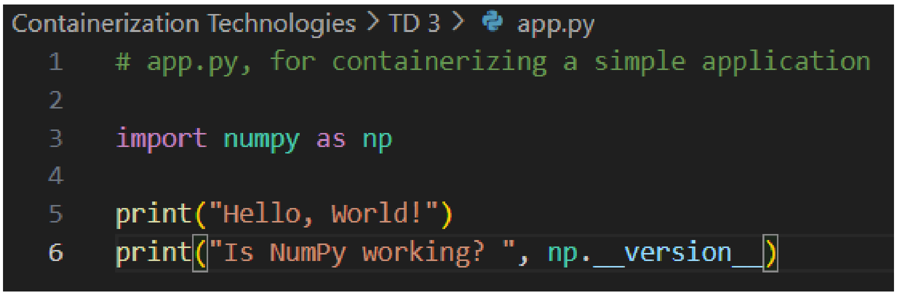
LE QUERNEC Loévan CDOF2

Containerization Technologies – TD 3

Create the script

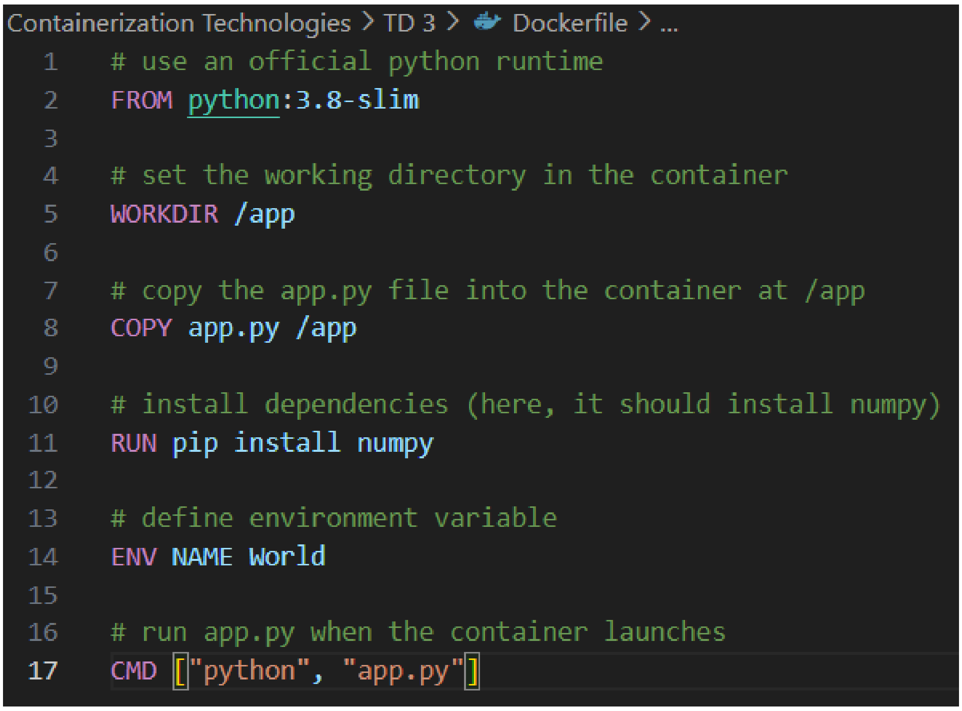
First, we create a simple python script that will be the base of our application:



```python
# app.py, for containerizing a simple application

import numpy as np

print("Hello, World!")
print("Is NumPy working? ", np.__version__)
```

Create the Docker file

As usual, we create a Dockerfile that package our application into a container:

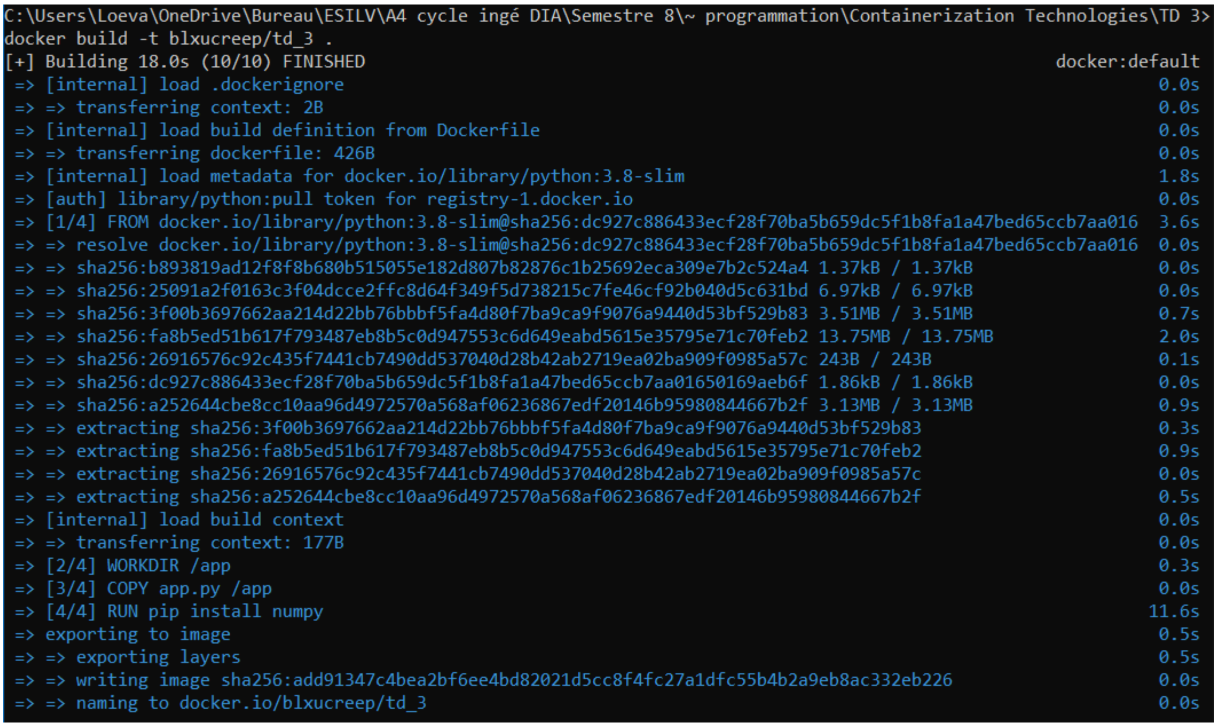


```dockerfile
# use an official python runtime
FROM python:3.8-slim

# set the working directory in the container
WORKDIR /app

# copy the app.py file into the container at /app
COPY app.py /app

# install dependencies (here, it should install numpy)
RUN pip install numpy

# define environment variable
ENV NAME World

# run app.py when the container launches
CMD ["python", "app.py"]
```

Build our image

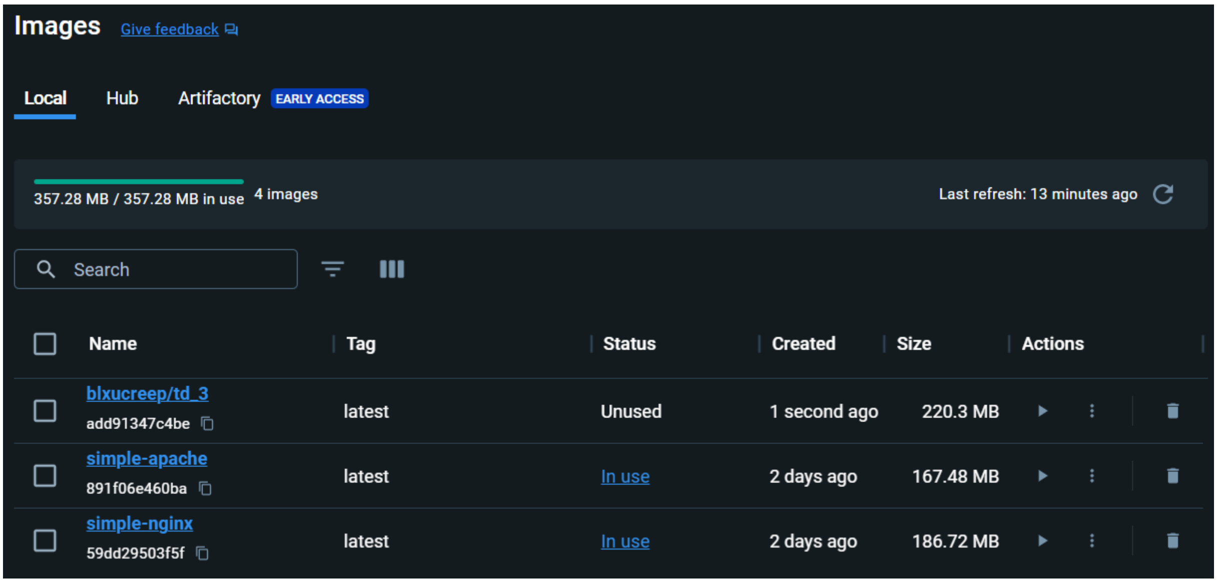
Here, we build our image in the current directory (‘.’ at the end) with a chosen name (‘-t blxucreep/td_3).

