

## Exercise 1: Getting Started with Containers

Objective: Learn how to run and inspect containers.

- Run hello-world, nginx, and alpine containers.

```
C:\Users\alefi>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:a0dfb0aac212703bfcb339d77d47ec32c8706ff250850ecc0e19c8737b18567
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

C:\Users\alefi>docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

```
C:\Users\alefi>docker run -d --name mynginx -p 8000:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
e5d9bb0b85cc: Pull complete
716cdf61af59: Pull complete
c3741b707ce6: Pull complete
14e422fd20a0: Pull complete
14a859b5ba24: Pull complete
b1badc6e5066: Pull complete
a2da0c0f2353: Pull complete
Digest: sha256:33e0bbc7ca9ecf108140af6288c7c9d1ecc77548cbfd3952fd8466a75edefe57
Status: Downloaded newer image for nginx:latest
50396b10458245bbfc76a5ea49fb44aebc48b53b7a674ba6f3d1a252f7fa610
```

```
C:\Users\alefi>docker run -it --name myalpine alpine sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
9824c27679d3: Pull complete
Digest: sha256:4bcff63911fcb4448bd4fdacec207030997caf25e9bea4045fa6c8c44de311d1
Status: Downloaded newer image for alpine:latest
/ # la
sh: la: not found
/ # ls
bin dev etc home lib media mnt opt proc root run sbin srv sys tmp usr v
/ # exit
```

- Use docker ps and docker ps -a to inspect states.

```
C:\Users\alefi>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
50396b104582 nginx "/docker-entrypoint..." 6 minutes ago Up 6 minutes 0.0.0.0:8000->80/tcp, [::]:8000->80/tcp mynginx

C:\Users\alefi>docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
992979e99282 alpine "sh" About a minute ago Exited (0) About a minute ago myalpin
e
50396b104582 nginx "/docker-entrypoint..." 7 minutes ago Up 7 minutes 0.0.0.0:8000->80/tcp, [::]:8000->80/tcp mynginx
0212e66b41dd hello-world "/hello" 27 minutes ago Exited (0) 27 minutes ago clever_gauss
```

- Explore docker run flags: --rm, -it, -d, -p.

```
C:\Users\alefi>docker rm myalpine
myalpine

C:\Users\alefi>docker rmi alpine
Untagged: alpine:latest
Deleted: sha256:4bcff63911fcb4448bd4fdacec207030997caf25e9bea4045fa6c8c44de311d1
```

```
C:\Users\alefi>docker run -it alpine sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
9824c27679d3: Pull complete
Digest: sha256:4bcff63911fcb4448bd4fdacec207030997caf25e9bea4045fa6c8c44de311d1
Status: Downloaded newer image for alpine:latest
/ # exit
```

```
C:\Users\alefi>docker run -d --name mynginx -p 8000:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
e5d9bb0b85cc: Pull complete
716cdf61af59: Pull complete
c3741b707ce6: Pull complete
14e422fd20a0: Pull complete
14a859b5ba24: Pull complete
b1badc6e5066: Pull complete
a2da0c0f2353: Pull complete
Digest: sha256:33e0bbc7ca9ecf108140af6288c7c9d1ecc77548cbfd3952fd8466a75edefe57
Status: Downloaded newer image for nginx:latest
50396b10458245bbfc76a5ea49fb44aebc48b53b7a674ba6f3d1a252f7fa610
```

- Use docker exec and docker logs.

```
C:\Users\alefi> docker exec -it mynginx sh
# ls
bin  boot  dev  docker-entrypoint.d  docker-entrypoint.sh  etc  home  lib  lib64  media  mnt  opt  proc  root
# exit

