CECS 327

Project 1: Docker

Ryan Tomas: 028210102

09/20/2025

## Introduction

In this project, we explore the basics of Docker with creating custom Docker images, running images, and building a multi-container setup with server and client communication. The goal was to understand the basics of Docker and how to set up a multi-container environment with a server and client. This deployed a simple Python TCP server with multiple clients.

## Docker

In this project, I worked on two tasks that use Docker. Task one, I had to run a web server with nginx and customize the page with a local index.html file. The way I ran this code was using the command *docker run -d -p 8080:80 -v ${PWD}/index.html:/usr/share/nginx/html/index.html nginx:latest,* and that makes a server. We would have to open a browser that is on http//localhost:8080. The website would have a custom message that I made. The message would be a header that says “this is a test for Docker + Nginx!”

For task 2, I had to make two files called server.py and client.py. These two files would be in their own folder with their respective Dockerfile. Server.py is making a server in Python that would pass a message to client.py. Since we are making multiple clients, I used threading to connect clients at the same time. If I didn’t use it, the server would shut down after the first connection. For client.py, the purpose is to get the message from the server and post the message to the terminal. To make a multi-container, I need to make a yml file. This file needs the location of the server.py and client.py. I need to give a name to the server container. From this, I need to

state that the clients are going to depend on that server. Finally, I would need to run it with *docker-compose up –build*. This command will get messages in the terminal from both the server container and client container. The clients would print “TCP Client Received from the TCP Server: This is the TCP server in the Docker container.”

## Challenges

I had issues setting up the Dockerfile because I had to do some research on how it works. When it was my first time running the Dockerfile, it didn’t work because I had the CMD wrong. Another difficult part I had was setting up the .yml and how to do multiple clients. I didn’t know that I could use the same client.py for multiple builds. Besides those two challenges, I didn’t have problems with the project.

## Conclusion

This project introduced me to the Docker concepts: images, containers, Dockerfiles, and yml file. I successfully ran the server and client Python containers. This built a multi-container setup with a server that had multiple clients. I learned how to make a Dockerfile and yml file. I learned to compose and manage containers.