The command: ‘**docker build -t blxucreep/td_3**’.



```
C:\Users\Loeva\OneDrive\Bureau\ESILV\A4 cycle ingé DIA\Semestre 8\~ programmation\Containerization Technologies\TD 3>
docker build -t blxucreep/td_3 .
[+] Building 18.0s (10/10) FINISHED                                                                   docker:default
 => [internal] load .dockerignore                                                                               0.0s
 => => transferring context: 2B                                                                                 0.0s
 => [internal] load build definition from Dockerfile                                                            0.0s
 => => transferring dockerfile: 426B                                                                            0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim                                              1.8s
 => [auth] library/python:pull token for registry-1.docker.io                                                   0.0s
 => [1/4] FROM docker.io/library/python:3.8-slim@sha256:dc927c886433ecf28f70ba5b659dc5f1b8fa1a47bed65ccb7aa016  3.6s
 => => resolve docker.io/library/python:3.8-slim@sha256:dc927c886433ecf28f70ba5b659dc5f1b8fa1a47bed65ccb7aa016  0.0s
 => => sha256:b893819ad12f8f8b680b515055e182d807b82876c1b25692eca309e7b2c524a4 1.37kB / 1.37kB                  0.0s
 => => sha256:25091a2f0163c3f04dcce2ffc8d64f349f5d738215c7fe46cf92b040d5c631bd 6.97kB / 6.97kB                  0.0s
 => => sha256:3f00b3697662aa214d22bb76bbbf5fa4d80f7ba9ca9f9076a9440d53bf529b83 3.51MB / 3.51MB                  0.7s
 => => sha256:fa8b5ed51b617f793487eb8b5c0d947553c6d649eabd5615e35795e71c70feb2 13.75MB / 13.75MB                2.0s
 => => sha256:26916576c92c435f7441cb7490dd537040d28b42ab2719ea02ba909f0985a57c 243B / 243B                      0.1s
 => => sha256:dc927c886433ecf28f70ba5b659dc5f1b8fa1a47bed65ccb7aa01650169aeb6f 1.86kB / 1.86kB                  0.0s
 => => sha256:a252644cbe8cc10aa96d4972570a568af06236867edf20146b95980844667b2f 3.13MB / 3.13MB                  0.9s
 => => extracting sha256:3f00b3697662aa214d22bb76bbbf5fa4d80f7ba9ca9f9076a9440d53bf529b83                       0.3s
 => => extracting sha256:fa8b5ed51b617f793487eb8b5c0d947553c6d649eabd5615e35795e71c70feb2                       0.9s
 => => extracting sha256:26916576c92c435f7441cb7490dd537040d28b42ab2719ea02ba909f0985a57c                       0.0s
 => => extracting sha256:a252644cbe8cc10aa96d4972570a568af06236867edf20146b95980844667b2f                       0.5s
 => [internal] load build context                                                                               0.0s
 => => transferring context: 177B                                                                               0.0s
 => [2/4] WORKDIR /app                                                                                          0.3s
 => [3/4] COPY app.py /app                                                                                      0.0s
 => [4/4] RUN pip install numpy                                                                                11.6s
 => exporting to image                                                                                          0.5s
 => => exporting layers                                                                                         0.5s
 => => writing image sha256:add91347c4bea2bf6ee4bd82021d5cc8f4fc27a1dfc55b4b2a9eb8ac332eb226                    0.0s
 => => naming to docker.io/blxucreep/td_3                                                                       0.0s
```

As usual, we can check our image in Docker directly:



357.28 MB / 357.28 MB in use 4 images

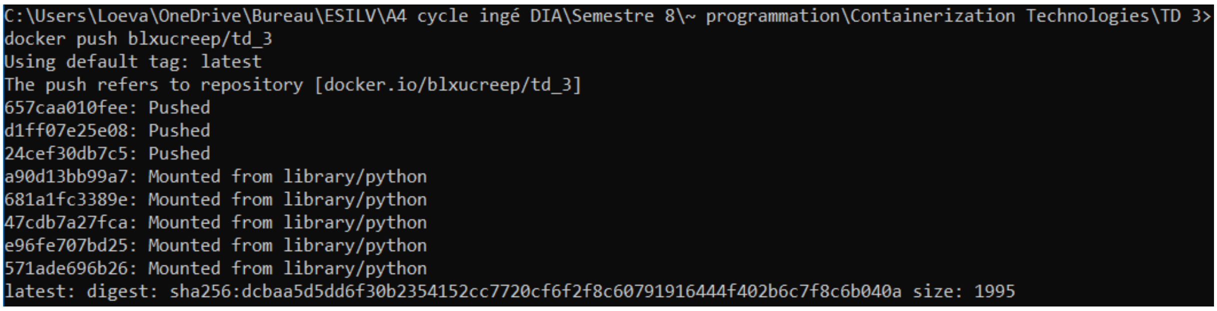
Last refresh: 13 minutes ago

☐	Name	Tag	Status	Created	Size	Actions
☐	blxucreep/td_3 add91347c4be	latest	Unused	1 second ago	220.3 MB	▶ ⋮ 🗑
☐	simple-apache 891f06e460ba	latest	In use	2 days ago	167.48 MB	▶ ⋮ 🗑
☐	simple-nginx 59dd29503f5f	latest	In use	2 days ago	186.72 MB	▶ ⋮ 🗑

Push our container

Before doing anything, make sure we are connected to Docker with our account. Then, we can simply push our newly created image to Docker.

The command: ‘**docker push blxucreep/td_3**’.

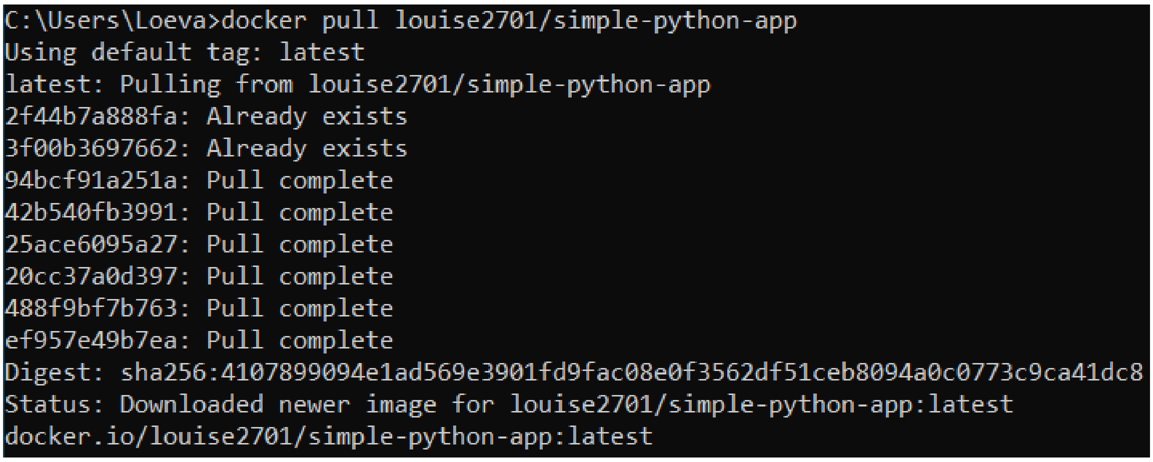


```
C:\Users\Loeva\OneDrive\Bureau\ESILV\A4 cycle ingé DIA\Semestre 8\~ programmation\Containerization Technologies\TD 3>
docker push blxucreep/td_3
Using default tag: latest
The push refers to repository [docker.io/blxucreep/td_3]
657caa010fee: Pushed
d1ff07e25e08: Pushed
24cef30db7c5: Pushed
a90d13bb99a7: Mounted from library/python
681a1fc3389e: Mounted from library/python
47cdb7a27fca: Mounted from library/python
e96fe707bd25: Mounted from library/python
571ade696b26: Mounted from library/python
latest: digest: sha256:dcbaa5d5dd6f30b2354152cc7720cf6f2f8c60791916444f402b6c7f8c6b040a size: 1995
```