C:\Users\alefi> docker logs mynginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/08/22 13:55:22 [notice] 1#1: using the "epoll" event method
2025/08/22 13:55:22 [notice] 1#1: nginx/1.29.1
2025/08/22 13:55:22 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u1)
2025/08/22 13:55:22 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
2025/08/22 13:55:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/08/22 13:55:22 [notice] 1#1: start worker processes
2025/08/22 13:55:22 [notice] 1#1: start worker process 29
2025/08/22 13:55:22 [notice] 1#1: start worker process 30
2025/08/22 13:55:22 [notice] 1#1: start worker process 31
2025/08/22 13:55:22 [notice] 1#1: start worker process 32
2025/08/22 13:55:22 [notice] 1#1: start worker process 33
2025/08/22 13:55:22 [notice] 1#1: start worker process 34
2025/08/22 13:55:22 [notice] 1#1: start worker process 35
2025/08/22 13:55:22 [notice] 1#1: start worker process 36
2025/08/22 13:55:22 [notice] 1#1: start worker process 37
2025/08/22 13:55:22 [notice] 1#1: start worker process 38
2025/08/22 13:55:22 [notice] 1#1: start worker process 39
2025/08/22 13:55:22 [notice] 1#1: start worker process 40
2025/08/22 13:55:22 [notice] 1#1: start worker process 41
2025/08/22 13:55:22 [notice] 1#1: start worker process 42
2025/08/22 13:55:22 [notice] 1#1: start worker process 43
2025/08/22 13:55:22 [notice] 1#1: start worker process 44
```

## Exercise 2: Working with Container State

Objective: Modify containers and commit custom images.

- Run an Ubuntu container, install curl and vim.

```
C:\Users\alefi> docker run -it --name myubuntu ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
b71466b94f26: Pull complete
Digest: sha256:7c06e91f61fa88c08cc74f7e1b7c69ae24910d745357e0dfe1d2c0322aaaf20f9
Status: Downloaded newer image for ubuntu:latest
root@0e1d161a52a3:/# apt-get update
apt-get install -y curl vim
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2066 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu/main amd64 Packages [1808 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1137 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [23.0 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1363 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [45.2 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2193 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1713 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1458 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [35.6 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [48.8 kB]
Fetched 32.3 MB in 5s (6385 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

- Exit and commit the image as ubuntu-tools.

```
w. Operation was interrupted before it could finish
root@0e1d161a52a3:/# exit
exit

C:\Users\alefi>docker commit myubuntu ubuntu-tools
sha256:2df4889a9471e4cf7590db404e6b9d8e43066e179e9baa5e31be823a58b29ed4
```

- Run a new container from the committed image.

```
C:\Users\alefi>docker run -it ubuntu-tools
root@b1be9cbf3db3:/# curl --version
vim --version
curl 8.5.0 (x86_64-pc-linux-gnu) libcurl/8.5.0 OpenSSL/3.0.13 zlib/1.3 brotli
enssl/zlib nghttp2/1.59.0 librtmp/2.3 OpenLDAP/2.6.7
Release-Date: 2023-12-06, security patched: 8.5.0-2ubuntu10.6
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps
```

- Tag the image and list it with docker images.

```
C:\Users\alefi>docker tag ubuntu-tools ubuntu-tools:1.0

C:\Users\alefi>docker images
REPOSITORY      TAG          IMAGE ID       CREATED        SIZE
ubuntu-tools    1.0          2df4889a9471   6 minutes ago  323MB
ubuntu-tools    latest        2df4889a9471   6 minutes ago  323MB
nginx           latest        33e0bbc7ca9e   9 days ago    279MB
hello-world     latest        a0dfb02aac21   13 days ago   20.3kB
ubuntu          latest        7c06e91f61fa   3 weeks ago   117MB
alpine          latest        4bcff63911fc   5 weeks ago   12.8MB
```

## Exercise 3: Build Custom Images Using Dockerfile

Objective: Write Docker files and build your own image.

