



CONTINUOUS ASSESSMENT

LAB 3

CLOUD DATA CENTRES

ZAPPC4202

CREATED BY: SAEDATOUL ASTHEERAH BINTI MISLAN

ID NUMBER: C00290913

YEAR: 3

COURSE: INFORMATION TECHNOLOGY MANAGEMENT

ISSUED YEAR: 2023

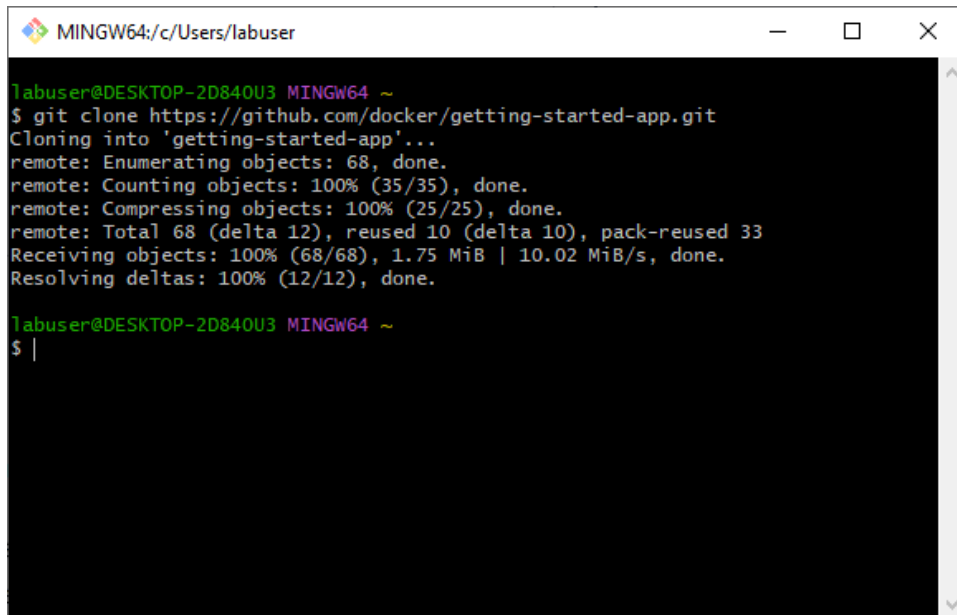
LECTURER: LEI SHI

Contents

Part 2: Containerize an application	3
Part 3: Update the application	12
Part 5: Persist the DB	15
Part 6: Use bind mounts	19
Part 7: Multi container apps	24

Part 2: Containerize an application

Here's get the application source code onto my machine

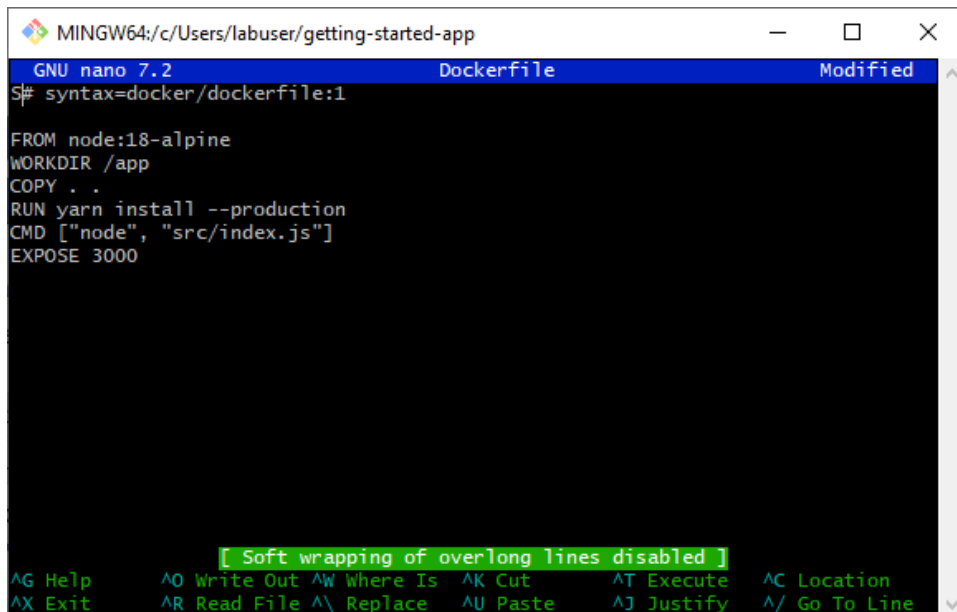


```
MINGW64:/c/Users/labuser

labuser@DESKTOP-2D840U3 MINGW64 ~
$ git clone https://github.com/docker/getting-started-app.git
Cloning into 'getting-started-app'...
remote: Enumerating objects: 68, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 68 (delta 12), reused 10 (delta 10), pack-reused 33
Receiving objects: 100% (68/68), 1.75 MiB | 10.02 MiB/s, done.
Resolving deltas: 100% (12/12), done.

labuser@DESKTOP-2D840U3 MINGW64 ~
$ |
```

Apps of all kinds, both front-end and back-end apps, can be containerised.



```
MINGW64:/c/Users/labuser/getting-started-app
GNU nano 7.2 Dockerfile Modified
$# syntax=docker/dockerfile:1

FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000

[ Soft wrapping of overlong lines disabled ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Now that I have created the Dockerfile, I can build the Docker image by running this command in my project directory:

```
MINGW64:/c/Users/labuser/getting-started-app
labuser@DESKTOP-2D840U3 MINGW64 ~
$ git clone https://github.com/docker/getting-started-app.git
fatal: destination path 'getting-started-app' already exists and is not an empty
directory.

labuser@DESKTOP-2D840U3 MINGW64 ~
$ cd getting-started-app

labuser@DESKTOP-2D840U3 MINGW64 ~/getting-started-app (main)
$ touch Dockerfile

labuser@DESKTOP-2D840U3 MINGW64 ~/getting-started-app (main)
$ nano Dockerfile

labuser@DESKTOP-2D840U3 MINGW64 ~/getting-started-app (main)
$ docker build -t getting-started
ERROR: error during connect: this error may indicate that the docker daemon is n
ot running: Get "http://%2F%2F.%2Fpipe%2Fdocker_engine/_ping": open //./pipe/doc
ker_engine: The system cannot find the file specified.

