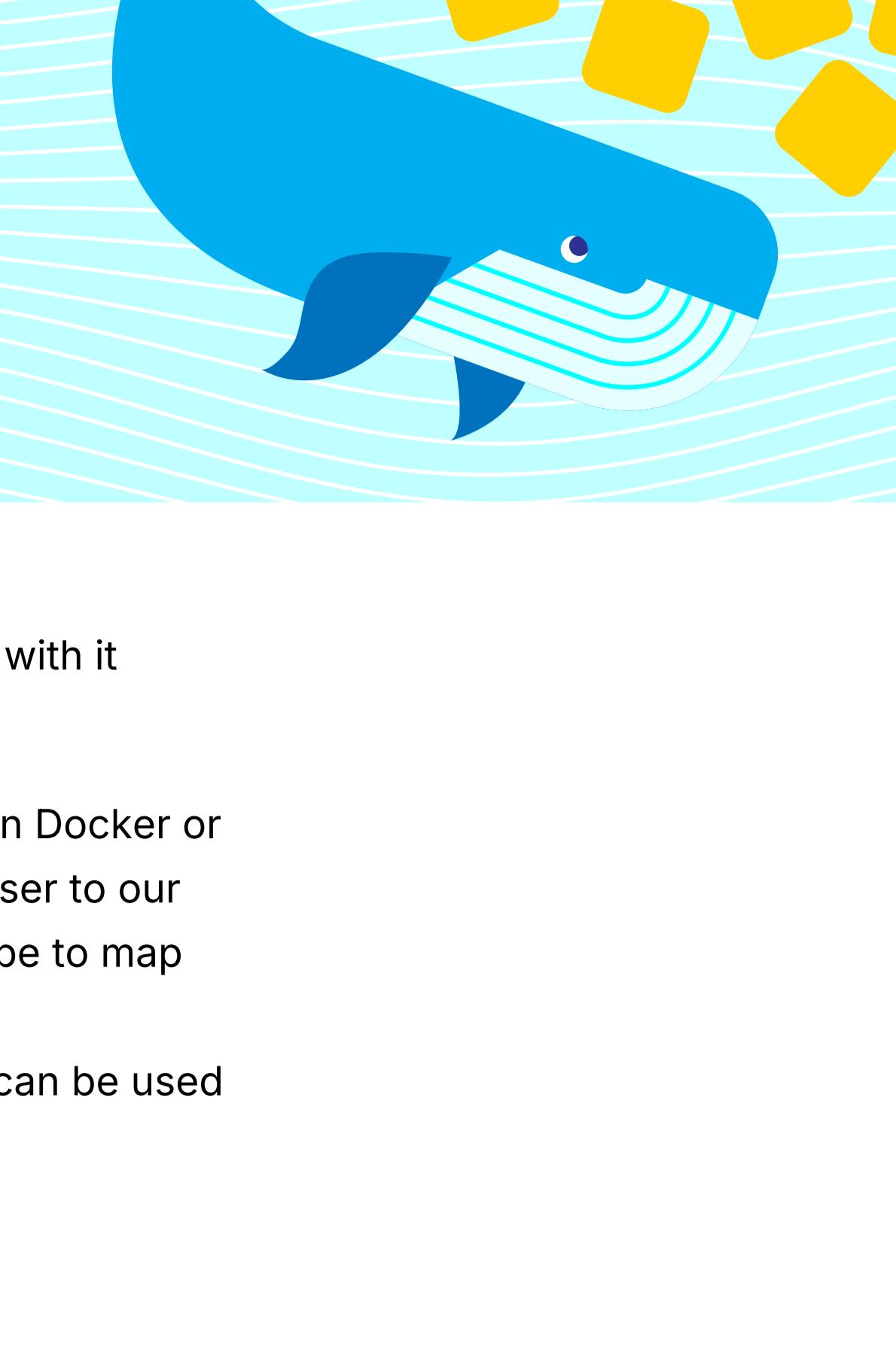


Chapter 4/Using-A-Non-Root-User

Using a non-root user



Let's get back to the yt-dlp application, that we for last time worked with it [Chapter 3](#).

The application could, in theory, escape the container due to a bug in Docker or Linux Kernel. To mitigate this security issue we will add a non-root user to our container and run our process with that user. Another option would be to map the root user to a high, non-existing user id on the host with <https://docs.docker.com/engine/security/users-remap/> which can be used in case you must use root within the container.

