

Containerizing a Multi-Service Application with Docker Compose

The main objective for this lab was to reduce the workload and hence provide efficiency in creating containers to run multiple services at the same time. A docker compose file was created and based on the docker compose file, multiple container images were created which was used to run containers providing a platform to test and develop business logic. The images created can be used in other platforms compatible with docker.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> mkdir chichi_docker_lab2

Directory: C:\WINDOWS\system32

Mode                LastWriteTime         Length Name
----                -
d-----          5/2/2025   6:55 PM                chichi_docker_lab2

PS C:\WINDOWS\system32> cd chichi_docker_lab2
PS C:\WINDOWS\system32\chichi_docker_lab2> mkdir frontend

Directory: C:\WINDOWS\system32\chichi_docker_lab2

Mode                LastWriteTime         Length Name
----                -
d-----          5/2/2025   7:15 PM                frontend

PS C:\WINDOWS\system32\chichi_docker_lab2> mkdir backend

Directory: C:\WINDOWS\system32\chichi_docker_lab2

Mode                LastWriteTime         Length Name
----                -
d-----          5/2/2025   7:15 PM                backend

PS C:\WINDOWS\system32\chichi_docker_lab2> cd frontend
PS C:\WINDOWS\system32\chichi_docker_lab2\frontend> echo "" > index.html
PS C:\WINDOWS\system32\chichi_docker_lab2\frontend> echo "" > Dockerfile
PS C:\WINDOWS\system32\chichi_docker_lab2\frontend> cd ..
```

```
Administrator: Windows PowerShell

Directory: C:\WINDOWS\system32\chichi_docker_lab2

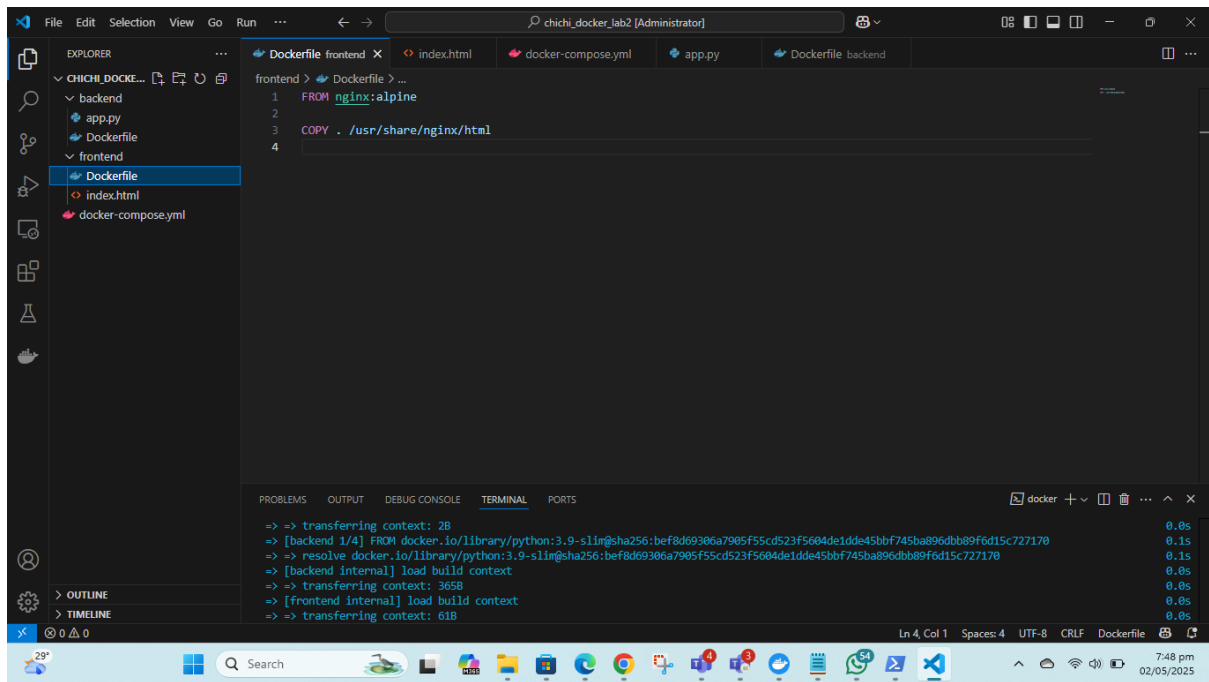
Mode                LastWriteTime         Length Name
----                -
d-----          5/2/2025   7:15 PM                backend