labuser@DESKTOP-2D840U3 MINGW64 ~/getting-started-app (main)
$ docker build -t getting-started .
ERROR: error during connect: this error may indicate that the docker daemon is n
ot running: Get "http://%2F%2F.%2Fpipe%2Fdocker_engine/_ping": open //./pipe/doc
ker_engine: The system cannot find the file specified.

labuser@DESKTOP-2D840U3 MINGW64 ~/getting-started-app (main)
$ nano Dockerfile

labuser@DESKTOP-2D840U3 MINGW64 ~/getting-started-app (main)
$ docker build -t getting-started .
ERROR: error during connect: this error may indicate that the docker daemon is not runn
ing: Get "http://%2F%2F.%2Fpipe%2Fdocker_engine/_ping": open //./pipe/docker_engine: Th
e system cannot find the file specified.

labuser@DESKTOP-2D840U3 MINGW64 ~/getting-started-app (main)
$ docker build -t getting-started
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage: docker buildx build [OPTIONS] PATH | URL | -
Start a build
```

```
MINGW64/c/Users/Astheerah/getting-started-app

Astheerah@theeslappy MINGW64 ~
$ git clone https://github.com/docker/getting-started-app.git
Cloning into 'getting-started-app'...
remote: Enumerating objects: 68, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 68 (delta 12), reused 10 (delta 10), pack-reused 33
Receiving objects: 100% (68/68), 1.75 MiB | 1.13 MiB/s, done.
Resolving deltas: 100% (12/12), done.

Astheerah@theeslappy MINGW64 ~
$ cd getting-started-app

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ touch Dockerfile

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ nano Dockerfile

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker build -t getting-started
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Start a build

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker run -dp 127.0.0.1:3000:3000 getting-started
Unable to find image 'getting-started:latest' locally
docker: Error response from daemon: pull access denied for getting-started, repository does not exist or
r may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ touch Dockerfile

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ nano Dockerfile

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker run -dp 127.0.0.1:3000:3000 getting-started
Unable to find image 'getting-started:latest' locally
docker: Error response from daemon: pull access denied for getting-started, repository does not exist or
r may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ |
```

Using a text editor or code editor, add the following contents to the Dockerfile.

```
MINGW64/c/Users/Astheerah/getting-started-app

# syntax=docker/dockerfile:1

FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```

Build the image.

The command's '.' specifies the build context, which is the current directory; -t stands for tag and gives the image a name; flask-app is the name we will give the image.

Once I run the command, Docker will build my image, and the output will come out like this:

```
MINGW64/c/Users/Astheerah/getting-started-app
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker build -t getting-started .
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 182B 0.0s done
#1 DONE 0.0s

#2 resolve image config for docker.io/docker/dockerfile:1
#2 ...

#3 [auth] docker/dockerfile:pull token for registry-1.docker.io
#3 DONE 0.0s

#2 resolve image config for docker.io/docker/dockerfile:1
#2 DONE 2.9s

#4 docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f50032edf31be0021
#4 resolve docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f50032edf31be0021 done
#4 sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f50032edf31be0021 8.40kB / 8.40kB done
#4 sha256:657fcc512c7369f4cb3d94ea329150f8daf626bc838b1a1e81f1834c73ecc77e 482B / 482B done
#4 sha256:a17ee7fff8f5e97b974f5b48f51647d2cf28d543f2aa6c11aaa0ea431b44bb89 1.27kB / 1.27kB done
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 0B / 11.80MB 0.1s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 1.05MB / 11.80MB 1.6s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 2.10MB / 11.80MB 1.8s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 4.19MB / 11.80MB 2.1s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 5.24MB / 11.80MB 3.5s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 6.29MB / 11.80MB 3.7s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 8.39MB / 11.80MB 3.9s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 9.44MB / 11.80MB 4.1s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 10.49MB / 11.80MB 4.3s
#4 sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 11.80MB / 11.80MB 5.6s done
#4 extracting sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232
#4 extracting sha256:9d9c93f4b00be908ab694a4df732570bcd3b8a96b7515d70ff93402179ad232 0.2s done
#4 DONE 5.9s

#5 [internal] load metadata for docker.io/library/node:18-alpine
#5 ...

#6 [auth] library/node:pull token for registry-1.docker.io
#6 DONE 0.0s

#5 [internal] load metadata for docker.io/library/node:18-alpine
#5 DONE 3.7s

#7 [internal] load .dockerignore
#7 transferring context: 2B done
#7 DONE 0.0s

#8 [1/4] FROM docker.io/library/node:18-alpine@sha256:0085670310d2879621f96a4216c893f92e2ded827e9e6ef8437672e1bd72f437
#8 resolve docker.io/library/node:18-alpine@sha256:0085670310d2879621f96a4216c893f92e2ded827e9e6ef8437672e1bd72f437 done
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 0B / 40.24MB 0.1s
#8 sha256:3d4a65156edf0208c8421995310d9e662e7ee63e2bcae660efb02f6c4ddef6a9 0B / 2.34MB 0.1s
#8 sha256:0085670310d2879621f96a4216c893f92e2ded827e9e6ef8437672e1bd72f437 1.43kB / 1.43kB done
#8 sha256:aacbcce05180c1dd8c33dba8a9c42b75dbfdd659aa57617497f1ce2c5d83d889 1.16kB / 1.16kB done
#8 sha256:c8eb770fbfacf54104162cc9035c478ddb7d8dc15dca5298af028257f1dbdb3f 7.14kB / 7.14kB done
#8 sha256:4abcf20661432fb2d719aaf90656f55c287f8ca915dc1c92ec14ff61e67fbaf8 0B / 3.41MB 0.1s
```

```
MINGW64:/c:/Users/Astheerah/getting-started-app
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 2.10MB / 40.24MB 4.2s
#8 sha256:4abcf20661432fb2d719aaf90656f55c287f8ca915dc1c92ec14ff61e67fbaf8 3.41MB / 3.41MB 4.1s done
#8 extracting sha256:4abcf20661432fb2d719aaf90656f55c287f8ca915dc1c92ec14ff61e67fbaf8 0.1s
#8 extracting sha256:4abcf20661432fb2d719aaf90656f55c287f8ca915dc1c92ec14ff61e67fbaf8 0.2s done
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 4.19MB / 40.24MB 5.9s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 6.29MB / 40.24MB 6.5s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 8.39MB / 40.24MB 7.9s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 10.49MB / 40.24MB 8.4s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 12.58MB / 40.24MB 8.8s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 14.68MB / 40.24MB 10.3s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 16.78MB / 40.24MB 10.7s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 18.87MB / 40.24MB 12.2s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 20.97MB / 40.24MB 12.8s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 23.07MB / 40.24MB 14.3s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 25.17MB / 40.24MB 14.6s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 27.26MB / 40.24MB 16.6s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 29.36MB / 40.24MB 18.2s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 31.46MB / 40.24MB 18.6s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 33.55MB / 40.24MB 19.0s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 35.65MB / 40.24MB 20.6s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 37.75MB / 40.24MB 21.1s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 39.85MB / 40.24MB 22.4s
#8 sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 40.24MB / 40.24MB 22.5s done
#8 extracting sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97
#8 extracting sha256:eb6c7c29ba4d368f2428cacd291f7821b750fac3b1fb65b937ef855c573cdf97 1.8s done
#8 extracting sha256:3d4a65156edf0208c8421995310d9e662e7ee63e2bcae660efb02f6c4ddef6a9
#8 extracting sha256:3d4a65156edf0208c8421995310d9e662e7ee63e2bcae660efb02f6c4ddef6a9 0.1s done
#8 extracting sha256:5bdb6c27eb32087b71a9dde411c1f1eeb87563c0445f89db4eb7639d2cf50f45 done
#8 DONE 24.7s

#10 [2/4] WORKDIR /app
#10 DONE 0.2s

#11 [3/4] COPY . .
#11 DONE 0.1s

#12 [4/4] RUN yarn install --production
#12 0.508 yarn install v1.22.19
#12 0.603 [1/4] Resolving packages...
#12 0.815 [2/4] Fetching packages...
#12 24.06 [3/4] Linking dependencies...
#12 25.09 [4/4] Building fresh packages...
#12 28.27 Done in 27.76s.
#12 DONE 28.5s

#13 exporting to image
#13 exporting layers
#13 exporting layers 1.1s done
#13 writing image sha256:6cf58e32e5e8e5134c74847f8d61537056a2fccdd63ff120d5fafc94e12211a3 done
#13 naming to docker.io/library/getting-started done
#13 DONE 1.1s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
```

