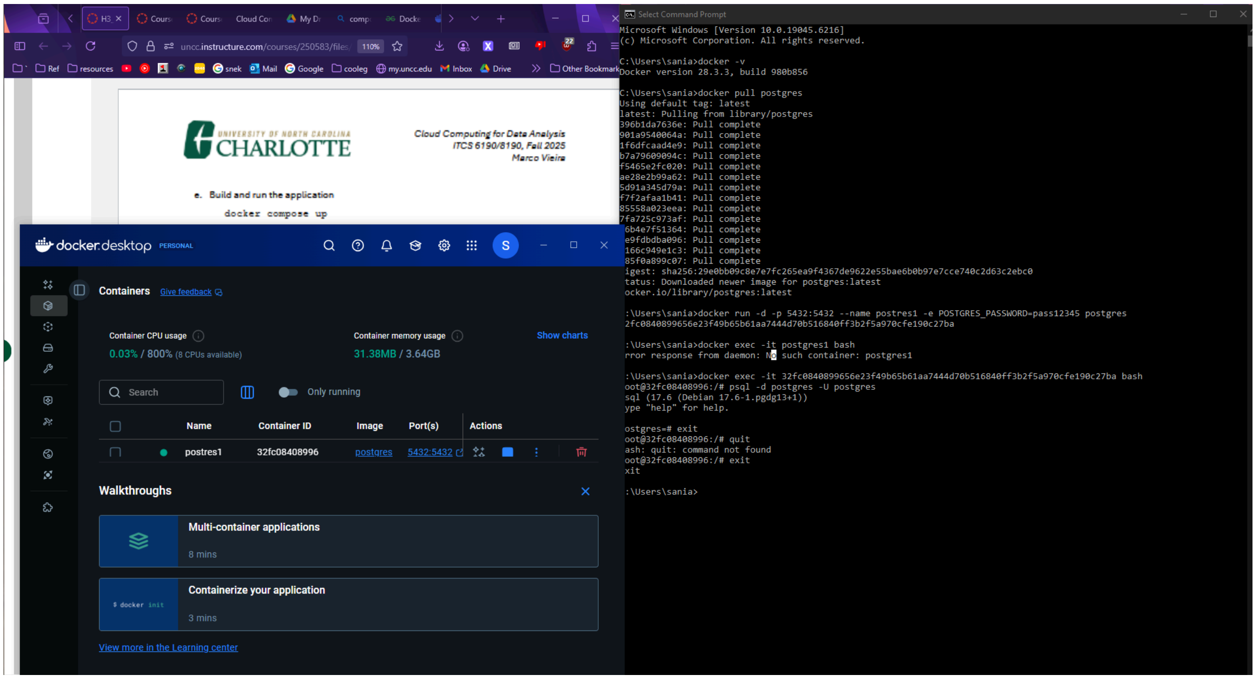
PostgreSQL Setup with Docker:



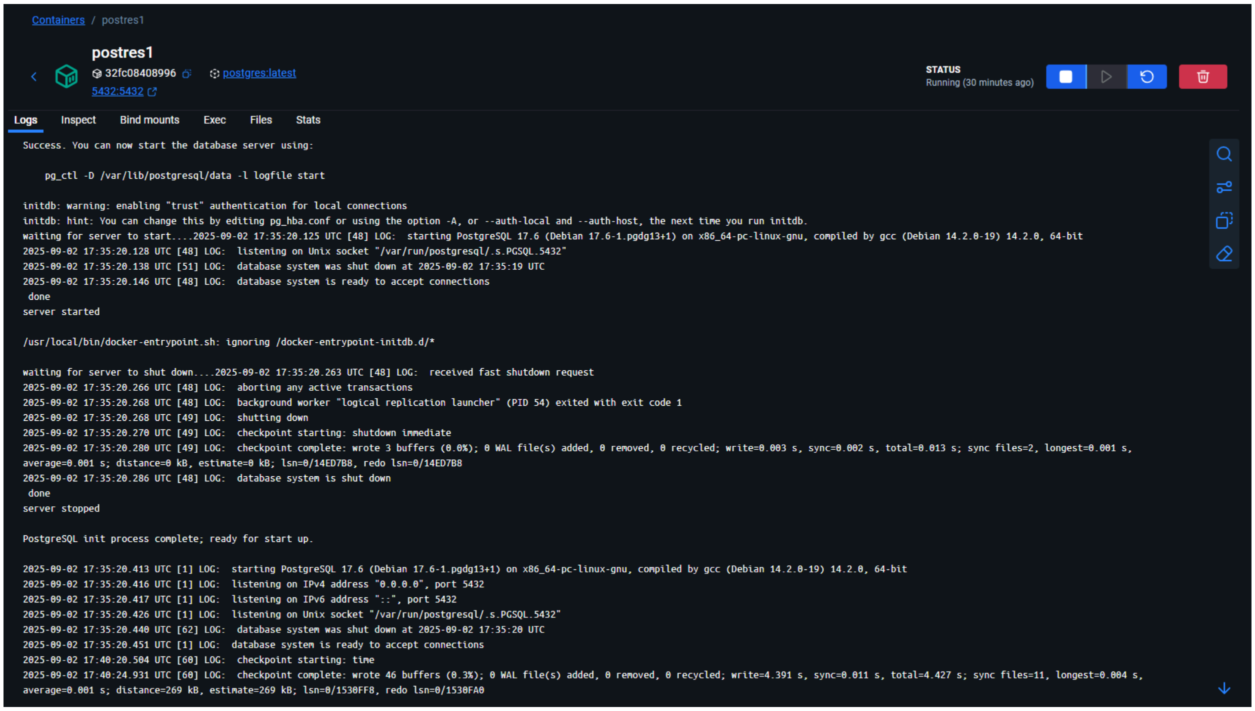
Cloud Computing for Data Analysis
ITCS 6190/8190, Fall 2025
Marco Vieira

