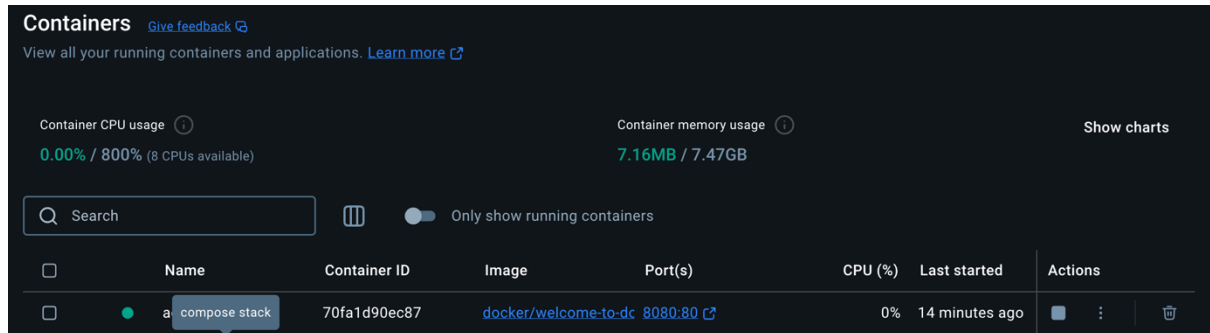


## Lab 2 – Cloud Data Centres

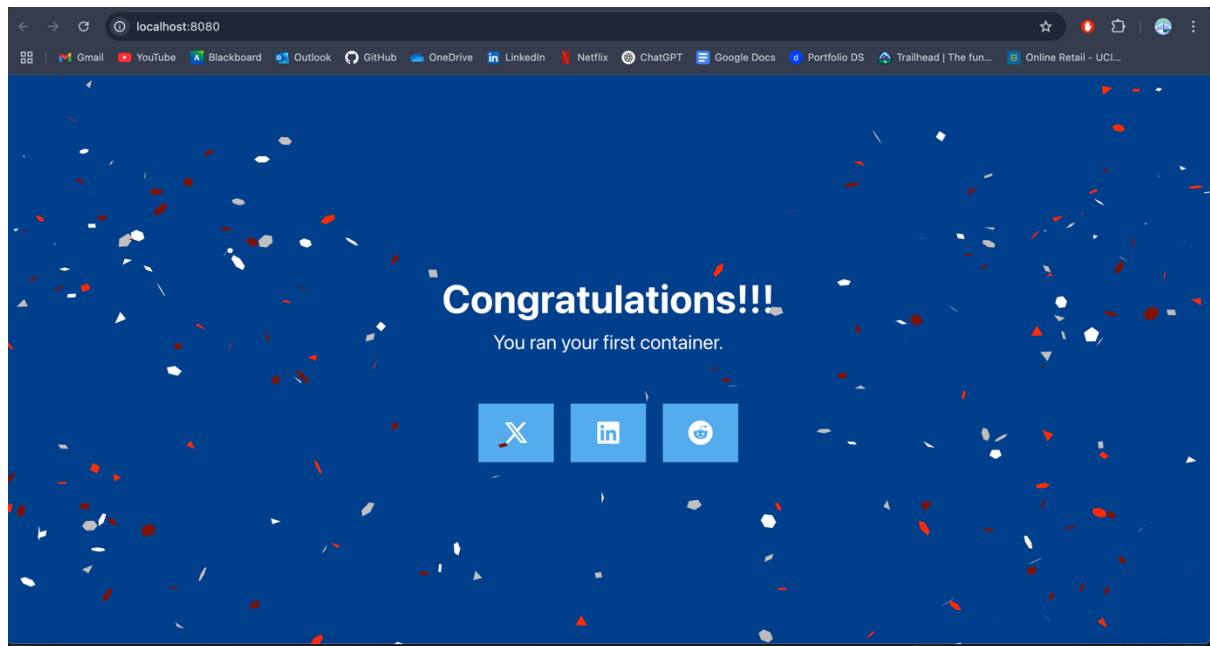
### Publishing and exposing ports:

I installed Docker Desktop using the link from their docs, once installed I ran the following command: `docker run -d -p 8080:80 docker/welcome-to-docker` to set up my container in the application