Running the Docker container

Now that the Docker image has been created, you can run the container with the following command in the project directory:

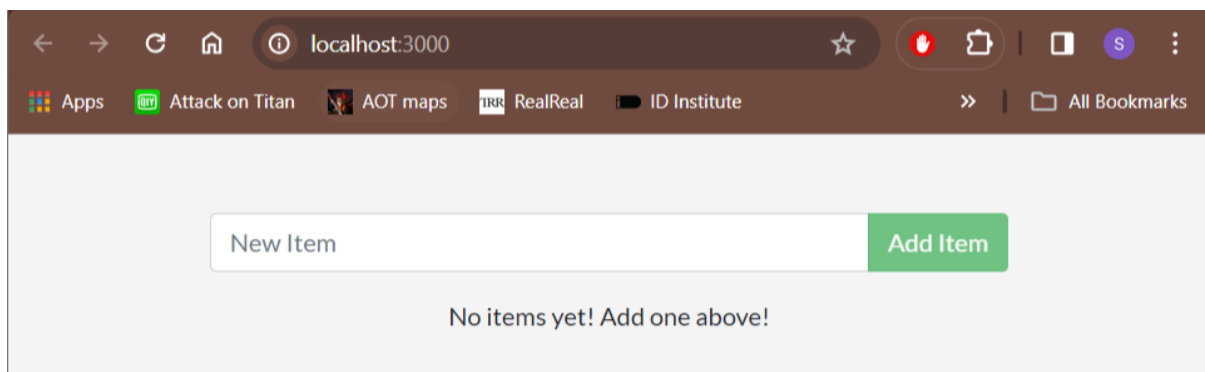
```
docker run -dp 127.0.0.1:3000:3000 getting-started
```

The command mentioned above instructs Docker to run the container in interactive mode, allowing you to communicate with it via its shell and assign a console to it that is text-based. However, since Flask isn't a command-line tool that needs an interactive terminal, using `-it` when launching the application isn't necessarily required. The port mapping of the container is specified by the `-p` flag. The port on the host system that you wish to map to the container port is the first 3000. The cargo port is located at 3000. The picture we just built and wish to execute is called `flask-app`.

```
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker run -dp 127.0.0.1:3000:3000 getting-started
08aae997bd4b044061f136b9c86d792d3e5e936692cfada5466e9fef82cb93c0

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ |
```

And when you browse to <http://localhost:3000>, you will see the very basic Flask application.



Run the following `docker ps` command in a terminal to list your containers.

```
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
08aae997bd4b   getting-started "docker-entrypoint.s..." About a minute ago Up About a minute 127.0.0.1:3000->3000/tcp
quirky_morse

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$
```


Part 3: Update the application

Update the source code

```
MINGW64/c/Users/Astheerah/getting-started-app
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ vi src/static/js/app.js

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ vi src/static/js/app.js

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker build -t getting-started .
2024/02/08 19:42:04 http2: server: error reading preface from client //./pipe/docker_engine: file has a
lready been closed
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 182B done
#1 DONE 0.0s

#2 resolve image config for docker.io/docker/dockerfile:1
#2 ...

#3 [auth] docker/dockerfile:pull token for registry-1.docker.io
#3 DONE 0.0s

#2 resolve image config for docker.io/docker/dockerfile:1
#2 DONE 2.5s

#4 docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f
50032edf31be0021
#4 CACHED

#5 [internal] load metadata for docker.io/library/node:18-alpine
#5 ...

#6 [auth] library/node:pull token for registry-1.docker.io
#6 DONE 0.0s

#5 [internal] load metadata for docker.io/library/node:18-alpine
#5 DONE 1.9s

#7 [internal] load .dockerignore
#7 transferring context: 2B done
#7 DONE 0.0s

#8 [1/4] FROM docker.io/library/node:18-alpine@sha256:0085670310d2879621f96a4216c893f92e2ded827e9e6ef84
37672e1bd72f437
#8 DONE 0.0s

#9 [internal] load build context
#9 transferring context: 10.05kB 0.0s done
#9 DONE 0.0s

#10 [2/4] WORKDIR /app
#10 CACHED

#11 [3/4] COPY . .
#11 DONE 0.1s

#12 [4/4] RUN yarn install --production
#12 0.559 yarn install v1.22.19
#12 0.640 [1/4] Resolving packages...
#12 0.882 [2/4] Fetching packages...

#13 exporting to image
#13 exporting layers
#13 exporting layers 1.1s done
#13 writing image sha256:376d299d2579d4fa83ccba8272348ff5a9585494efe94d2e6a12e7e9f204a9d4 done
#13 naming to docker.io/library/getting-started done
#13 DONE 1.1s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker run -dp 127.0.0.1:3000:3000 getting-started
bdd6d854598aba6e8458781dddf83259181061d6285a38f098fe596b0b1574de
docker: Error response from daemon: driver failed programming external connectivity on endpoint zen_lal
ande (c3f844e9ee0c4ab2c8bf2967b2757f6f600c99322b5d67a3c132eeea1549e070): Bind for 127.0.0.1:3000 failed
: port is already allocated.
```