PS C:\WINDOWS\system32\chichi_docker_lab2> cd frontend
PS C:\WINDOWS\system32\chichi_docker_lab2\frontend> echo "" > index.html
PS C:\WINDOWS\system32\chichi_docker_lab2\frontend> echo "" > Dockerfile
PS C:\WINDOWS\system32\chichi_docker_lab2\frontend> cd ..
PS C:\WINDOWS\system32\chichi_docker_lab2> cd backend
PS C:\WINDOWS\system32\chichi_docker_lab2\backend> echo "" > Dockerfile
PS C:\WINDOWS\system32\chichi_docker_lab2\backend> echo "" > app.py
PS C:\WINDOWS\system32\chichi_docker_lab2\backend> cd ..
PS C:\WINDOWS\system32\chichi_docker_lab2> ls

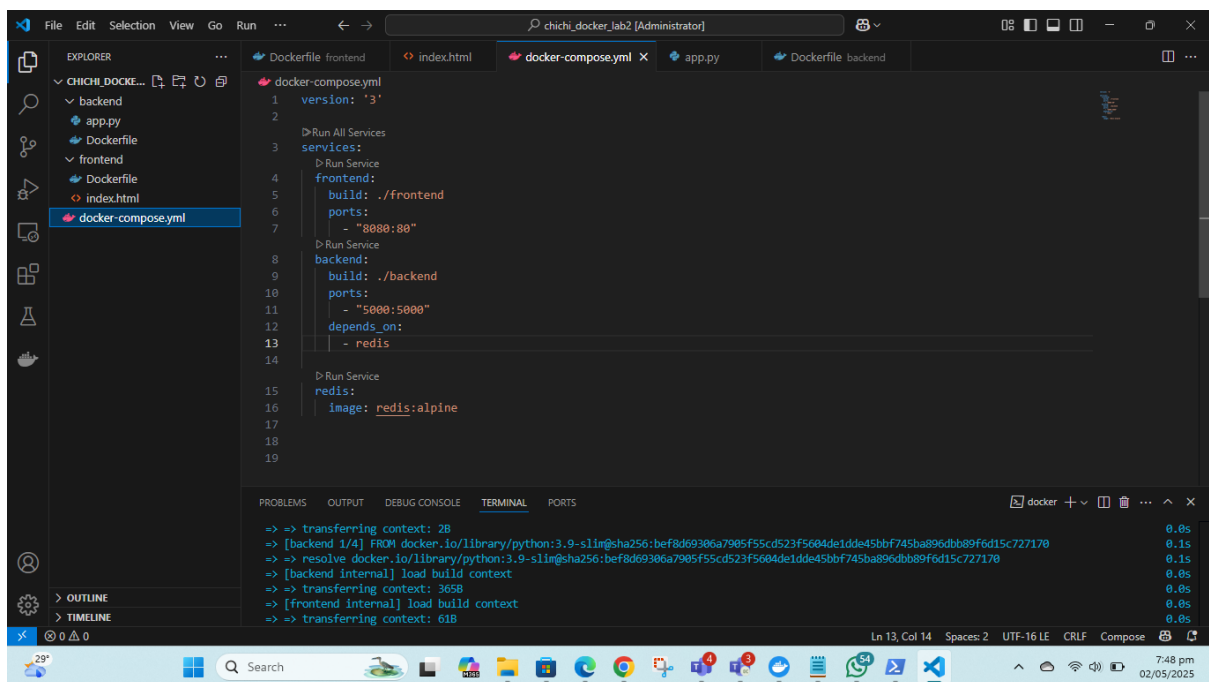
Directory: C:\WINDOWS\system32\chichi_docker_lab2

Mode                LastWriteTime         Length Name
----                -
d-----          5/2/2025   7:17 PM                backend
d-----          5/2/2025   7:16 PM                frontend