The Dockerfile that we did in [Chapter 2](#) was this:

```
FROM ubuntu:24.04
WORKDIR /mydir
RUN apt-get update && apt-get install -y curl python3 ffmpeg
RUN curl -L https://github.com/yt-dlp/yt-dlp/releases/latest/download/yt-dlp -o /usr/local/bin/yt-dlp
RUN chmod a+x /usr/local/bin/yt-dlp
ENTRYPOINT ["/usr/local/bin/yt-dlp"]
```

We will add a user called `appuser` with the following command

```
RUN useradd -m appuser
```

After that we change the user with the directive `USER` - so all commands after this line will be executed as our new user, including the `CMD` and `ENTRYPOINT`.

```
FROM ubuntu:24.04
WORKDIR /mydir
RUN apt-get update && apt-get install -y curl python3 ffmpeg
RUN curl -L https://github.com/yt-dlp/yt-dlp/releases/latest/download/yt-dlp -o /usr/local/bin/yt-dlp
RUN chmod a+x /usr/local/bin/yt-dlp

RUN useradd -m appuser
USER appuser
ENTRYPOINT ["/usr/local/bin/yt-dlp"]
```

When we build and run this image without bind mounting our local directory:

```
$ docker build -t yt-dlp .
$ docker run yt-dlp https://www.youtube.com/watch?v=Xsq1HHTGQrw
...
[info] Xsq1HHTGQrw: Downloading 1 format(s): 137+251
[download] Sleeping 4.00 seconds as required by the site...
[download] Unable to open file: [Errno 13] Permission denied: 'Master's Programme in Computer Science | University of Helsinki [Xsq1HHTGQrw].f137.mp4.part'. Retrying (1/3)...
[download] Unable to open file: [Errno 13] Permission denied: 'Master's Programme in Computer Science | University of Helsinki [Xsq1HHTGQrw].f137.mp4.part'. Retrying (2/3)...
[download] Unable to open file: [Errno 13] Permission denied: 'Master's Programme in Computer Science | University of Helsinki [Xsq1HHTGQrw].f137.mp4.part'. Retrying (3/3)...

ERROR: unable to open for writing: [Errno 13] Permission denied: 'Master's Programme in Computer Science | University of Helsinki [Xsq1HHTGQrw].f137.mp4.part'
...
...
```

We can see that our `appuser` user has no access to write to the container filesystem. This can be fixed with `chown` or not fix it at all, if the intended usage is to always have a `/mydir` mounted from the host. By mounting the directory the application works as intended.

If we want to give the `appuser` permission to write inside the container, the permission change must be done when we are still executing as root, that is, before the directive `USER` is used to change the user:

```
FROM ubuntu:24.04
...
WORKDIR /mydir
# create the appuser
RUN useradd -m appuser
```

```
# change the owner of current dir to appuser
RUN chown appuser .
```

```
# now we can change the user
USER appuser
```

```
ENTRYPOINT ["/usr/local/bin/yt-dlp"]
```

Exercise:

3.5. Optimized project (mandatory exercise)

TRIES

POINTS

1/1

Your answer has been reviewed and graded. New submissions are no longer allowed.

Instructions

In exercises [1.12](#) and [1.13](#) we created Dockerfiles for both `frontend` and `backend`.

Security issues with the user being a root are serious for the example frontend and backend as the containers for web services are supposed to be accessible through the Internet.

Make sure the containers start their processes as non-root user.

The backend image is based on [Alpine Linux](#), which does not support the command `useradd`. Google will surely help you a way to create a user in an `alpine` based image.

Submit the Dockerfiles as answer.

Dockerfile for the frontend:

```
FROM node:16
WORKDIR /usr/src/app
COPY --chown=node:node .
RUN npm install && npm install -g serve
ENV REACT_APP_BACKEND_URL=http://localhost:8080/api
RUN npm run build
USER node
CMD ["serve", "-s", "-l", "3000", "build"]
```

Dockerfile for the backend:

```
FROM golang:1.16
WORKDIR /usr/src/app
RUN addgroup --system backendgroup && adduser --system backenduser --ingroup backendgroup
COPY --chown=backenduser .
RUN go build
RUN go test ./...
ENV PORT=80
USER backenduser
CMD ["./server"]
```

Word count: 69

↑ Chapter front page

You've reached the end of this topic.

Proceed to the next topic

Next page:

Optimizing the image size