In the src/static/js/app.js file, update line 56 to use the new empty text. This is to change the "empty text" to "You have no todo items yet! Add one above!"

```
MINGW64/c:/Users/Astheerah/getting-started-app
if (items === null) return 'Loading...';

return (
  <React.Fragment>
    <AddItemForm onNewItem={onNewItem} />
    {items.length === 0 && (
      <p className="text-center">You have no todo items yet! Add one above!</p>
    )}
    {items.map(item => (
      <ItemDisplay
        item={item}
        key={item.id}
        onItemUpdate={onItemUpdate}
        onItemRemoval={onItemRemoval}
      />
    ))}
  </React.Fragment>
);
};

function AddItemForm({ onNewItem }) {
  const { Form, InputGroup, Button } = ReactBootstrap;

  const [newItem, setNewItem] = React.useState('');
  const [submitting, setSubmitting] = React.useState(false);

  const submitNewItem = e => {
    e.preventDefault();
    setSubmitting(true);
    fetch('/items', {
      method: 'POST',
      body: JSON.stringify({ name: newItem }),
      headers: { 'Content-Type': 'application/json' },
    })
      .then(r => r.json())
      .then(item => {
        onNewItem(item);
        setSubmitting(false);
        setNewItem('');
      });
  };

  return (
    <Form onSubmit={submitNewItem}>
      <InputGroup className="mb-3">
        <Form.Control
          value={newItem}
          onChange={e => setNewItem(e.target.value)}
          type="text"
          placeholder="New Item"
          aria-describedby="basic-addon1"
        />
        <InputGroup.Append>
          <Button
            type="submit"
            variant="success"
            disabled={!newItem.length}
            className={submitting ? 'disabled' : ''}
          />
        </InputGroup.Append>
      </Form>
    </Form>
  );
};

src/static/js/app.js[+] [dos] (19:39 08/02/2024) 56,86 40%
-- INSERT --
```

Remove a container using the CLI and start the updated app container

Updating Docker images essentially involves deleting the previous image and launching fresh containers using the updated version of the image. Maintaining a proactive approach and consistently updating your Docker images guarantees that your apps flourish in a constantly evolving setting.

```
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
08aae997bd4b   6cf58e32e5e8                        "docker-entrypoint.s..." 15 minutes ago Up 15 minutes 127.0.0.1:3000->3000/tcp
quirky_morse

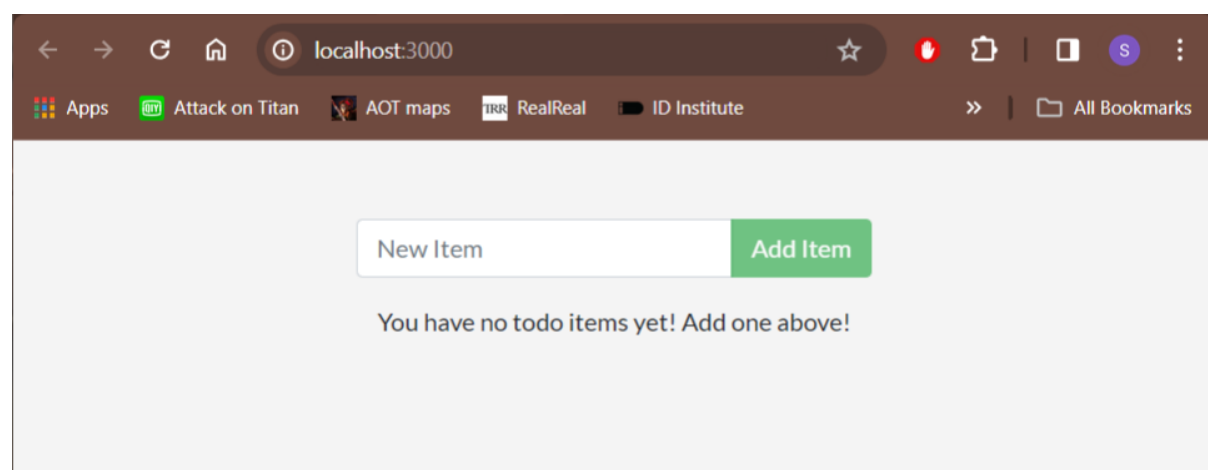
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker stop <the-container-id>

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker stop 08aae997bd4b
08aae997bd4b

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker rm 08aae997bd4b
08aae997bd4b

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker run -dp 127.0.0.1:3000:3000 getting-started
dacb4bfef7e8b31ad6ca11e51b202c017b03fc5da3108b3d801108e7a803283c

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ |
```



Part 5: Persist the DB

A container's filesystem is made up of the different layers from the image when it is operating. Also, each container has a dedicated "scratch space" for adding, deleting, and updating files. Even if two containers use the same image, none of the changes will be visible to the other.