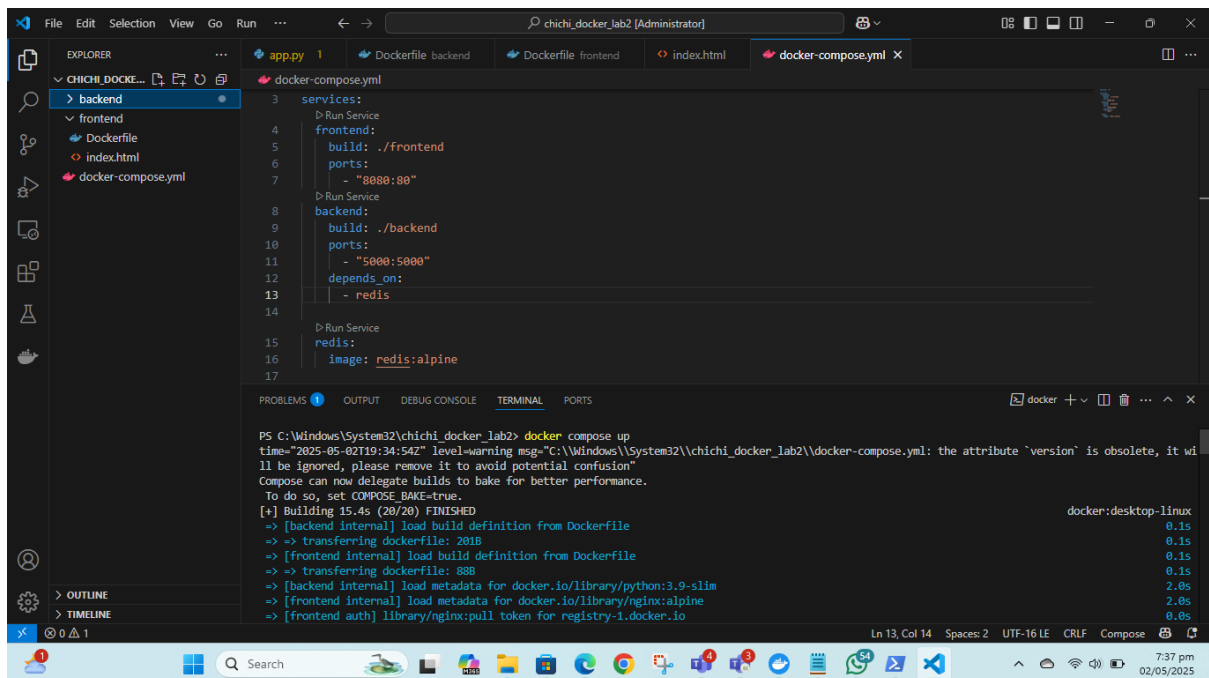
PS C:\WINDOWS\system32\chichi_docker_lab2> echo "" > docker-compose.yml
PS C:\WINDOWS\system32\chichi_docker_lab2> code .
```



In this lab, two subdirectories named frontend and backend were created. The frontend directory contained an index.html file which served as the main entry point for the frontend application as well as a docker file which was executed to build an image to run a container. The backend directory also contained an app.py file which served as the main entry point for the backend application. The backend application also had its corresponding docker file for the creation of the docker image. The running of the backend application depended on the redis service.



The docker compose file compiled all the services and was used to generate an image for each service. Based on the version of the docker compose file which was version 3, It was able to house and build images for three services (backend, frontend and redis service).

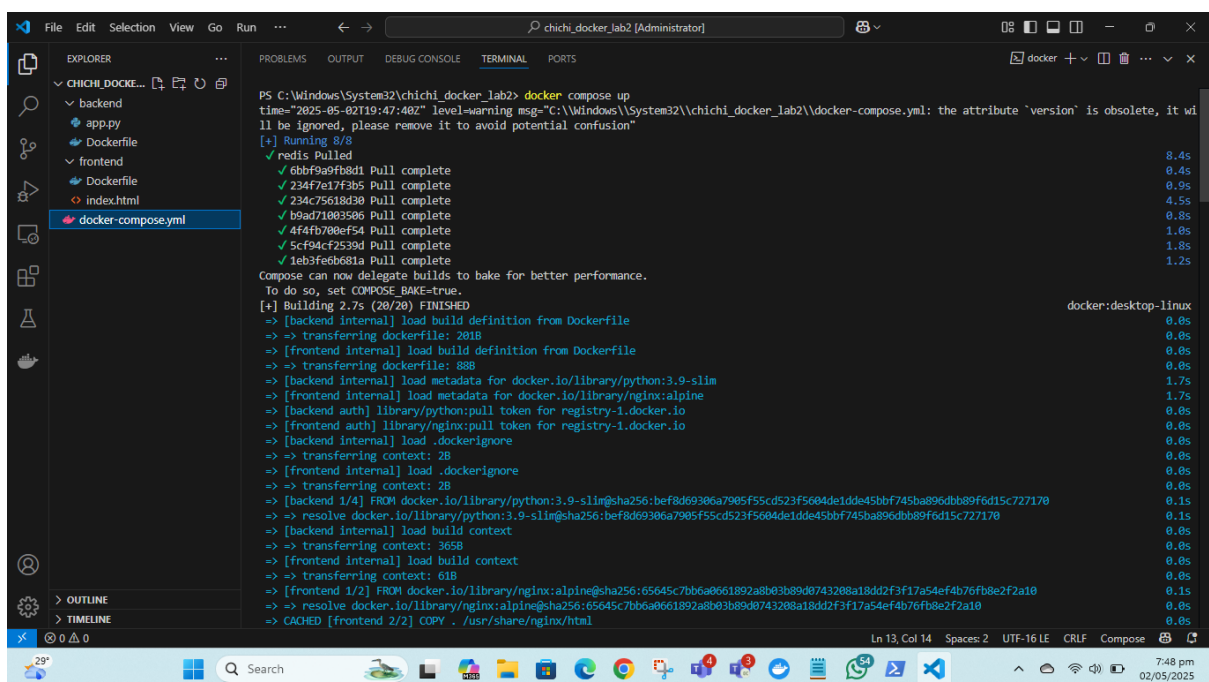


The screenshot shows the Visual Studio Code editor with the `docker-compose.yml` file open. The file defines three services: `frontend`, `backend`, and `redis`. The `frontend` service builds from `./frontend` and listens on port 8080. The `backend` service builds from `./backend` and listens on port 5000, with a dependency on the `redis` service. The `redis` service uses the `redis:alpine` image.

