

Using Docker Compose

```
image: mysql:8.0
volumes:
  - todo-mysql-data:/var/lib/mysql
environment:
  MYSQL_ROOT_PASSWORD: secret
  MYSQL_DATABASE: todos
```

```
volumes:
  todo-mysql-data:
```

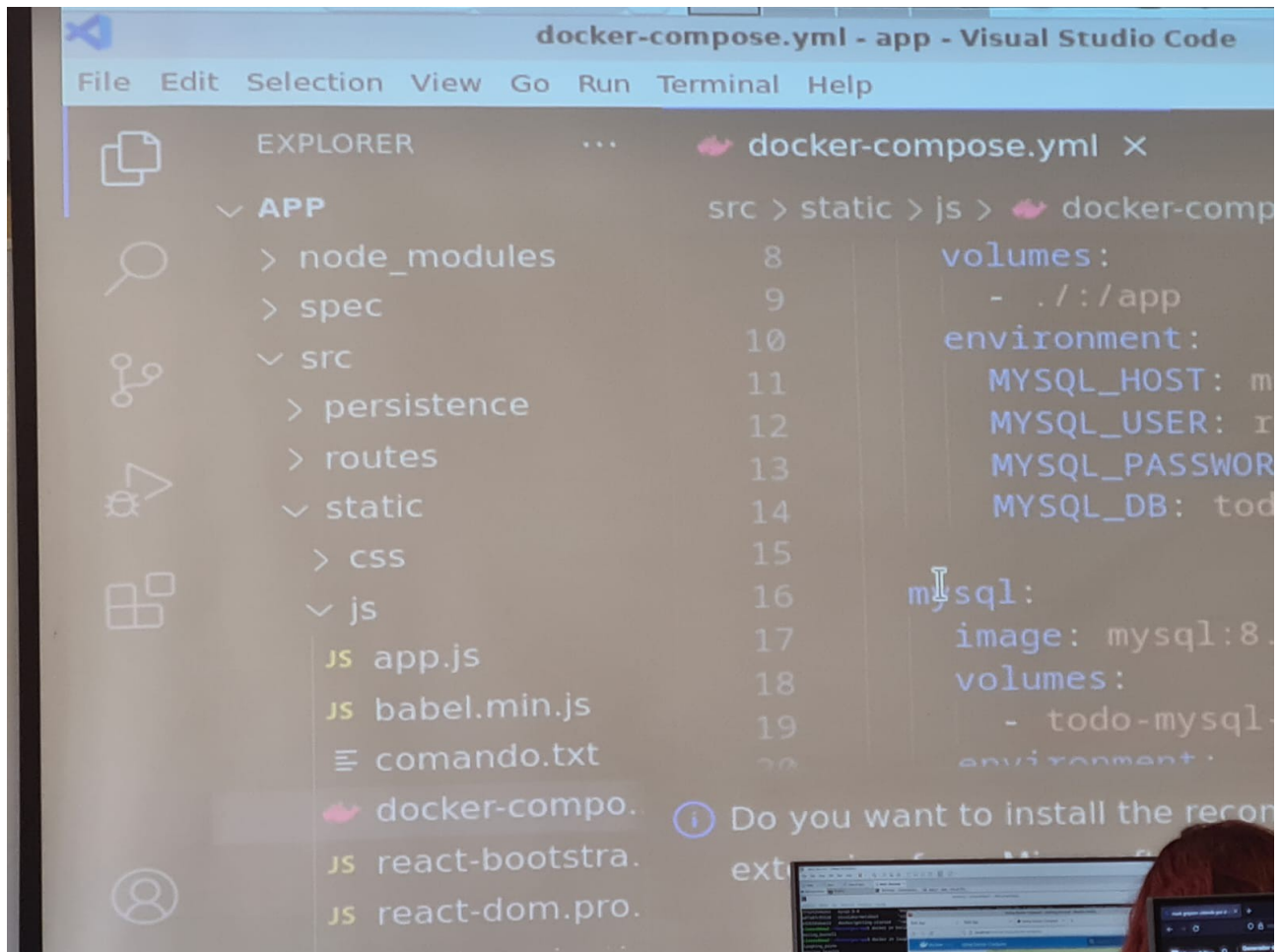
At this point, our complete `docker-compose.yml` should look like this:

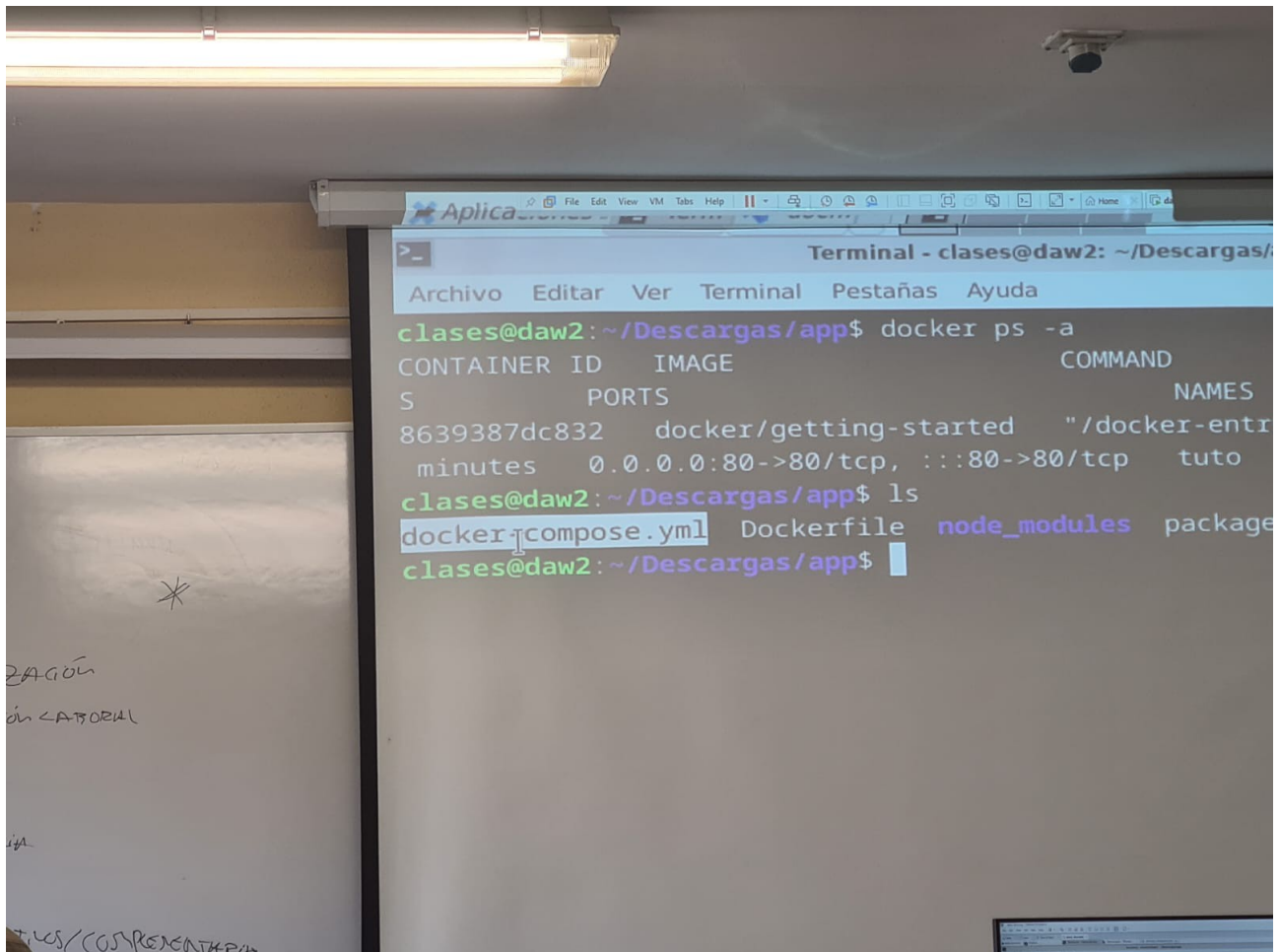
```
services:
  app:
    image: node:16-alpine
    command: sh -c "yarn install && yarn run dev"
    ports:
      - 3000:3000
    working_dir: /app
    volumes:
      - ./:/app
    environment:
      MYSQL_HOST: mysql
      MYSQL_USER: root
      MYSQL_PASSWORD: secret
      MYSQL_DB: todos
```

ENTERED

mysql:8.0

Entrar en el visual studio code en la carpeta Descargas/app y meter ahí dentro el docker-compose de la anterior captura.





clases@daw2:~/Descargas/app\$ docker ps -a

CONTAINER ID	IMAGE	COMMAND	CR
S	PORTS	NAMES	
8639387dc832	docker/getting-started	"/docker-entrypoint...."	16
minutes	0.0.0.0:80->80/tcp, :::80->80/tcp	tuto	

clases@daw2:~/Descargas/app\$ ls

docker-compose.yml Dockerfile node_modules package.json spec

clases@daw2:~/Descargas/app\$ docker compose up -d

Aplica

Terminal - clases@daw2: ~/Descargas/app

Archivo Editar Ver Terminal Pestañas Ayuda

```
clases@daw2:~/Descargas/app$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	STATUS	PORTS	NAMES
8639387dc832	docker/getting-started	"/docker-entrypoint	minutes	0.0.0.0:80->80/tcp, :::80->80/tcp	tuto

```
clases@daw2:~/Descargas/app$ ls
```

docker-compose.yml Dockerfile node_modules package.json

```
clases@daw2:~/Descargas/app$ docker compose up -d
```

[+] Running 4/4

- ✓ Network app_default Created
- ✓ Volume "app_todo-mysql-data" Created
- ✓ Container app-mysql-1 Started
- ✓ Container app-app-1 Started

```
clases@daw2:~/Descargas/app$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	STATUS	PORTS	NAME
14d8a261f58a	mysql:8.0	"docker-entrypoint	seconds	3306/tcp, 33060/tcp	ap
67016775c07d	node:18-alpine		seconds	0.0.0.0:3000->3000/tcp,	
8639387dc832	docker/getting-started		s	0.0.0.0:80->80/tcp, :::8	

daw2:~/Descargas/app\$