The container's filesystem

```
MINGW64/c/Users/Astheerah/getting-started-app
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker run -d ubuntu bash -c "shuf -i 1-10000 -n 1 -o /data.txt && tail -f /dev/null"
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
57c139bbda7e: Pulling fs layer
57c139bbda7e: Download complete
57c139bbda7e: Pull complete
Digest: sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999
Status: Downloaded newer image for ubuntu:latest
cce8507f7af0340e4247fbb82913f124d3eb7fdc8fd7b374bd4ad57d5b6835bd

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker exec <container-id> cat /data.txt
bash: container-id: No such file or directory

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
cce8507f7af0   ubuntu        "bash -c 'shuf -i 1-..." 50 seconds ago Up 46 seconds
dadb4bfef7e8   quirky_ride   "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes  127.0.0.1:30
00->3000/tcp   getting-started fervent_aryabhata

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker exec cce8507f7af0 cat /data.txt
cat: 'C:/Program Files/Git/data.txt': No such file or directory

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker run -it ubuntu ls /
the input device is not a TTY. If you are using mintty, try prefixing the command with 'winpty'
```

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>docker exec cce8507f7af0 cat /data.txt
4103

C:\Windows\System32>docker run -it ubuntu ls /
bin    dev    home  lib32  libx32  mnt    proc  run    srv    tmp    var
boot  etc    lib   lib64  media   opt    root /sbin  sys    usr

C:\Windows\System32>docker rm -f cce8507f7af0
cce8507f7af0
```

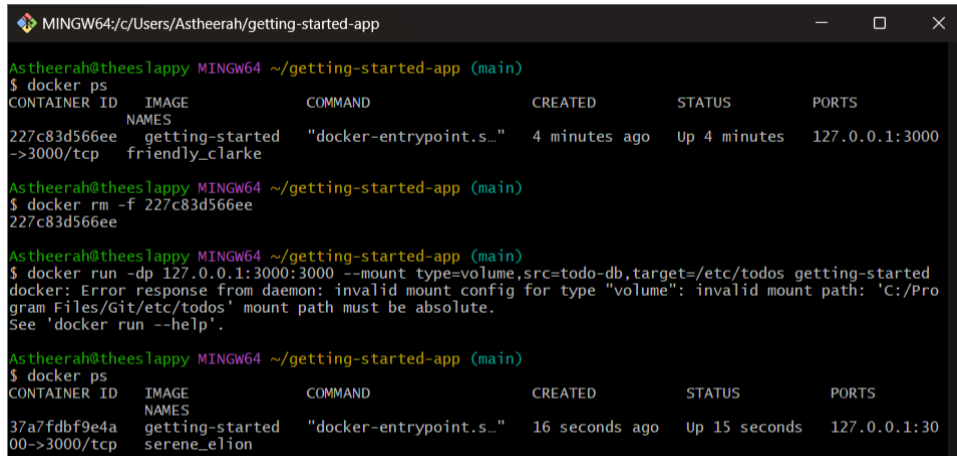
Container volumes

Docker isolates all changes made to a container, even if it has the ability to create, update, and delete files. These changes are lost when the container is removed. You may adjust all of this using volumes.

Persist the todo data

Create a volume and start the container

Since the todo app container is still operating without using the persistent volume, stop and delete it again with `docker rm -f <id>`.



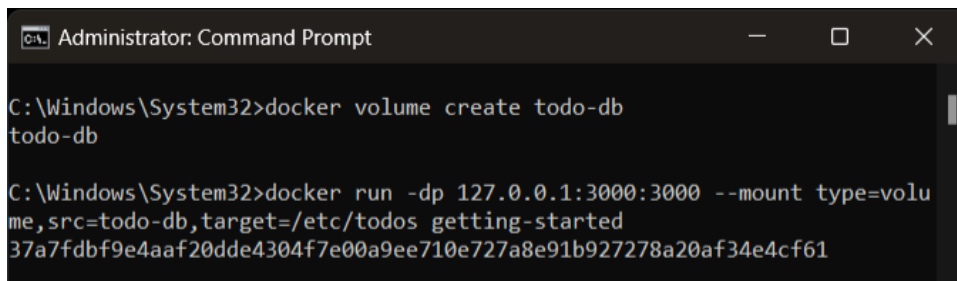
```
MINGW64/c/Users/Astheerah/getting-started-app
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
227c83d566ee   getting-started  "docker-entrypoint.s..." 4 minutes ago  Up 4 minutes  127.0.0.1:3000->3000/tcp
friendly_clarke

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker rm -f 227c83d566ee
227c83d566ee

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker run -dp 127.0.0.1:3000:3000 --mount type=volume,src=todo-db,target=/etc/todos getting-started
docker: Error response from daemon: invalid mount config for type "volume": invalid mount path: 'C:/Program Files/Git/etc/todos' mount path must be absolute.
See 'docker run --help'.

Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
37a7fdbf9e4a   getting-started  "docker-entrypoint.s..." 16 seconds ago  Up 15 seconds  127.0.0.1:3000->3000/tcp
serene_elion
```

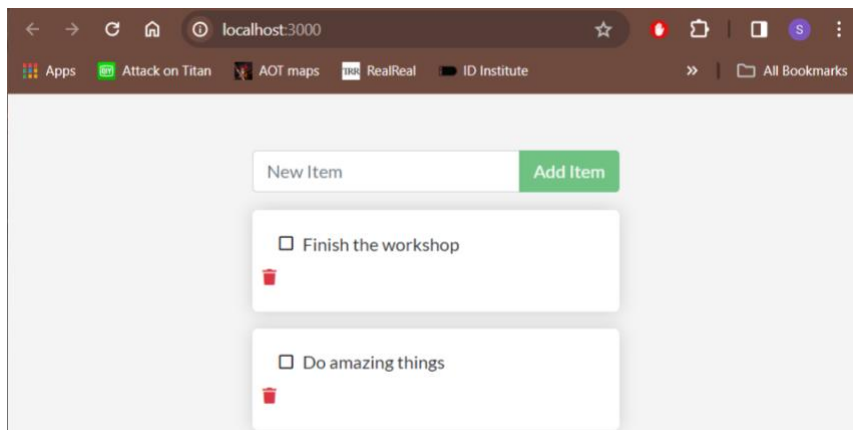
Create a volume by using the docker volume create command.



```
Administrator: Command Prompt
C:\Windows\System32>docker volume create todo-db
todo-db

C:\Windows\System32>docker run -dp 127.0.0.1:3000:3000 --mount type=volume,src=todo-db,target=/etc/todos getting-started
37a7fdbf9e4aaf20dde4304f7e00a9ee710e727a8e91b927278a20af34e4cf61
```


Once the container starts up, open the app and add a few items to your todo list.