## Video

Recording 2025-09-19 113727.mp4

Output

```
PS E:\School work\year 5\CECS 327\Project_1_Docker> docker-compose up --build
 => [client2 internal] load .dockerignore
 => => transferring context: 2B
 => [client1 internal] load .dockerignore
 => => transferring context: 2B
 => [client3 internal] load .dockerignore
 => => transferring context: 2B
 => [client2 internal] load build context
 => => transferring context: 31B
 => [client1 internal] load build context
 => => transferring context: 31B
 => [client3 internal] load build context
 => => transferring context: 31B
 => CACHED [client3 2/3] WORKDIR /project_1_Docker/task6_client
 => CACHED [client2 3/3] COPY client.py .
 => [client1] exporting to image
 => => exporting layers
 => => exporting manifest sha256:3ba6d609bb6a1ef932466c569267b8ebb046f500b6a7c86c2d72ba6ab367c951
 => => exporting config sha256:fdc648cf260f9a50e921a80d978fa303ab048c62ede95ed6ec13d1cfee15e03a
 => => exporting attestation manifest sha256:a0432ccc39739431dc5d0e8e7de850838d6fd9e968d8118ce0388d62bcae6e39
 => => exporting manifest list sha256:50715c1704b2f6e88235a1cec3e4a31933466a27d999ec46612e855f24200ab6
 => => naming to docker.io/library/project_1_docker-client1:latest
 => => unpacking to docker.io/library/project_1_docker-client1:latest
 => [client3] exporting to image
 => => exporting layers
 => => exporting manifest sha256:985cbe7707ab218cc937507be33949bcebe398f1c0334e8b26d16c1fef3deaad
 => => exporting config sha256:a061c8caa309af0d18ddb70cd5e685738d5bc4078973126099883bae83729a35
 => => exporting attestation manifest sha256:3dce0a8947d105cd300c3d388457ebda0c174a4278420722e597ba59c6599bb1
 => => exporting manifest list sha256:b89626bd773041703af33f140a284ed89b98c16a2db09260a7da931451fcec72
 => => naming to docker.io/library/project_1_docker-client3:latest
 => => unpacking to docker.io/library/project_1_docker-client3:latest
 => [client2] exporting to image
 => => exporting layers
 => => exporting manifest sha256:3a734b91624ae6f2d349d21b95fa04f986db2a775ac8ee7483f1fdd14faced31
 => => exporting config sha256:0664391140be1f02415a8b13179ef665f3c0c487e633810e7602eb65b6b3deb0
 => => exporting attestation manifest sha256:22c4d0955cae33ae48641757148bbbb19318a0210968d781ccc19add36517e11
 => => exporting manifest list sha256:b35eb2d42d2470b320ba15f206b189871dd8c0a6d3fca372dd382134bade2dec
 => => naming to docker.io/library/project_1_docker-client2:latest
 => => unpacking to docker.io/library/project_1_docker-client2:latest
 => [client1] resolving provenance for metadata file
 => [client3] resolving provenance for metadata file
 => [client2] resolving provenance for metadata file
[+] Running 8/8
 ✓ client1                              Built
 ✓ client2                              Built
 ✓ client3                              Built
 ✓ server                               Built
 ✓ Container project_1_docker-client3-1  Recreated
 ✓ Container project_1_docker-client1-1  Recreated
 ✓ Container project_1_docker-client2-1  Recreated
 ✓ Container my_server                  Recreated
Attaching to my_server, client1-1, client2-1, client3-1
my_server  | Server is running
my_server  | TCP Listening on the host and port: 0.0.0.0:5000
my_server  | TCP Connected by the address ('172.19.0.3', 54458)
my_server  | TCP Received from  address ('172.19.0.3', 54458): Hello from TCP client that is using Docker
client1-1  | TCP Client Received from the TCP Server: This is the TCP server in the Docker container
my_server  | TCP Connected by the address ('172.19.0.5', 42906)
my_server  | TCP Received from  address ('172.19.0.5', 42906): Hello from TCP client that is using Docker
client3-1  | TCP Client Received from the TCP Server: This is the TCP server in the Docker container
my_server  | TCP Connected by the address ('172.19.0.4', 59012)
client2-1  | TCP Client Received from the TCP Server: This is the TCP server in the Docker container
my_server  | TCP Received from  address ('172.19.0.4', 59012): Hello from TCP client that is using Docker
client1-1 exited with code 0
client2-1 exited with code 0
client3-1 exited with code 0
Gracefully stopping... (press Ctrl+C again to force)
[+] Stopping 4/4
 ✓ Container project_1_docker-client3-1  Stopped
 ✓ Container project_1_docker-client1-1  Stopped
 ✓ Container project_1_docker-client2-1  Stopped
 ✓ Container my_server                  Stopped
```

# Logs

time="2025-09-18T19:47:02-07:00" level=warning msg="E:\\School work\\year 5\\CECS 327\\Project_1_Docker\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"

client2-1  | TCP Client Received from the TCP Server: This is the TCP server in the Docker container

client1-1  | TCP Client Received from the TCP Server: This is the TCP server in the Docker container

client3-1  | TCP Client Received from the TCP Server: This is the TCP server in the Docker container

my_server  | Server is running

my_server  | TCP Listening on the host and port: 0.0.0.0:5000

my_server  | TCP Connected by the address ('172.19.0.3', 54458)

my_server  | TCP Received from  address ('172.19.0.3', 54458): Hello from TCP client that is using Docker

my_server  | TCP Connected by the address ('172.19.0.5', 42906)

my_server  | TCP Received from  address ('172.19.0.5', 42906): Hello from TCP client that is using Docker

my_server  | TCP Connected by the address ('172.19.0.4', 59012)

my_server  | TCP Received from  address ('172.19.0.4', 59012): Hello from TCP client that is using Docker