```
services:
  frontend:
    build: ./frontend
    ports:
      - "8080:80"
  backend:
    build: ./backend
    ports:
      - "5000:5000"
    depends_on:
      - redis
  redis:
    image: redis:alpine
```

The terminal output shows the command `docker compose up` being executed. It displays a warning about the deprecated `version` attribute and proceeds to build the images. The build process for the `frontend` and `backend` services is shown, including the transfer of Dockerfiles and the loading of build definitions. The `redis` service is pulled from the Docker registry.

```
PS C:\Windows\System32\chichi_docker_lab2> docker compose up
time="2025-05-02T19:34:54Z" level=warning msg="C:\Windows\System32\chichi_docker_lab2\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
Compose can now delegate builds to bake for better performance.
To do so, set COMPOSE_BAKE=true.
[+] Building 15.4s (20/20) FINISHED
=> [backend internal] load build definition from Dockerfile
=> transferring dockerfile: 201B
=> [frontend internal] load build definition from Dockerfile
=> transferring dockerfile: 88B
=> [backend internal] load metadata for docker.io/library/python:3.9-slim
=> [frontend internal] load metadata for docker.io/library/nginx:alpine
=> [frontend auth] library/nginx:pull token for registry-1.docker.io
docker:desktop-linux
0.1s
0.1s
0.1s
0.1s
2.0s
2.0s
0.0s
```



The screenshot shows the Visual Studio Code editor with the `docker-compose.yml` file open. The file defines three services: `frontend`, `backend`, and `redis`. The `frontend` service builds from `./frontend` and listens on port 8080. The `backend` service builds from `./backend` and listens on port 5000, with a dependency on the `redis` service. The `redis` service uses the `redis:alpine` image.

```
services:
  frontend:
    build: ./frontend
    ports:
      - "8080:80"
  backend:
    build: ./backend
    ports:
      - "5000:5000"
    depends_on:
      - redis
  redis:
    image: redis:alpine
```

The terminal output shows the command `docker compose up` being executed. It displays a warning about the deprecated `version` attribute and proceeds to build the images. The build process for the `frontend` and `backend` services is shown, including the transfer of Dockerfiles and the loading of build definitions. The `redis` service is pulled from the Docker registry.

```
PS C:\Windows\System32\chichi_docker_lab2> docker compose up
time="2025-05-02T19:47:40Z" level=warning msg="C:\Windows\System32\chichi_docker_lab2\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 8/8
✔ redis Pulled
✔ 6bbf9a9f8d1 Pull complete
✔ 234c75618d30 Pull complete
✔ 234c75618d30 Pull complete
✔ b9ad71083506 Pull complete
✔ 4f4fb708ef54 Pull complete
✔ 5cf94cf2539d Pull complete
✔ 1eb3feb681a Pull complete
Compose can now delegate builds to bake for better performance.
To do so, set COMPOSE_BAKE=true.
[+] Building 2.7s (20/20) FINISHED
=> [backend internal] load build definition from Dockerfile
=> transferring dockerfile: 201B
=> [frontend internal] load build definition from Dockerfile
=> transferring dockerfile: 88B
=> [backend internal] load metadata for docker.io/library/python:3.9-slim
=> [frontend internal] load metadata for docker.io/library/nginx:alpine
=> [backend auth] library/python:pull token for registry-1.docker.io
=> [frontend auth] library/nginx:pull token for registry-1.docker.io
=> [backend internal] load .dockerignore
=> transferring context: 2B
=> [frontend internal] load .dockerignore
=> transferring context: 2B
=> [backend 1/4] FROM docker.io/library/python:3.9-slim@sha256:bef8d69306a7905f55cd523f5604de1dde45bbf745ba896dbb89fd15c727170
=> resolve docker.io/library/python:3.9-slim@sha256:bef8d69306a7905f55cd523f5604de1dde45bbf745ba896dbb89fd15c727170
=> [backend internal] load build context
=> transferring context: 365B
=> [frontend internal] load build context
=> transferring context: 61B
=> [frontend 1/2] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10
=> resolve docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10
=> CACHED [frontend 2/2] COPY ./usr/share/nginx/html
docker:desktop-linux
0.0s
0.0s
0.0s
0.0s
1.7s
1.7s
0.0s
0.0s
0.0s
0.0s
0.0s
0.0s
0.1s
0.1s
0.0s
0.0s
0.0s
```