```
Administrator: Command Prompt

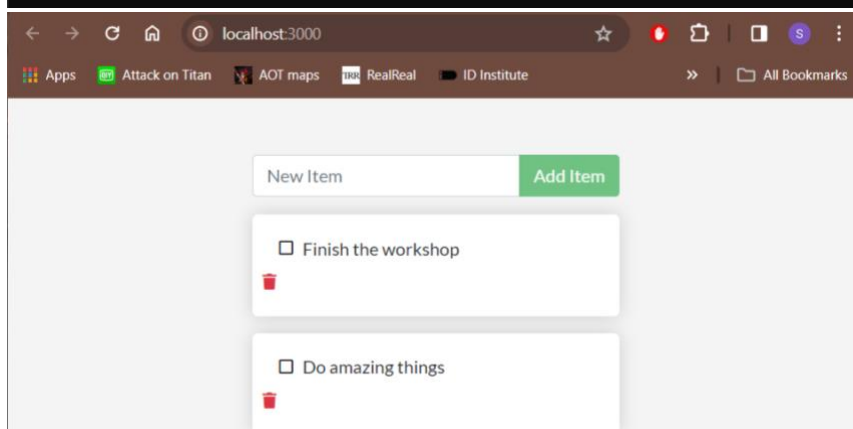
C:\Windows\System32>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS        PORTS          NAMES
9d83916b6cf1   getting-started "docker-entrypoint.s..." 36 seconds ago
Up 35 seconds  127.0.0.1:3000->3000/tcp  vigilant_satoshi

C:\Windows\System32>docker rm -f 9d83916b6cf1
9d83916b6cf1

C:\Windows\System32>docker volume create todo-db
todo-db

C:\Windows\System32>docker run -dp 127.0.0.1:3000:3000 --mount type=volume,src=todo-db,target=/etc/todos getting-started
Unable to find image 'getting-started' locally
docker: Error response from daemon: pull access denied for getting-started, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.

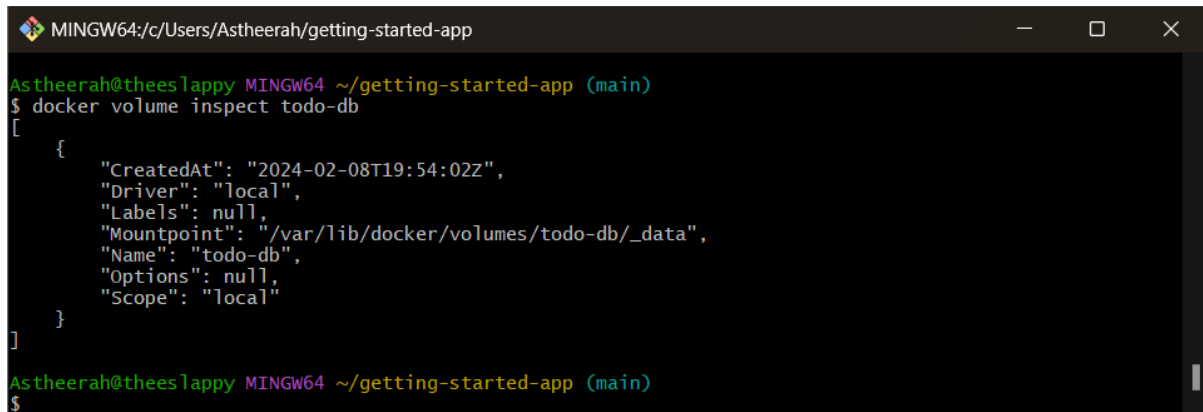
C:\Windows\System32>docker run -dp 127.0.0.1:3000:3000 --mount type=volume,src=todo-db,target=/etc/todos getting-started
f33829d3308a68a0fe742f855691e5a4c9434d72f478058ee87540dba421e9b2
```



Dive into the volume

Data that is accessed by Docker containers is stored persistently in a Docker volume. When the container is removed, this data is still there because it is saved outside of the containers.

This enables you to keep data created and utilised by your containers around, even in the event that the container is removed.

A terminal window titled 'MINGW64:/c/Users/Astheerah/getting-started-app' with standard window controls. The prompt is 'Astheerah@theeslappy MINGW64 ~/getting-started-app (main)'. The command '\$ docker volume inspect todo-db' has been executed, resulting in a JSON array output. The output shows a single volume object with fields: 'CreatedAt' (2024-02-08T19:54:02Z), 'Driver' (local), 'Labels' (null), 'Mountpoint' (/var/lib/docker/volumes/todo-db/_data), 'Name' (todo-db), 'Options' (null), and 'Scope' (local).

```
MINGW64:/c/Users/Astheerah/getting-started-app
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$ docker volume inspect todo-db
[
  {
    "CreatedAt": "2024-02-08T19:54:02Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/todo-db/_data",
    "Name": "todo-db",
    "Options": null,
    "Scope": "local"
  }
]
Astheerah@theeslappy MINGW64 ~/getting-started-app (main)
$
```

Part 6: Use bind mounts

Trying out bind mounts

I had an issue in this part where 'touch' command doesn't have permissions.