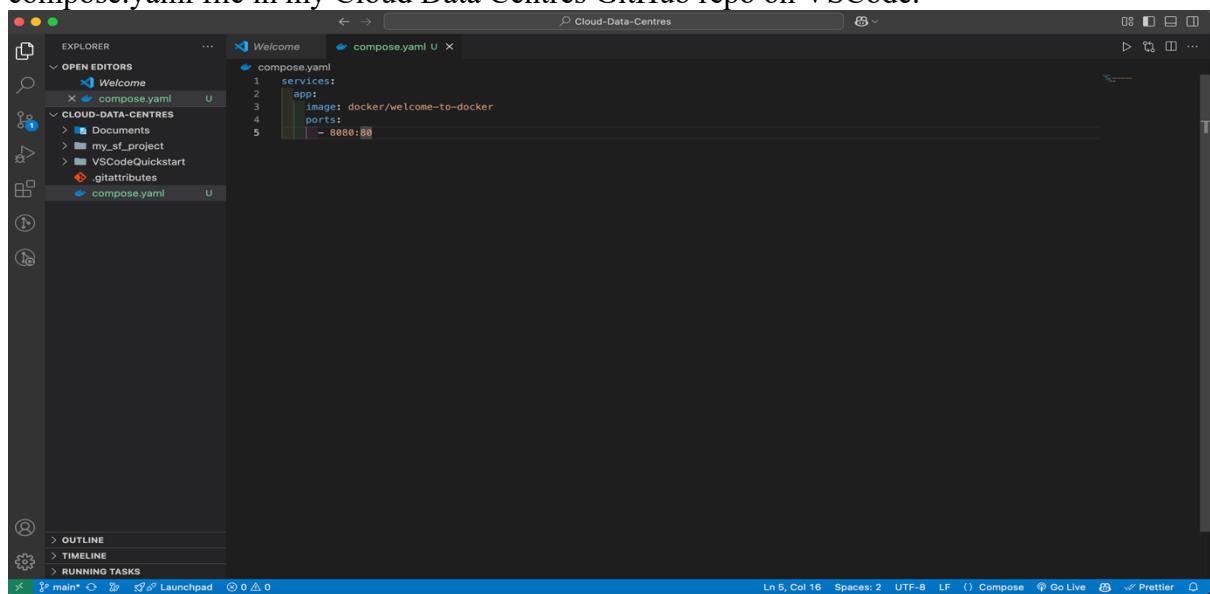


Then I was able to run the localhost server on Google Chrome by clicking the link given to me.

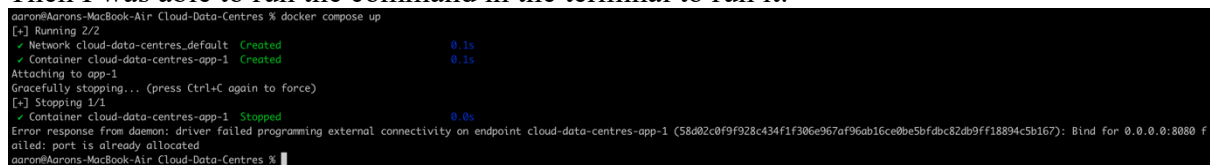
```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run -d -p 8080:80 docker/welcome-to-docker
Unable to find image 'docker/welcome-to-docker:latest' locally
latest: Pulling from docker/welcome-to-docker
89578ce72c35: Download complete
1c2214f9937c: Download complete
d11a451e6399: Download complete
54b19e12c655: Download complete
b42a2f288f4d: Download complete
94be7e780731: Download complete
1fb28e078240: Download complete
579b34f0a95b: Download complete
Digest: sha256:eedaff45e3c78538087bdd9dc7afafac7e110061bbdd836af4104b10f10ab693
Status: Downloaded newer image for docker/welcome-to-docker:latest
```



The next step is then to use Docker compose to do the same thing so I created the `compose.yaml` file in my Cloud Data Centres GitHub repo on VSCode.



Then I was able to run the command in the terminal to run it.