```
back-end-1 | * Debug mode: off
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 31
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 32
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 33
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 34
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 35
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 36
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 37
back-end-1 | * Serving Flask app 'app'
back-end-1 | * Debug mode: off
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 33
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 34
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 35
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 36
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 37
back-end-1 | * Serving Flask app 'app'
back-end-1 | * Debug mode: off
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 36
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 37
back-end-1 | * Serving Flask app 'app'
back-end-1 | * Debug mode: off
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 37
back-end-1 | * Serving Flask app 'app'
back-end-1 | * Debug mode: off
back-end-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
back-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 36
front-end-1 | 2025/05/02 19:47:53 [notice] 1#1: start worker process 37
back-end-1 | * Serving Flask app 'app'
back-end-1 | * Debug mode: off
back-end-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
back-end-1 | * Running on all addresses (0.0.0.0)
back-end-1 | * Running on http://127.0.0.1:5000
back-end-1 | * Running on http://172.19.0.4:5000
back-end-1 | Press CTRL+C to quit
```

chichi_docker_lab2
C:\Windows\System32\chichi_docker_lab2

frontend •
chichi_docker_lab2-8080:80

redis •
redis:alpine

backend •
chichi_docker_lab2-5000:5000

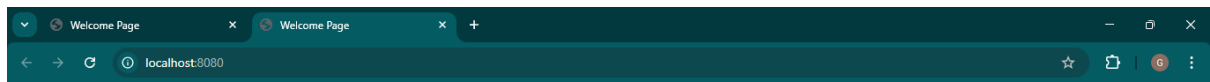
View configurations

/docker-entrypoint.sh: Configuration complete; ready for start up
2025/05/02 19:47:53 [notice] 1#1: using the "epoll" event method
2025/05/02 19:47:53 [notice] 1#1: nginx/1.27.5
2025/05/02 19:47:53 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
2025/05/02 19:47:53 [notice] 1#1: OS: Linux 5.15.167.4-microsoft-standard-WSL2
2025/05/02 19:47:53 [notice] 1#1: getrlimit(RLIMIT_NOFILE):
1048576:1048576
2025/05/02 19:47:53 [notice] 1#1: start worker processes
2025/05/02 19:47:53 [notice] 1#1: start worker process 30
2025/05/02 19:47:53 [notice] 1#1: start worker process 31
2025/05/02 19:47:53 [notice] 1#1: start worker process 32
2025/05/02 19:47:53 [notice] 1#1: start worker process 33
2025/05/02 19:47:53 [notice] 1#1: start worker process 34
2025/05/02 19:47:53 [notice] 1#1: start worker process 35
2025/05/02 19:47:53 [notice] 1#1: start worker process 36
2025/05/02 19:47:53 [notice] 1#1: start worker process 37

backend

* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on <http://127.0.0.1:5000>
* Running on <http://172.19.0.4:5000>
Press CTRL+C to quit

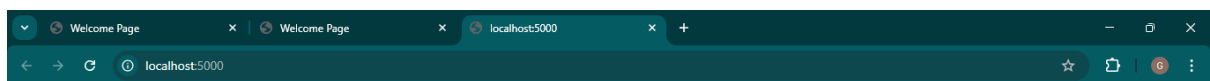
Engine running | RAM 1.30 GB CPU 0.00% Disk: 1.51 GB used (limit 1006.85 GB) | Terminal | New version available



index.html

Hello, welcome to our application! I'm Gertrude Chichi, and I'm excited to have you here. Our team is working hard to bring you the best experience possible. Stay tuned for updates and new features! !

This is a sample HTML page.



Hello, World!