```
root@1583f048117c: /src
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>docker run -it --mount "type=bind,src=%cd%,target=/src" ubuntu bash
root@1583f048117c:/# pwd
/
root@1583f048117c:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  src  sys  usr
boot  etc  lib  lib64  media  opt  root  sbin  srv  tmp  var
root@1583f048117c:/# cd src
root@1583f048117c:/src# ls
'$Acer$.cmd'
0409
07409496-a423-4a3e-b620-2cfb01a9318d_HyperV-ComputeNetwork.dll
0ae3b998-9a38-4b72-a4c4-06849441518d_Servicing-Stack.dll
4545ffe2-0dc4-4df4-9d02-299ef204635e_hvsocket.dll
69fe178f-26e7-43a9-aa7d-2b616b672dde_eventlogservice.dll
6bea57fb-8dfb-4177-9ae8-42e8b3529933_RuntimeDeviceInstall.dll
@AdvancedKeySettingsNotification.png
@AppHelpToast.png
@AudioToastIcon.png
@BackgroundAccessToastIcon.png
@EnrollmentToastIcon.png
@StorageSenseToastIcon.png
@VpnToastIcon.png
@WLOGO_96x96.png
@WindowsHelloFaceToastIcon.png
@WindowsUpdateToastIcon.contrast-black.png
@WindowsUpdateToastIcon.contrast-white.png
```

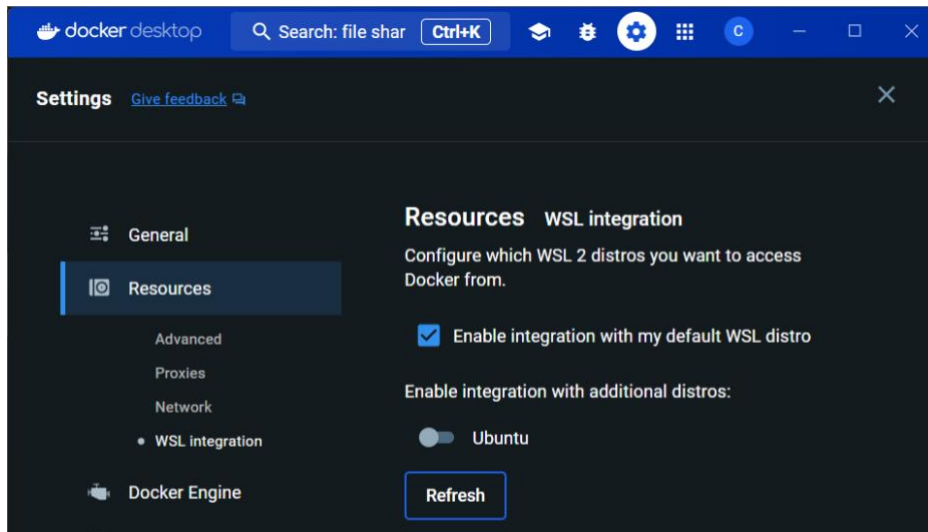
```
root@1583f048117c: /src
xactengine3_4.dll
xactengine3_5.dll
xactengine3_6.dll
xactengine3_7.dll
xboxgipsvc.dll
xboxgipsynthetic.dll
xcopy.exe
xinput1_1.dll
xinput1_2.dll
xinput1_3.dll
xmlfilter.dll
xmllite.dll
xmlprovi.dll
xolehlp.dll
xpspushlayer.dll
xpsservices.dll
xwizard.dtd
xwizard.exe
xwizards.dll
xwreg.dll
xwtpdui.dll
xwtpw32.dll
zh-CN
zh-TW
zipcontainer.dll
zipfldr.dll
ztrace_maps.dll
root@1583f048117c:/src# touch myfile.txt
touch: cannot touch 'myfile.txt': Permission denied
root@1583f048117c:/src#
```

I figured out that I didn't have WSL in my Docker Desktop, so I had to install it manually.

```
unix@theeslappy: ~  
C:\Windows\System32>wsl --install  
Installing: Ubuntu  
Ubuntu has been installed.  
Launching Ubuntu...  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows  
username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: unix  
New password:  
Retype new password:  
passwd: password updated successfully  
Installation successful!  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.133.1-microsoft-standard-WSL2 x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
This message is shown once a day. To disable it please create the  
/home/unix/.hushlogin file.  
unix@theeslappy:~$ wsl --set-version  
Command 'wsl' not found, but can be installed with:  
sudo apt install wsl  
unix@theeslappy:~$ wsl -l -v
```

```
unix@theeslappy: ~  
See "man sudo_root" for details.  
  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.133.1-microsoft-standard-WSL2 x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
This message is shown once a day. To disable it please create the  
/home/unix/.hushlogin file.  
unix@theeslappy:~$ wsl --set-version  
Command 'wsl' not found, but can be installed with:  
sudo apt install wsl  
unix@theeslappy:~$ wsl -l -v  
Command 'wsl' not found, but can be installed with:  
sudo apt install wsl  
unix@theeslappy:~$ sudo apt install wsl  
[sudo] password for unix:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
E: Unable to locate package wsl  
unix@theeslappy:~$ wsl.exe -l -v  
NAME                STATE      VERSION  
* docker-desktop    Running    2  
  Ubuntu            Running    2  
  docker-desktop-data Running    2  
unix@theeslappy:~$
```

Here's the outcome:



```
root@380b8b74dd58: /src
C:\Windows\System32>docker run -it --mount "type=bind,src=%cd%,target=/src" ubuntu bash
root@380b8b74dd58: /# pwd
/
root@380b8b74dd58: /# ls
bin  dev  home  lib32  libx32  mnt  proc  run  src  sys  usr
boot  etc  lib  lib64  media  opt  root  sbin  srv  temp  var
root@380b8b74dd58: /# cd src
root@380b8b74dd58: /src# ls
```

```
Select root@380b8b74dd58: /src
MtcModel.dll
MtfDecoder.dll
MuiUnattend.exe
MultiDigiMon.exe
MusUpdateHandlers.dll
Mystify.scr
NAPCRYPT.DLL
NDKPerfCmd.exe
NDKPing.exe
NETSTAT.EXE
NFCProvisioningPlugin.dll
NPSM.dll
NPSMDesktopProvider.dll
```