## Overriding Container Defaults:

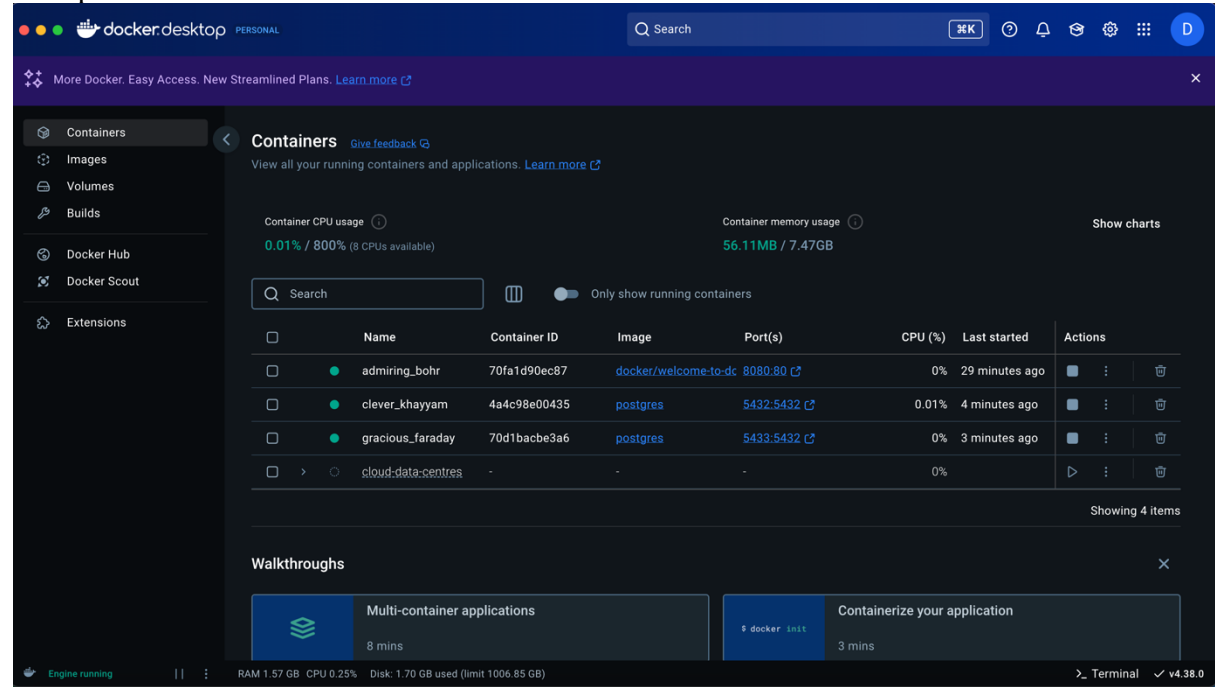
The first step to this one was to run this command through the terminal to start a container using the postgres image.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run -d -e POSTGRES_PASSWORD=secret -p 5432:5432 postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
8c21e9d8fa8c: Download complete
f45b4b1e09b9: Download complete
cf8078a8a245: Download complete
774c54d2d309: Download complete
7b8417b4134b: Download complete
1cfbe96df98f: Download complete
77b1b7588421: Download complete
d51c377d94da: Download complete
a1dedce88ca1: Download complete
434de25e58a4: Download complete
1655e767f751: Download complete
405f36104e57: Download complete
551bc275c65f: Download complete
2c6810bb3d75: Download complete
Digest: sha256:81f32a88ec561664634637dd446487efd5f9d90996304b96210078e90e5c8b21
Status: Downloaded newer image for postgres:latest
4a4c98e00435b0a31e6c03b6bb0273bac98635e53b5dcacd6cbdb913db4039ec
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

Then I ran the second command to start the second container mapped to a different port.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run -d -e POSTGRES_PASSWORD=secret -p 5433:5432 postgres
70d1bacbe3a6132b9024623eace44ae13a909c705314073fca7ff2f1488e0312
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

Now as can be seen in the image below of my Docker Desktop, both containers are running as expected.



