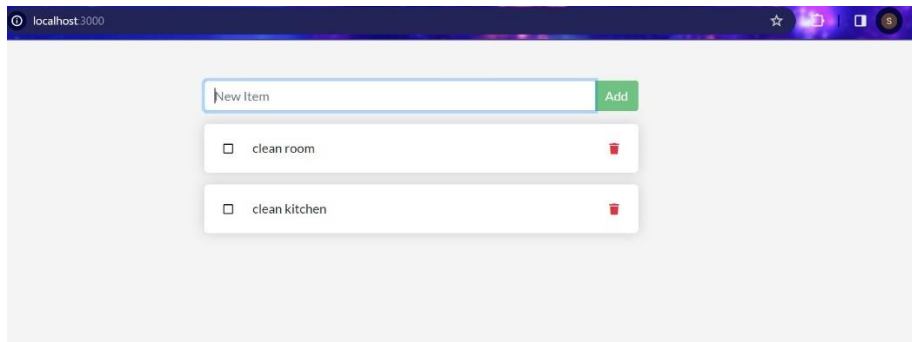
;; Query time: 10 msec
;; SERVER: 127.0.0.11#53(127.0.0.11) (UDP)
;; WHEN: Thu Feb 01 09:07:10 UTC 2024
;; MSG SIZE rcvd: 44

7d366e1bd8b5 [root@~]#
```

I then started the container and connected the container to the app network.

```
C:\Users\sarah\Downloads\getting-started-app>docker run -dp 127.0.0.1:3000:3000 ^
More? -w /app -v "%cd%:/app" ^
More? --network todo-app ^
More? -e MYSQL_HOST=mysql ^
More? -e MYSQL_USER=root ^
More? -e MYSQL_PASSWORD=secret ^
More? -e MYSQL_DB=todos ^
More? node:18-alpine ^
More? sh -c "yarn install && yarn run dev"
2ad13d2ee28c6f894b39728c87298fc7bdfcfc72a678a4b92de1de02b709baf0
C:\Users\sarah\Downloads\getting-started-app>
```

I then added a few items to the todo list.



I then connected to the mysql database to see the items being written to the database.

```
C:\Users\sarah\getting-started-app>docker exec -it db6b3e8624bf 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: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, 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.

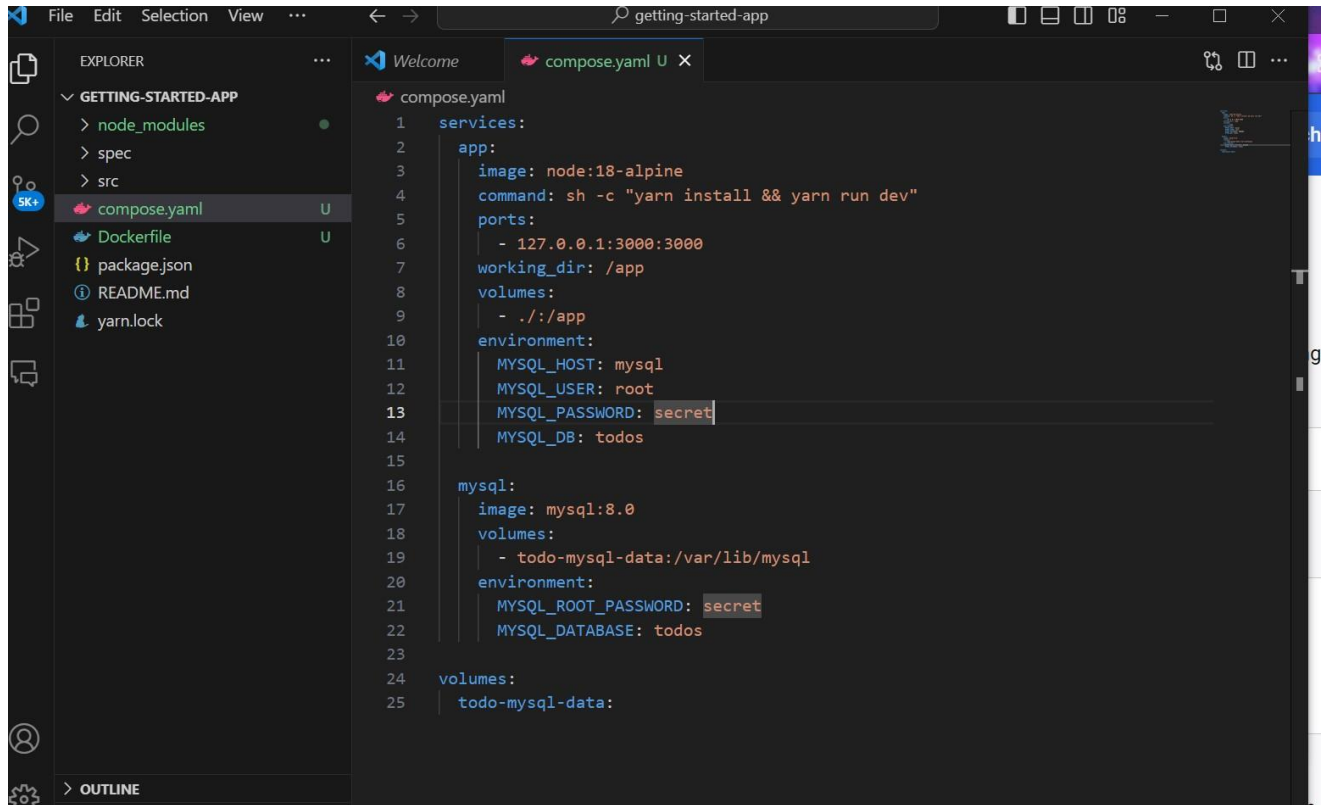
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from todo_items
-> ;
+-----+-----+-----+
| id                | name      | completed |
+-----+-----+-----+
| d78b24db-27ed-4157-8a5f-0fc2f13e6c3c | clean room | 0         |
| d3745b3d-a042-4b8d-b5f6-4c0ab1dab0e3 | clean kitchen | 0         |
+-----+-----+-----+
2 rows in set (0.01 sec)
```



## Part 8: Use Docker Compose

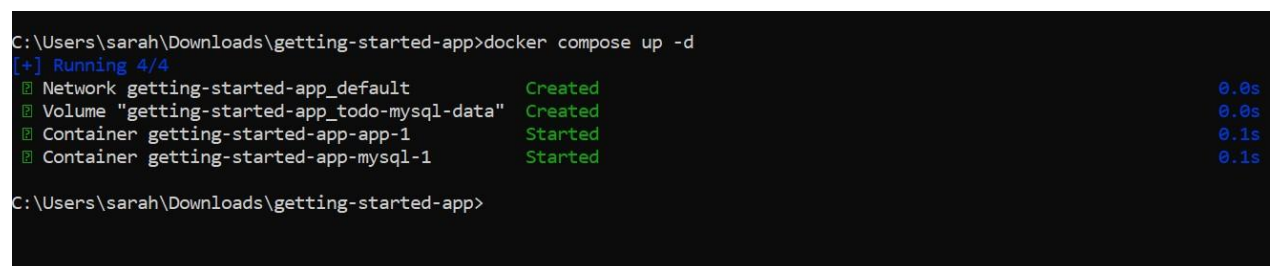
I created a file called `compose.yaml` inside the `getting-started-app` and added all of the code to it.



The screenshot shows the Visual Studio Code interface with the `getting-started-app` project open. The Explorer sidebar on the left shows the file structure: `GETTING-STARTED-APP` containing `node_modules`, `spec`, `src`, `compose.yaml` (selected), `Dockerfile`, `package.json`, `README.md`, and `yarn.lock`. The main editor displays the content of `compose.yaml`:

```
1 services:
2   app:
3     image: node:18-alpine
4     command: sh -c "yarn install && yarn run dev"
5     ports:
6       - 127.0.0.1:3000:3000
7     working_dir: /app
8     volumes:
9       - ./:/app
10    environment:
11      MYSQL_HOST: mysql
12      MYSQL_USER: root
13      MYSQL_PASSWORD: secret
14      MYSQL_DB: todos
15
16  mysql:
17    image: mysql:8.0
18    volumes:
19      - todo-mysql-data:/var/lib/mysql
20    environment:
21      MYSQL_ROOT_PASSWORD: secret
22      MYSQL_DATABASE: todos
23
24  volumes:
25    todo-mysql-data:
```

Now that I have all the code in the `compose.yaml` file I can now start the application.



The screenshot shows a terminal window with the following output:

```
C:\Users\sarah\Downloads\getting-started-app>docker compose up -d
[+] Running 4/4
  Network getting-started-app_default          Created           0.0s
  Volume "getting-started-app_todo-mysql-data" Created           0.0s
  Container getting-started-app-app-1          Started           0.1s
  Container getting-started-app-mysql-1        Started           0.1s
C:\Users\sarah\Downloads\getting-started-app>
```

I then looked at the logs to see the logs from each services interleaved into a single stream.

```
C:\Users\sarah\Downloads\getting-started-app>docker compose logs -f
mysql-1 2024-02-01 10:06:19+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.36-1.el8 started.
mysql-1 2024-02-01 10:06:19+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
mysql-1 2024-02-01 10:06:19+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.36-1.el8 started.
mysql-1 2024-02-01 10:06:19+00:00 [Note] [Entrypoint]: Initializing database files
app-1    yarn install v1.22.19
mysql-1 2024-02-01T10:06:19.631952Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use S
GLOBAL host_cache_size=0 instead.
mysql-1 2024-02-01T10:06:19.632042Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.36) initializing of server in progress as process 80
app-1    [1/4] Resolving packages...
app-1    success Already up-to-date.
app-1    Done in 0.52s.
app-1    yarn run v1.22.19
app-1    $ nodemon -L src/index.js
app-1    [nodemon] 2.0.20
mysql-1 2024-02-01T10:06:19.639402Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql-1 2024-02-01T10:06:20.064153Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql-1 2024-02-01T10:06:21.493944Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider switching off the --initialize-
secure option.
app-1    [nodemon] to restart at any time, enter 'rs'
mysql-1 2024-02-01 10:06:24+00:00 [Note] [Entrypoint]: Database files initialized
app-1    [nodemon] watching path(s): *.*
mysql-1 2024-02-01 10:06:24+00:00 [Note] [Entrypoint]: Starting temporary server
app-1    [nodemon] watching extensions: js,mjs,json
app-1    [nodemon] starting 'node src/index.js'
app-1    Waiting for mysql:3306.....
app-1    Connected!
app-1    Connected to mysql db at host mysql
app-1    Listening on port 3000
mysql-1 2024-02-01T10:06:24.743833Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use S
GLOBAL host_cache_size=0 instead.
mysql-1 2024-02-01T10:06:24.745819Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.36) starting as process 124
mysql-1 2024-02-01T10:06:24.757768Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
mysql-1 2024-02-01T10:06:24.874955Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
mysql-1 2024-02-01T10:06:25.169971Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
```

I then used this command to tear it all down. The containers will stop, and the network will be removed.

```
C:\Users\sarah\Downloads\getting-started-app>docker compose down
[+] Running 3/3
  Container getting-started-app-mysql-1   Removed
  Container getting-started-app-app-1     Removed
  Network getting-started-app_default     Removed

C:\Users\sarah\Downloads\getting-started-app>
```