e. Build and run the application

```
docker compose up
```

□		Name	Container ID	Image	Port(s)	Actions
□	●	postres1	32fc08408996	postgres	5432:5432	

```
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sania>docker -v
Docker version 28.3.3, build 980b856

C:\Users\sania>docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
396b1da7636e: Pull complete
901a9540064a: Pull complete
1f6dfcaad4e9: Pull complete
b7a79609094c: Pull complete
f5465e2fc020: Pull complete
ae28e2b99a62: Pull complete
5d91a345d79a: Pull complete
f7f2afaa1b41: Pull complete
85558a023eea: Pull complete
7fa725c973af: Pull complete
6b4e7f51364: Pull complete
e9fdbdba096: Pull complete
166c949e1c3: Pull complete
85f0a899c07: Pull complete
igest: sha256:29e0bb09c8e7e7fc265ea9f4367de9622e55bae6b0b97e7cce740c2d63c2ebc0
tatus: Downloaded newer image for postgres:latest
ocker.io/library/postgres:latest

:\Users\sania>docker run -d -p 5432:5432 --name postres1 -e POSTGRES_PASSWORD=pass12345 postgres
2fc0840899656e23f49b65b61aa7444d70b516840ff3b2f5a970cfe190c27ba

:\Users\sania>docker exec -it postgres1 bash
rror response from daemon: No such container: postgres1

:\Users\sania>docker exec -it 32fc0840899656e23f49b65b61aa7444d70b516840ff3b2f5a970cfe190c27ba bash
oot@32fc08408996:/# psql -d postgres -U postgres
sql (17.6 (Debian 17.6-1.pgdg13+1))
ype "help" for help.

ostgres=# exit
oot@32fc08408996:/# quit
ash: quit: command not found
oot@32fc08408996:/# exit
xit

:\Users\sania>
```