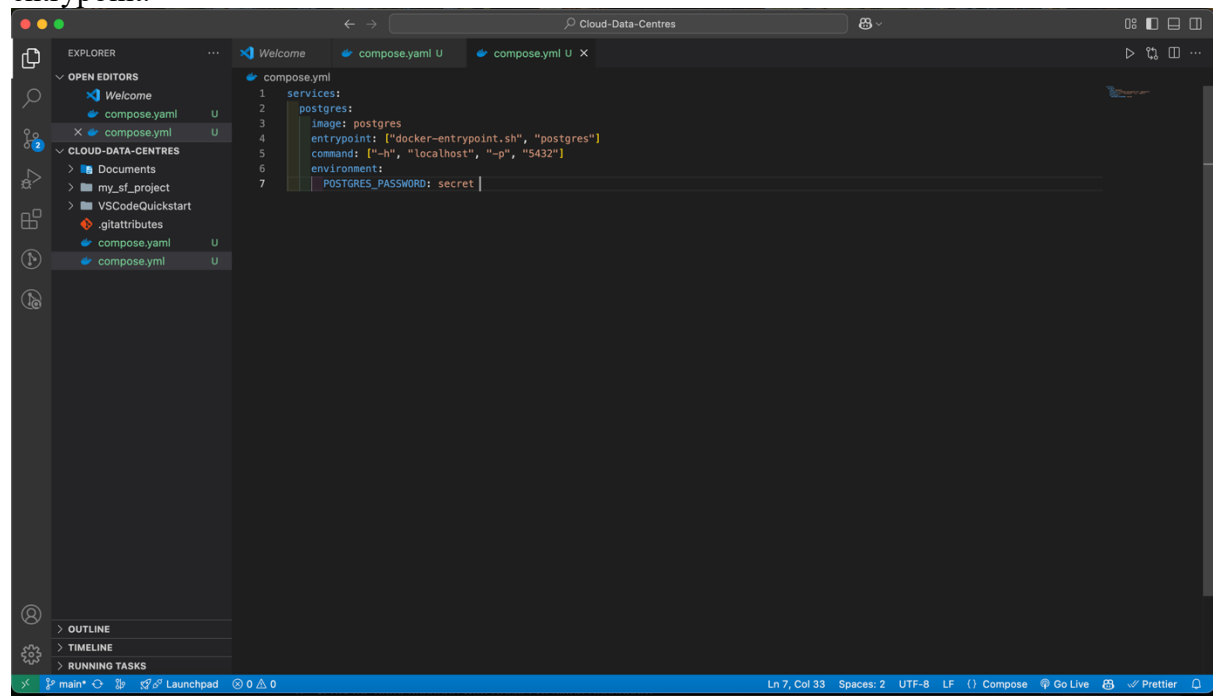
Then I was tasked with connecting a Postgres container to a custom network and I did this by using the following commands:

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker network create mynetwork
512ef3bb5f755bbdb5295a29de740908f431674a82f8e6fda1d4152e3b23dbc9
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
8c648234cdf7        bridge              bridge              local
28d71bbbc0fd        cloud-data-centres_default bridge              local
b107b5b1d76e        host                host                local
512ef3bb5f75        mynetwork           bridge              local
77e395d748c7        none                null                local
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run -d -e POSTGRES_PASSWORD=secret -p 5434:5432 --network mynetwork postgres
9d56db4599cc4860c9e943355b1a43b3164d6a1180fbc51cda805ef17bfd954
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

The next step is then to manage the resources for the containers as they aren't limited by default, this was done by running the following command:

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run -d -e POSTGRES_PASSWORD=secret --memory="512m" --cpus=".5" postgres
daf9105b3a339f4b5be7ec8d25a134195eac6abb868e4715915fe7d854ad120d
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

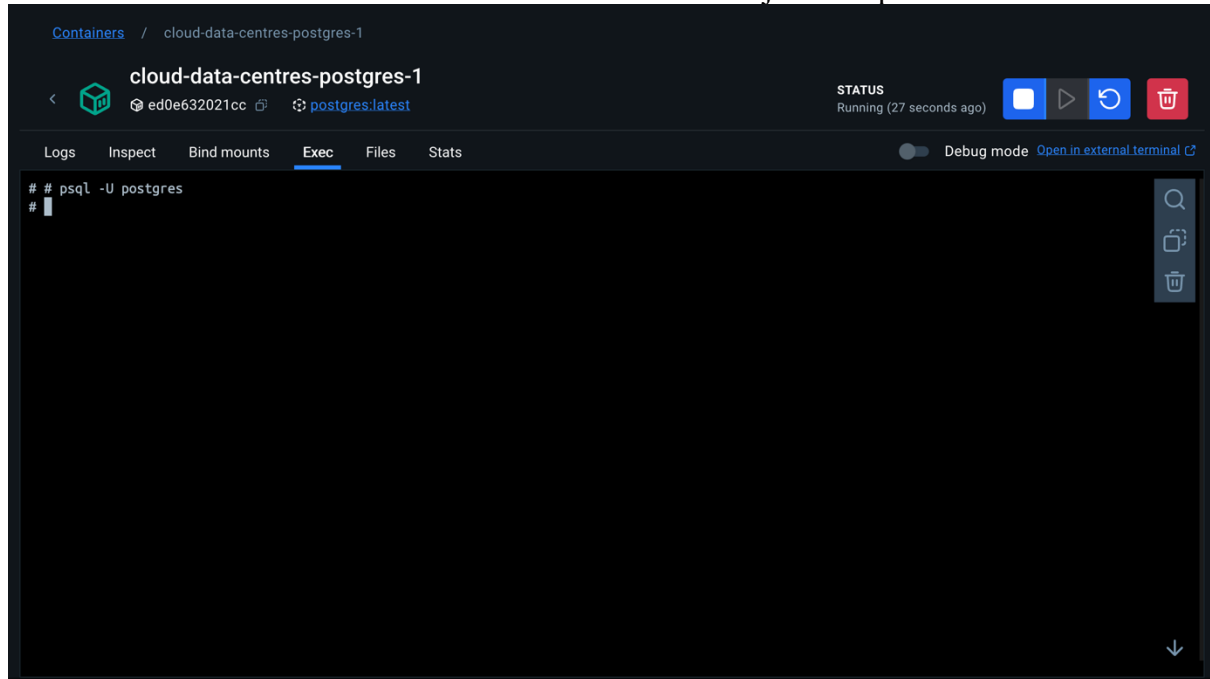
The next step in the guide was to set up a compose.yml file to override the default cmd and entrypoint.