- Create a simple Node or Python web server.
  - Write a Dockerfile to copy the code and expose a port.
  - Add metadata using LABEL and set CMD or ENTRYPOINT.
  - Build and run the image. Test with curl.

```
C:\Users\alefi\dockerpractice>docker build -t python-webserver .
[+] Building 12.2s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 455B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerrcignore
=> transferring context: 2B
=> [1/3] FROM docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0
=> resolve docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0
=> sha256:5ec99fe17015e703c289d110b020e4362d5b425be957d68fb400d56d83f234 13.37MB / 13.37MB
=> sha256:ea3499df3804f0a84e9f076a05f0fce2a64d8fc88494ce682df9204c6a18a91 249B / 249B
=> sha256:0219e1e5e6ef3ef9d91f78826576a112b1c20622c10c294a4a105811454d1cb1 1.29MB / 1.29MB
=> sha256:396b1da7636e2cd10565bc4bf2f952cbb4a838b58d3b86a2cacb172fb70117c 29.77MB / 29.77MB
=> extracting sha256:396b1da7636e2cd10565bc4bf2f952cbb4a838b58d3b86a2cacb172fb70117c
=> extracting sha256:0219e1e5e6ef3ef9d91f78826576a112b1c20622c10c294a4a105811454d1cb1
=> extracting sha256:5ec99fe17015e703c289d110b020e4362d5b425be957d68fb400d56d83f234
=> extracting sha256:ea3499df3804f0a84e9f076a05f0fce2a64d8fc88494ce682df9204c6a18a91
=> [internal] load build context
=> transferring context: 254B
=> [2/3] WORKDIR /app
=> [3/3] COPY app.py /app/
=> exporting to image
=> exporting layers
=> exporting manifest sha256:d3bb582d52c76082f55bdd6fb35ef2dd84ff3eed0696644fe02d4d44a00cf60f
=> exporting config sha256:dd0864752cc043c73231d94a243d834cf6d16aadbe034611d850fe98bf2877ec
=> exporting attestation manifest sha256:5f62e7a2880fa95822f3dce8d829a30183e4817d41cf23b7a7c43b5957d9cca0
=> exporting manifest list sha256:079d48b5d79380f3b5be289bbe7129c1ecf03ca4ae351607c922d4e2b51894fd
=> naming to docker.io/library/python-webserver:latest
=> => unpacking to docker.io/library/python-webserver:latest

C:\Users\alefi\dockerpractice>docker run -d -p 7000:5000 python-webserver
eb1e68a28b5bef3cb9f08dac069a72509cb518bb4ead8fdc775203097ecfcfd6

C:\Users\alefi\dockerpractice>curl http://localhost:7000
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
</head>
```

```
C:\Users\alefi\dockerpractice>docker run -d -p 7000:5000 python-webserver  
eb1e68a28b5bef3cb9f08dac069a72509cb518bb4ead8dfdc775203097ecfcfd6  
  
C:\Users\alefi\dockerpractice>curl http://localhost:7000  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
<title>Directory listing for /</title>  
</head>  
<body>  
<h1>Directory listing for /</h1>  
<hr>  
<ul>  
<li><a href="app.py">app.py</a></li>  
</ul>  
<hr>  
</body>  
</html>
```

## Exercise 4: Sharing Images

Objective: Push images to Docker Hub.

- Create a Docker Hub account.
- Tag your custom image.
- Push it to Docker Hub.
- Pull it from Docker Hub.

```
C:\Users\alefi\dockerpractice>docker login
Authenticating with existing credentials... [Username: alefiya21]

[ Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded

C:\Users\alefi\dockerpractice>docker tag python-webserver alefiya21/python-webserver:1.0

C:\Users\alefi\dockerpractice>docker push alefi/python-webserver:1.0
The push refers to repository [docker.io/alefi/python-webserver]
tag does not exist: alefi/python-webserver:1.0

C:\Users\alefi\dockerpractice>docker push alefiya21/python-webserver:1.0
The push refers to repository [docker.io/alefiya21/python-webserver]
396b1da7636e: Pushed
717b74f0166e: Pushed
6500893b6ac9: Pushed
0219e1e5e6ef: Pushed
ea3499df304f: Pushed
e17d5ee478f5: Pushed
5ec99fe17015: Pushed
1.0: digest: sha256:079d48b5d79380f3b5be289bbe7129c1ecf03ca4ae351607c922d4e2b51894fd size: 856

C:\Users\alefi\dockerpractice>docker pull alefiya21/python-webserver:1.0
1.0: Pulling from alefiya21/python-webserver
Digest: sha256:079d48b5d79380f3b5be289bbe7129c1ecf03ca4ae351607c922d4e2b51894fd
Status: Image is up to date for alefiya21/python-webserver:1.0
docker.io/alefiya21/python-webserver:1.0
```

## Exercise 5: Data Persistence with Volumes

Objective: Use volumes to persist container data.

- Launch a busybox container with a named volume.
- Insert sample data.
- Stop, remove, and relaunch to verify persistence.
- Try bind mount using -v \${pwd}/data:/data.

```
C:\Users\alefi>docker run -it --name busy1 -v mydata:/data busybox sh
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
80bfbb8a41a2: Pull complete
Digest: sha256:ab33eacc8251e3807b85bb6dba570e4698c3998eca6f0fc2ccb60575a563ea74
Status: Downloaded newer image for busybox:latest
/ # echo "Hello Docker Volume" > /data/hello.txt
/ # cat /data/hello.txt
Hello Docker Volume
/ # exit

C:\Users\alefi>docker stop busy1
busy1

C:\Users\alefi>docker rm busy1
busy1

C:\Users\alefi>docker run -it --name busy2 -v mydata:/data busybox sh
/ # cat /data/hello.txt
Hello Docker Volume
/ # exit

C:\Users\alefi>mkdir data

C:\Users\alefi>docker run -it -v ${PWD}/data:/data busybox sh
docker: Error response from daemon: create ${PWD}/data: "${PWD}/data" includes
are allowed. If you intended to pass a host directory, use absolute path

Run 'docker run --help' for more information

C:\Users\alefi>docker run -it -v C:/Users/alefi/data:/data busybox sh
/ # echo "Bind mount works!" > /data/test.txt
/ # exit
```

## Exercise 6: Container Networking Basics

Objective: Set up communication between containers.