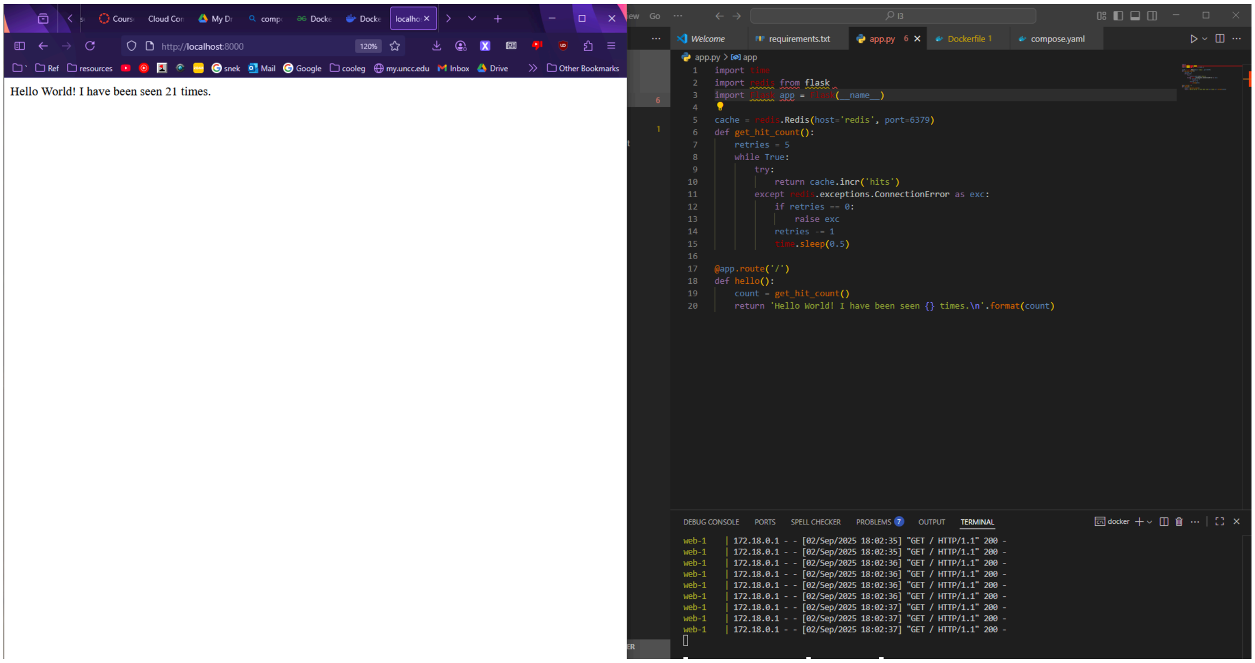
Containers / postres1

postres1
32fc08408996
postgres:latest
5432:5432

STATUS
Running (30 minutes ago)

```
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....2025-09-02 17:35:20.125 UTC [48] LOG:  starting PostgreSQL 17.6 (Debian 17.6-1.pgdg13+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 14.2.0-19) 14.2.0, 64-bit
2025-09-02 17:35:20.128 UTC [48] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2025-09-02 17:35:20.138 UTC [51] LOG:  database system was shut down at 2025-09-02 17:35:19 UTC
2025-09-02 17:35:20.146 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....2025-09-02 17:35:20.263 UTC [48] LOG:  received fast shutdown request
2025-09-02 17:35:20.266 UTC [48] LOG:  aborting any active transactions
2025-09-02 17:35:20.268 UTC [48] LOG:  background worker "logical replication launcher" (PID 54) exited with exit code 1
2025-09-02 17:35:20.268 UTC [49] LOG:  shutting down
2025-09-02 17:35:20.270 UTC [49] LOG:  checkpoint starting: shutdown immediate
2025-09-02 17:35:20.280 UTC [49] LOG:  checkpoint complete: wrote 3 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.003 s, sync=0.002 s, total=0.013 s; sync files=2, longest=0.001 s,
average=0.001 s; distance=0 kB, estimate=0 kB; lsn=0/14ED7B8, redo lsn=0/14ED7B8
2025-09-02 17:35:20.286 UTC [48] LOG:  database system is shut down
 done
server stopped

PostgreSQL init process complete; ready for start up.

2025-09-02 17:35:20.413 UTC [1] LOG:  starting PostgreSQL 17.6 (Debian 17.6-1.pgdg13+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 14.2.0-19) 14.2.0, 64-bit
2025-09-02 17:35:20.416 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
2025-09-02 17:35:20.417 UTC [1] LOG:  listening on IPv6 address "::", port 5432
2025-09-02 17:35:20.426 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2025-09-02 17:35:20.440 UTC [62] LOG:  database system was shut down at 2025-09-02 17:35:20 UTC
2025-09-02 17:35:20.451 UTC [1] LOG:  database system is ready to accept connections
2025-09-02 17:40:20.504 UTC [60] LOG:  checkpoint starting: time
2025-09-02 17:40:24.931 UTC [60] LOG:  checkpoint complete: wrote 46 buffers (0.3%); 0 WAL file(s) added, 0 removed, 0 recycled; write=4.391 s, sync=0.011 s, total=4.427 s; sync files=11, longest=0.004 s,
average=0.001 s; distance=269 kB, estimate=269 kB; lsn=0/1530FF8, redo lsn=0/1530FA0
```

Build a Python Web App with Docker Compose in VS Code:



Hello World! I have been seen 21 times.

```
import time
import redis from flask
import Flask app = Flask(__name__)

cache = redis.Redis(host='redis', port=6379)
def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)

@app.route('/')
def hello():
    count = get_hit_count()
    return 'Hello World! I have been seen {} times.\n'.format(count)
```

```
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:35] "GET / HTTP/1.1" 200 -
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:35] "GET / HTTP/1.1" 200 -
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:36] "GET / HTTP/1.1" 200 -
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:36] "GET / HTTP/1.1" 200 -
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:36] "GET / HTTP/1.1" 200 -
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:36] "GET / HTTP/1.1" 200 -
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:37] "GET / HTTP/1.1" 200 -
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:37] "GET / HTTP/1.1" 200 -
web-1    | 172.18.0.1 - - [02/Sep/2025 18:02:37] "GET / HTTP/1.1" 200 -
```