Then I ran the container by using the following command.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker compose up -d
WARN[0000] Found multiple config files with supported names: /Users/aaron/Documents/GitHub/Cloud-Data-Centres/compose.yaml, /Users/aaron/Documents/GitHub/Cloud-Data-Centres/compose.yml
WARN[0000] Using /Users/aaron/Documents/GitHub/Cloud-Data-Centres/compose.yaml
[+] Running 1/1
✔ Container cloud-data-centres-app-1 Started
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

Then I ran the command on the exec tab on the container I just set up.



Then I ran the final command: `docker run -e POSTGRES_PASSWORD=secret postgres docker-entrypoint.sh -h localhost -p 5432` that was given to me in the instructions through my terminal.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run -e POSTGRES_PASSWORD=secret postgres docker-entrypoint.sh -h localhost -p 5432
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /var/lib/postgresql/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default "max_connections" ... 100
selecting default "shared_buffers" ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

    pg_ctl -D /var/lib/postgresql/data -l logfile start

initdb: warning: enabling "trust" authentication for local connections
initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.
waiting for server to start....2025-03-05 16:18:00.592 UTC [48] LOG:  starting PostgreSQL 17.4 (Debian 17.4-1.pgdg120+2) on aarch64-unknown-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
2025-03-05 16:18:00.593 UTC [48] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2025-03-05 16:18:00.595 UTC [51] LOG:  database system was shut down at 2025-03-05 16:18:00 UTC
2025-03-05 16:18:00.597 UTC [48] LOG:  database system is ready to accept connections
done
server started

/usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*

waiting for server to shut down....2025-03-05 16:18:00.715 UTC [48] LOG:  received fast shutdown request
2025-03-05 16:18:00.717 UTC [48] LOG:  aborting any active transactions
2025-03-05 16:18:00.718 UTC [48] LOG:  background worker "logical replication launcher" (PID 54) exited with exit code 1
2025-03-05 16:18:00.718 UTC [49] LOG:  shutting down
2025-03-05 16:18:00.719 UTC [49] LOG:  checkpoint starting: shutdown immediate
2025-03-05 16:18:00.722 UTC [49] LOG:  checkpoint complete: wrote 3 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.002 s, sync=0.001 s, total=0.005 s; sync files=2, longest=0.001 s, average=0.001 s; distance=0 kB, estimate=0 kB; lsn=0/14E4F98, redo lsn=0/14E4F98
2025-03-05 16:18:00.724 UTC [48] LOG:  database system is shut down
done
```

## Persisting Container Data:

I started up the container by running this command in my terminal.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run --name=db -e POSTGRES_PASSWORD=secret -d -v postgres_data:/var/lib/postgresql/data postgres
9d14038f9bb378370876d535f16ff5b9c7104eb9ab271cc4b2d5c6ee8544ea86
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

I then connected to the DB by using this command.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker exec -ti db psql -U postgres
psql (17.4 (Debian 17.4-1.pgdg120+2))
Type "help" for help.

postgres=#
```