- Start an nginx container and a busybox container.
- Create a user-defined bridge network.
- Attach both containers to the network.
- From busybox, use wget or curl to access nginx.

```
C:\Users\alefi>docker network create mynet
f37ade3d59f1366f58ea4b977ab94d04eef71209dd1d6c8982e2348aca4af669

C:\Users\alefi>docker run -d --name nginx2 --network mynet nginx
9fa5698f6e625ca3635213d9e8bab65080f5aed2f4ac4c979d8e2983f782226f

C:\Users\alefi>docker run -it --name busybox2 --network mynet busybox sh
/ # wget -qO- http://nginx2
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ # exit
```

## Exercise 7: Building a Two-Tier App

Objective: Deploy a small web + DB stack without Compose.

- Manually run a Python app container and a Postgres container.
- Use environment variables to configure the connection.
- Verify the app connects to the DB and serves content.

Extension:

Add a volume to persist the DB data.

```
C:\Users\alefi>docker run -d --name mypostgres -e POSTGRES_USER=testuser -e POSTGRES_PASSWORD=testpass -e POSTGRES_DB=testdb -p 5432:5432 -v pgdata:/var/lib/postgresql/data postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
085f0a899c07: Pull complete
f7f2afa1b41: Pull complete
be9fdbdba096: Pull complete
ae28e2b99a62: Pull complete
b7a79609094c: Pull complete
36b4e7f51364: Pull complete
f5465e2fc020: Pull complete
c166c949e1c3: Pull complete
85558a023eea: Pull complete
7fa725c973af: Pull complete
901a9540064a: Pull complete
5d91a345d79a: Pull complete
1f6dfcaad4e9: Pull complete
Digest: sha256:29e0bb09c8e7fc265ea9f4367de9622e55bae6b0b97e7cce740c2d63c2ebc0
Status: Downloaded newer image for postgres:latest
cfaaa687ffae51e057868e6b11fa826ef7dc6e6130f1f38658719f6be0fef030
```

```
C:\Users\alefi\myfastapiapp> docker build -t myfastapiapp .
[+] Building 16.2s (11/11) FINISHED                                            docker:desktop-linux
=> [internal] load build definition from Dockerfile                         0.0s
=> => transferring dockerfile: 240B                                         0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim          3.3s
=> [auth] library/python:pull token for registry-1.docker.io                0.0s
=> [internal] load .dockerrcignore                                         0.0s
=> => transferring context: 2B                                           0.0s
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39 0.0s
=> => resolve docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39 0.0s
=> [internal] load build context                                         0.0s
=> => transferring context: 1.07kB                                         0.0s
=> CACHED [2/5] WORKDIR /app                                         0.0s
=> [3/5] COPY requirements.txt .                                         0.1s
=> [4/5] RUN pip install --no-cache-dir -r requirements.txt            10.2s
=> [5/5] COPY . .                                         0.1s
=> exporting to image                                                 2.3s
=> => exporting layers                                         1.4s
=> => exporting manifest sha256:23b305e75b1f70f8df6998ab0affa1a6f8ee3da01da75c3c9af6fb789e8a9 0.0s
=> => exporting config sha256:252f5b25b7d4c3b083898b2f940787e8c9fbb10dc4a23e794e6be390bb088ce 0.0s
=> => exporting attestation manifest sha256:a17875fc76196ec2f34950d5a51389f21adc2e160ce5c23a1 0.0s
=> => exporting manifest list sha256:37a29a1f998c41378e123c01c6b81f64994bcec67f04b2957957324e 0.0s
=> => naming to docker.io/library/myfastapiapp:latest                      0.0s
=> => unpacking to docker.io/library/myfastapiapp:latest                   0.7s
```

```
C:\Users\alefi\myfastapiapp>docker network create mynetwork
8ee7470244fd83da4a176900cecbabe4c8fa32531419dcc14c7c61048f0137b0

C:\Users\alefi\myfastapiapp>docker network connect mynetwork mypostgres

C:\Users\alefi\myfastapiapp>docker run -d --name myfastapi --network mynetwork -e DB_HOST=mypostgres -e DB_NAME=testdb -e DB_USER=testuser -e DB_PASS=testpass -p 8000:8000 myfastapiapp
4e36cd1ccf08d6fd7d1c4e55f06eb7179129652aa7d4a2ae1aff54454a0fd2d5

C:\Users\alefi\myfastapiapp>curl http://localhost:8000/
{"message": "Connected to DB! Time: (datetime.datetime(2025, 8, 24, 9, 2, 56, 871756, tzinfo=datetime.timezone.utc),)"}
C:\Users\alefi\myfastapiapp>
```