```
root@380b8b74dd58: /src
root@380b8b74dd58: /src# touch myfile.txt
root@380b8b74dd58: /src#
```

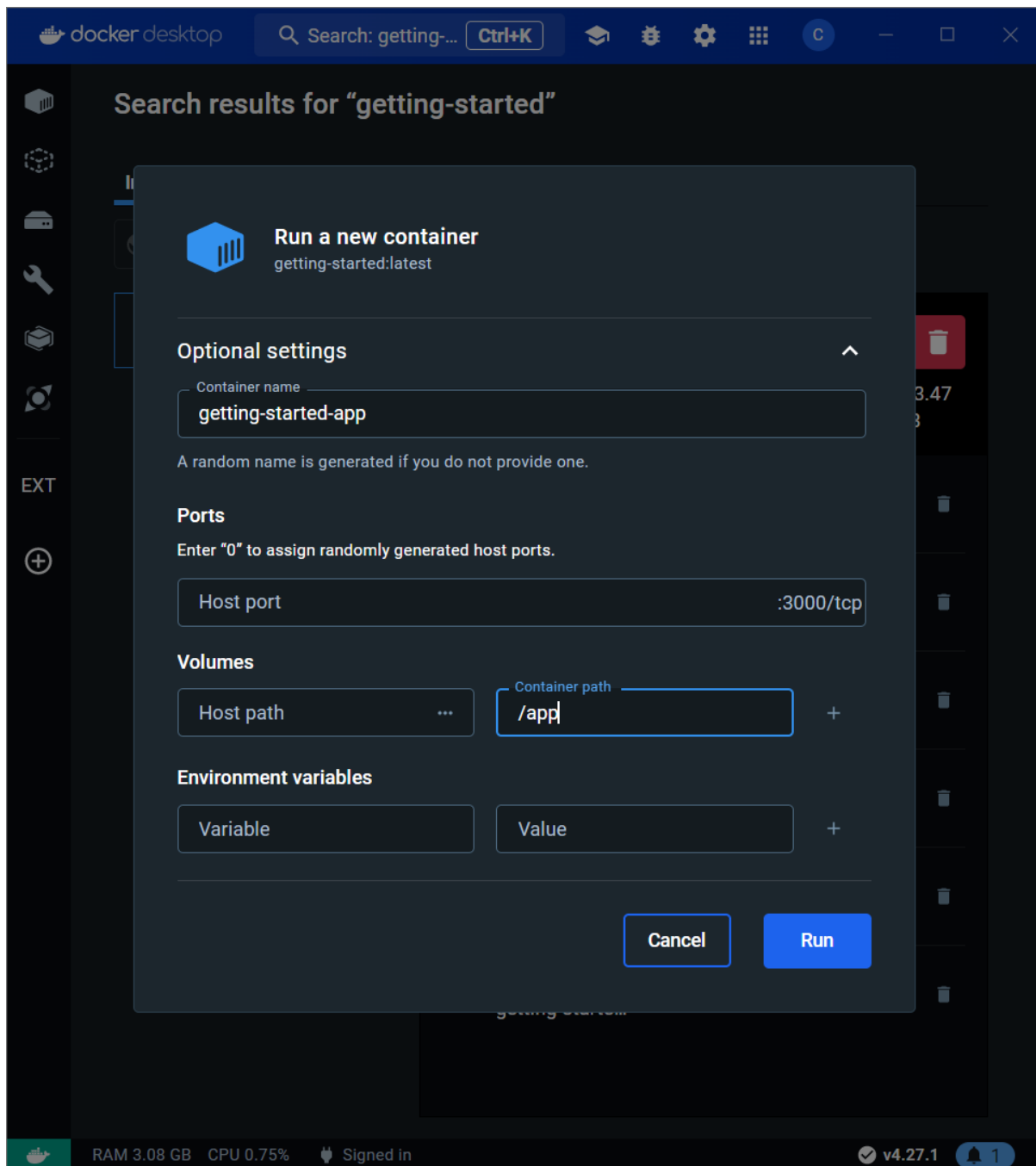
```
Select root@380b8b74dd58: /src
myfile.txt
navshutdown.dll
nbtstat.exe
ncbservice.dll
nci.dll
ncobjapi.dll
ncpa.cpl
ncrypt.dll
ncryptprov.dll
ncryptsslp.dll
ncsi.dll
ncuprov.dll
```

```
root@380b8b74dd58: /src
root@380b8b74dd58: /src# rm myfile.txt
root@380b8b74dd58: /src#
```

```
root@380b8b74dd58: /src
mtxex.dll
mtxoci.dll
muifontsetup.dll
museuxdocked.dll
mycomput.dll
mydocs.dll
navshutdown.dll
nbtstat.exe
ncbservice.dll
nci.dll
ncobjapi.dll
ncpa.cpl
ncrypt.dll
ncryptprov.dll
ncryptsslp.dll
ncsi.dll
ncuprov.dll
ndadmin.exe
nddeapi.dll
ndfapi.dll
ndfapi.dll
ndfapi.dll
ndfhcdiscovery.dll
ndishc.dll
ndproxystub.dll
nduprov.dll
negoexts.dll
net.exe
net1.exe
netapi32.dll
netbios.dll
netbtugc.exe
netcenter.dll
netcfg.exe
netcfgx.dll
netcorehc.dll
netdiagfx.dll
netevent.dll
netfxperf.dll
neth.dll
netid.dll
netiohelp.dll
netioug.exe
netjoin.dll
netlogon.dll
netman.dll
netmsg.dll
netplwiz.dll
netprofm.dll
netprofsvc.dll
netprovfw.dll
netprovisionsp.dll
```

Part 7: Multi container apps

I didn't get to do this part and onwards.




```
Administrator: Command Prompt

C:\Windows\System32>docker run -dp 127.0.0.1:3000:3000
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Create and run a new container from an image

C:\Windows\System32>docker run -d ^
More?      --network todo-app --network-alias mysql ^
More?      -v todo-mysql-data:/var/lib/mysql ^
More?      -e MYSQL_ROOT_PASSWORD=secret ^
More?      -e MYSQL_DATABASE=todos ^
More?      mysql:8.0
69851f2b855b3caf22962015f45af95fa04f3277ed08f8503db7ff9532d8ada7

C:\Windows\System32>docker exec -it 69851f2b855b3caf22962015f45af95fa04f3277ed08f8503db7ff9532d8ada7 mys
ql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

C:\Windows\System32>docker exec -it 69851f2b855b3caf22962015f45af95fa04f3277ed08f8503db7ff9532d8ada7 mys
ql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

C:\Windows\System32>docker exec -it 69851f2b855b3caf22962015f45af95fa04f3277ed08f8503db7ff9532d8ada7 mys
ql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

C:\Windows\System32>docker exec -it 69851f2b855b3caf22962015f45af95fa04f3277ed08f8503db7ff9532d8ada7 mys
ql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

C:\Windows\System32>ROOT
'ROOT' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32>root
'root' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32>docker exec -it 69851f2b855b3caf22962015f45af95fa04f3277ed08f8503db7ff9532d8ada7 mys
ql -u root
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)

C:\Windows\System32>sudo docker exec -it 69851f2b855b3caf22962015f45af95fa04f3277ed08f8503db7ff9532d8ada
7 mysql -u root
'sudo' is not recognized as an internal or external command,
```