ZAP

I then added this table to the database.

```
postgres=# CREATE TABLE tasks (
           id SERIAL PRIMARY KEY,
           description VARCHAR(100)
);
INSERT INTO tasks (description) VALUES ('Finish work'), ('Have fun');
CREATE TABLE
INSERT 0 2
postgres=#
```

I then stopped the database by running the following command.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker stop db
docker rm db
db
db
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

Here I am starting a new container and checking to make sure the database still contains the data I entered earlier.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run --name=new-db -d -v postgres_data:/var/lib/postgresql/data postgres
f658f7dc34eb643ae6359a0628effe9ffe8be914a5cfb5753d5534183baa69ec
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker exec -ti new-db psql -U postgres -c "SELECT * FROM tasks"
 id | description
----+-----
  1 | Finish work
  2 | Have fun
(2 rows)

aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

Here I am removing the db I created earlier.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker rm -f new-db
new-db
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

Here I am clearing the volume and pruning.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker volume rm postgres_data
postgres_data
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker volume prune
WARNING! This will remove anonymous local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
942a66d01df14989b5930ff4e587daf03635f10b421b66ac34e5b8ce7fd64fe2

Total reclaimed space: 40.38MB
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

### Sharing local files with containers:

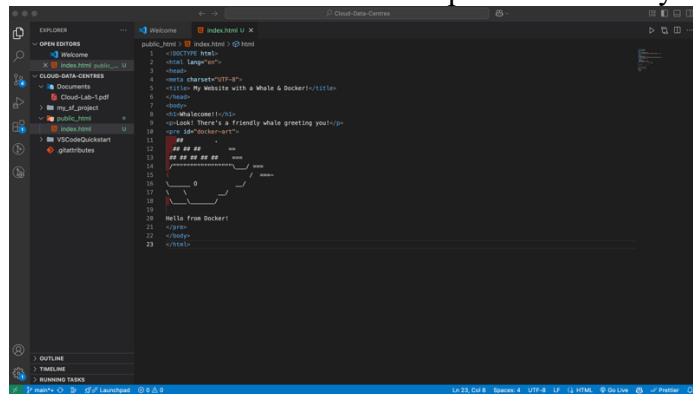
Here I am starting a container using the httpd image.

```
aaron@Aarons-MacBook-Air ~ % docker run -d -p 8080:80 --name my_site httpd:2.4
Unable to find image 'httpd:2.4' locally
2.4: Pulling from library/httpd
c74daaed869f: Download complete
eae886c0d67e: Download complete
d6cc655c251c: Download complete
ccf64fec0567: Download complete
4f4fb700ef54: Download complete
Digest: sha256:10381816bb7e60ae3a9db3784f2966a8910b6ff07c4da54bd2d62d2671c8ab6e
Status: Downloaded newer image for httpd:2.4
34b647454da0678d4c82ce75a29afa102c0c17e4ad53d117f47cc241f7f92322
aaron@Aarons-MacBook-Air ~ %
```

As you can see it's working.

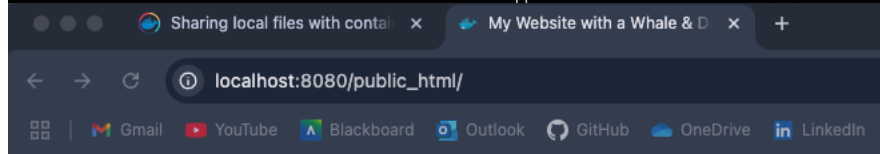
```
aaron@Aarons-MacBook-Air ~ % curl localhost:8080
<html><body><h1>It works!</h1></body></html>
aaron@Aarons-MacBook-Air ~ %
```

I then created the html file in the specified directory.



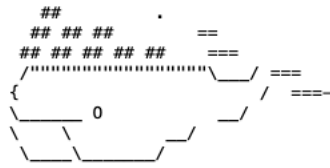
I then used this command to get it running on my localhost server.