## Exercise 8: Docker Compose Basics

Objective: Use Compose to simplify multi-container apps.

- Write a docker-compose.yml for FastAPI + Postgres.
- Use docker compose up, inspect logs and containers.
- Add health checks and environment variables.
- Use depends\_on, restart policies.

```
C:\Users\alefi\myfastapiapp>docker compose up --build
time="2025-08-24T15:19:16+05:30" level=warning msg="C:\\\\Users\\\\alefi\\\\myfastapiapp\\\\docker-compose.yml: the attribute red, please remove it to avoid potential confusion"
#1 [internal] load local bake definitions
#1 reading from stdin 512B done
#1 DONE 0.0s

#2 [internal] load build definition from Dockerfile
#2 transferring dockerfile: 236B 0.0s done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/python:3.9-slim
#3 DONE 1.1s

#4 [internal] load .dockerrcignore
#4 transferring context: 2B done
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 822B 0.0s done
#5 DONE 0.0s

#6 [1/5] FROM docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0f2566e
#6 resolve docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0f2566e6
#6 DONE 0.1s

#7 [2/5] WORKDIR /app
#7 CACHED

#8 [3/5] COPY requirements.txt .
#8 CACHED

#9 [4/5] RUN pip install --no-cache-dir -r requirements.txt
#9 CACHED

#10 [5/5] COPY .
#10 DONE 0.1s

#11 exporting to image
#11 exporting layers 0.1s done
```

```

#11 exporting manifest sha256:bb4d6d1e5145321c8b738ba7beab5faf86c4354026e942bf9b92c880810f62 0.0s done
#11 exporting config sha256:068abecd2ce3c700bc27cca64783e338e3acce6eb0e7f50b94e76d197db894 0.0s done
#11 exporting attestation manifest sha256:798d3dc98200cf62779f37a565496d9aca0a08027c5ee75faacda3f30e2bfb72bb 0.1s done
#11 exporting manifest list sha256:86da357727c0ccfd5c491d051aa355f897d91ba79dc7bbf56bc1c715ce32cdf0 0.0s done
#11 naming to docker.io/library/myfastapiapp-web:latest done
#11 unpacking to docker.io/library/myfastapiapp-web:latest 0.1s done
#11 DONE 0.4s

#12 resolving provenance for metadata file
#12 DONE 0.0s
[+] Running 3/3
  ✓ myfastapiapp-web      Built          0.0s
  ✓ Container postgres_db  Running        0.0s
  ✓ Container fastapi_app Recreated     0.3s
Attaching to fastapi_app, postgres_db
fastapi_app | INFO: Started server process [1]
fastapi_app | INFO: Waiting for application startup.
fastapi_app | INFO: Application startup complete.
fastapi_app | INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
fastapi_app | INFO: Started server process [1]
fastapi_app | INFO: Waiting for application startup.
fastapi_app | INFO: Application startup complete.
fastapi_app | INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
postgres_db | 2025-08-24 09:53:14.683 UTC [62] LOG:  checkpoint starting: time
postgres_db | 2025-08-24 09:53:19.197 UTC [62] LOG:  checkpoint complete: wrote 47 buffers (0.3%); 0 WAL file(s) added, 0 removed, 0 recycled; write=4.492
s, sync=0.009 s, total=4.514 s; sync files=12, longest=0.004 s, average=0.001 s; distance=270 kB; lsn=0/1959248, redo lsn=0/1959188

```

```

C:\Users\alefi\myfastapiapp>docker compose logs web | tail
time="2025-08-24T15:25:07+05:30" level=warning msg="C:\\\\Users\\\\alefi\\\\myfastapiapp\\\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
fastapi_app | INFO: Started server process [1]
fastapi_app | INFO: Waiting for application startup.
fastapi_app | INFO: Application startup complete.
fastapi_app | INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
fastapi_app | INFO: Shutting down
fastapi_app | INFO: Waiting for application shutdown.
fastapi_app | INFO: Application shutdown complete.
fastapi_app | INFO: Finished server process [1]

C:\Users\alefi\myfastapiapp>docker compose logs db | tail
time="2025-08-24T15:25:07+05:30" level=warning msg="C:\\\\Users\\\\alefi\\\\myfastapiapp\\\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
postgres_db | The files belonging to this database system will be owned by user "postgres".
postgres_db | This user must also own the server process.
postgres_db | The database cluster will be initialized with locale "en_US.utf8".
postgres_db | The default database encoding has accordingly been set to "UTF8".
postgres_db | The default text search configuration will be set to "english".
postgres_db | Data page checksums are disabled.
postgres_db | fixing permissions on existing directory /var/lib/postgresql/data ... ok
postgres_db | creating subdirectories ... ok
postgres_db | selecting dynamic shared memory implementation ... posix
postgres_db | selecting default "max_connections" ... 100
postgres_db | selecting default "shared_buffers" ... 128MB
postgres_db | selecting default time zone ... Etc/UTC
postgres_db | creating configuration files ... ok
postgres_db | running bootstrap script ... ok
postgres_db | performing post-bootstrap initialization ... ok
postgres_db | syncing data to disk ... ok
postgres_db |
postgres_db | Success. You can now start the database server using:
postgres_db | pg_ctl -D /var/lib/postgresql/data -l logfile start
postgres_db |

```

## Exercise 9: Healthchecks and Best Practices

Objective: Make robust, production-like Dockerfiles.

- Add HEALTHCHECK instruction to your Dockerfile.
  - Use ENTRYPOINT vs CMD appropriately.
  - Minimize layers and image size (e.g., using alpine).
  - Inspect container health via docker inspect.

## Exercise 10: Debugging, Cleanup & Troubleshooting

Objective: Learn to manage resources and solve issues.

- Run containers with bad commands or missing ports.

- Clean up unused images, containers, volumes with:

```
C:\Users\alefi\myfastapiapp>docker run -d --name nginx_noport nginx
ce48bccced0654e071c73155362792786af7217e587a39da6c9d61d78d513b0a

C:\Users\alefi\myfastapiapp>docker run -d --name nginx_fixed -p 8080:80 nginx
1930ede65bca8fd1bc7226cd3e9f7f0e8d291d0c7e884df5b67b1a2b82e5b47a

C:\Users\alefi\myfastapiapp>docker stop bad_container nginx_noport
bad_container
nginx_noport

C:\Users\alefi\myfastapiapp>docker rm bad_container nginx_noport
bad_container
nginx_noport

C:\Users\alefi\myfastapiapp>docker image prune -a
WARNING! This will remove all images without at least one container associated to them.
Are you sure you want to continue? [y/N] y
Deleted Images:
untagged: ubuntu-tools:1.0
untagged: myfastapiapp:latest
deleted: sha256:37a29a1f998c41378e123c01c6b81f64994bcec67f04b2957957324ea428f766
deleted: sha256:23b305e75b1f70f8df6998ab0affa1a6f8ee3da01da75c3c9af6fb789e8a9c89
deleted: sha256:252f5b25b7d4c3b083898b2f940787e8c9fbb10dc4a23e794e6be390bb088cef
deleted: sha256:a17875fc76196ec2f34950d5a51389f21adc2e160ce5c23a136be7d186287583
deleted: sha256:290f24cc8fc9d4ed76c3bc1f9abc3917d893be8fda0f3bd7822839fec658d94c
deleted: sha256:96a7ee155f6bab8ef12230d8b749a1f94cedda0faf982f37434146fc3e82a229

Total reclaimed space: 10.8kB

C:\Users\alefi\myfastapiapp>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:
8314cb9adbe23b571989ca02c031cd3f1b40042750660c9156ef806c4591c752
2c92d6c9086c16113466ab879235c30668603889e8b51916e3ba699b2d952e45
cc3c9c61b4f5a537e7afb9bd6cde9a24a110e73027d940ca3635d73c06517e0f

Total reclaimed space: 143.9MB

C:\Users\alefi\myfastapiapp>docker system prune -a --volumes
```