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % docker run -d --name my_site -p 8080:80 -v ./usr/local/apache2/htdocs/: httpd:2.4
32559e2f0f26681f493c73463ebfa55ebd03e9a9353a24c75a4966d3a135ace1
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```



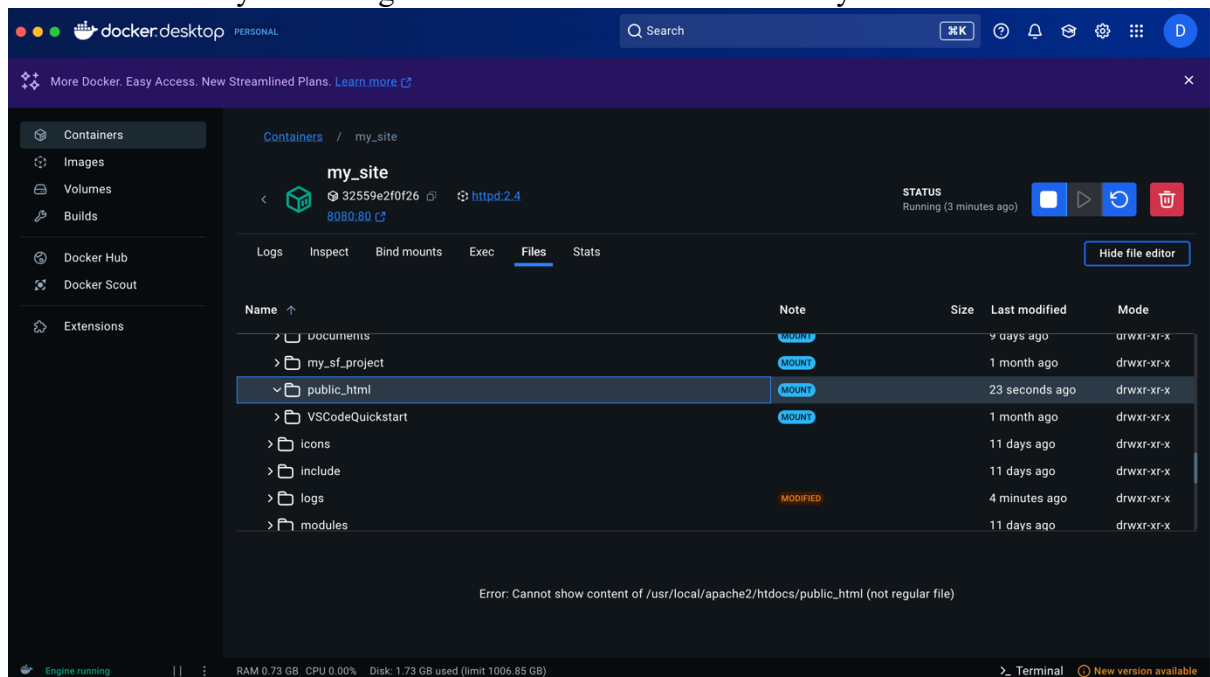
## Whalecome!!

Look! There's a friendly whale greeting you!



Hello from Docker!

When I deleted my file through VSCode it was also deleted in my container.





## Multi-container applications:

Firstly, I am cloning the given git repo

```
aaron@Aarons-MacBook-Air Cloud-Data-Centres % git clone https://github.com/docker-samples/nginx-node-redis
Cloning into 'nginx-node-redis'...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (73/73), done.
remote: Total 78 (delta 25), reused 4 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (78/78), 75.06 KiB | 10.72 MiB/s, done.
Resolving deltas: 100% (25/25), done.
aaron@Aarons-MacBook-Air Cloud-Data-Centres %
```

I then created a network in docker

```
aaron@Aarons-MacBook-Air nginx-node-redis % docker network create sample-app
ae8a5eb5f9fe768222d3355bbae4a6336db71c34b4b6c38778b7307fdf982ebc
aaron@Aarons-MacBook-Air nginx-node-redis %
```

Next, I started the redis container

```
aaron@Aarons-MacBook-Air nginx-node-redis % docker run -d --name redis --network sample-app --network-alias
redis redis
Unable to find image 'redis:latest' locally
latest: Pulling from library/redis
673ef6dcec52: Download complete
135541f93cfd: Download complete
4f4fb700ef54: Already exists
507ad0ab32f2: Download complete
ee8be9135819: Download complete
588ff5b1957c: Download complete
171f85931ecd: Download complete
Digest: sha256:6aafb7f25fc93c4ff74e99cff8e85899f03901bc96e61ba12cd3c39e95503c73
Status: Downloaded newer image for redis:latest
bb64df841e8cf93ab4493e72e03f2900cbaac1b8335896d981326b9b1649ce9c
aaron@Aarons-MacBook-Air nginx-node-redis %
```