Test it on another computer

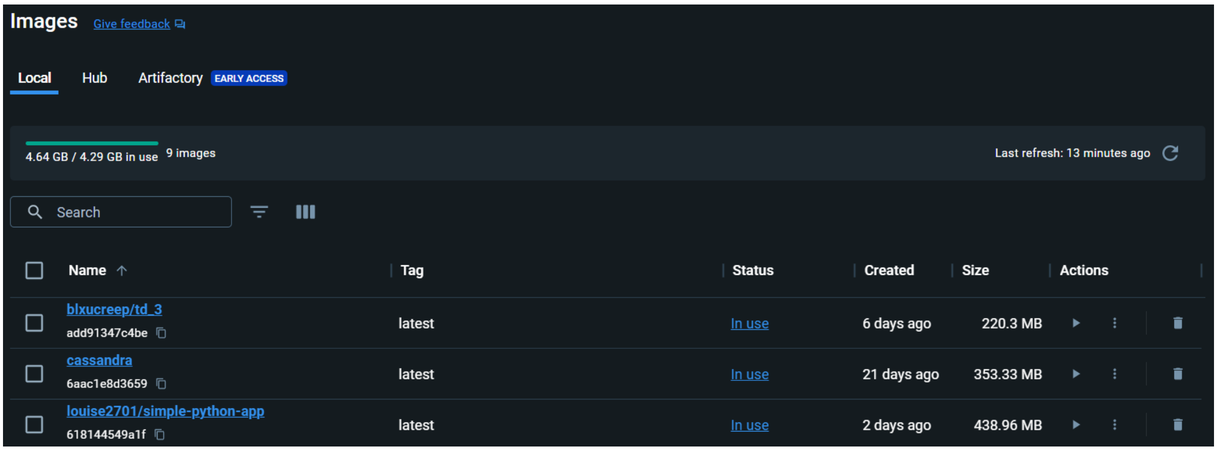
On the other computer, we try to pull the image.

The command: ‘**docker pull louise2701/simple-python-app**’.



```
C:\Users\Loeva>docker pull louise2701/simple-python-app
Using default tag: latest
latest: Pulling from louise2701/simple-python-app
2f44b7a888fa: Already exists
3f00b3697662: Already exists
94bcf91a251a: Pull complete
42b540fb3991: Pull complete
25ace6095a27: Pull complete
20cc37a0d397: Pull complete
488f9bf7b763: Pull complete
ef957e49b7ea: Pull complete
Digest: sha256:4107899094e1ad569e3901fd9fac08e0f3562df51ceb8094a0c0773c9ca41dc8
Status: Downloaded newer image for louise2701/simple-python-app:latest
docker.io/louise2701/simple-python-app:latest
```

Let’s check our image on Docker desktop:



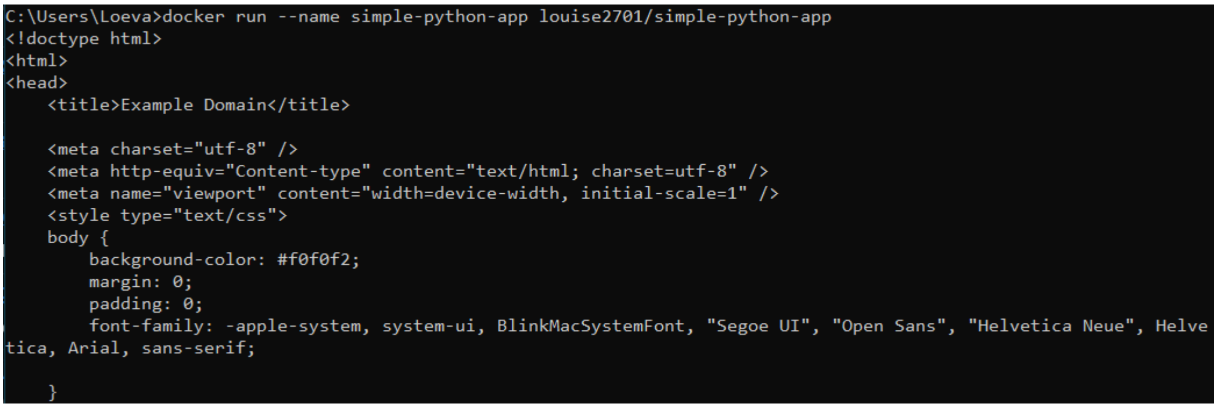
4.64 GB / 4.29 GB in use 9 images

Last refresh: 13 minutes ago

□	Name ↑	Tag	Status	Created	Size	Actions
□	blxucreep/td_3 add91347c4be	latest	In use	6 days ago	220.3 MB	▶ ⋮ 🗑
□	cassandra 6aac1e8d3659	latest	In use	21 days ago	353.33 MB	▶ ⋮ 🗑
□	louise2701/simple-python-app 618144549a1f	latest	In use	2 days ago	438.96 MB	▶ ⋮ 🗑

When everything is ok, it should run without any issue.

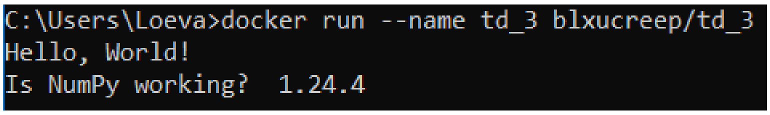
The command: ‘**docker run --name simple-python-app louise2701/simple-python-app**’.



```
C:\Users\Loeva>docker run --name simple-python-app louise2701/simple-python-app
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helve
tica, Arial, sans-serif;

    }
```

Here, I run the image on my computer to be able to do bonus part.

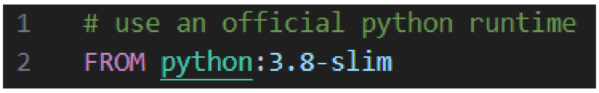
The command: ‘**docker run --name td_3 blxucreep/td_3**’.



```
C:\Users\Loeva>docker run --name td_3 blxucreep/td_3
Hello, World!
Is NumPy working?  1.24.4
```

Bonus smallest possible image size

This line is the one responsible for the smallest possible image size (we had ‘-slim’ at the end):



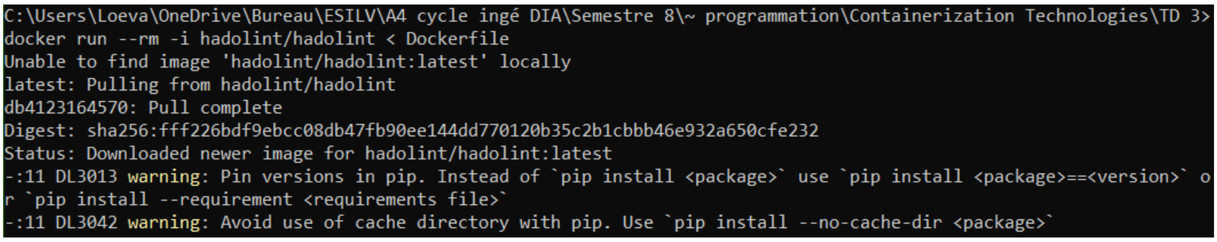
```
# use an official python runtime
FROM python:3.8-slim
```

Bonus run a linter

With Docker, we can install ‘hadolint’ to run a linter on a Dockerfile. First, it downloads the image, then it outputs the warnings/errors in our Dockerfile.

‘--rm’ is used to remove the container when we successfully verified our file, ‘-i’ is for interactive and ‘< Dockerfile’ permits ‘hadolint’ to read the content of the file named ‘Dockerfile’ we want to check.

The command: ‘**docker run --rm -i hadolint/hadolint < Dockerfile**’.



```
C:\Users\Loeva\OneDrive\Bureau\ESILV\A4 cycle ingé DIA\Semestre 8\~ programmation\Containerization Technologies\TD 3>
docker run --rm -i hadolint/hadolint < Dockerfile
Unable to find image 'hadolint/hadolint:latest' locally
latest: Pulling from hadolint/hadolint
db4123164570: Pull complete
Digest: sha256:fff226bdf9ebcc08db47fb90ee144dd770120b35c2b1cbbb46e932a650cfe232
Status: Downloaded newer image for hadolint/hadolint:latest
-:11 DL3013 warning: Pin versions in pip. Instead of `pip install <package>` use `pip install <package>==<version>` o
r `pip install --requirement <requirements file>`
-:11 DL3042 warning: Avoid use of cache directory with pip. Use `pip install --no-cache-dir <package>`
```

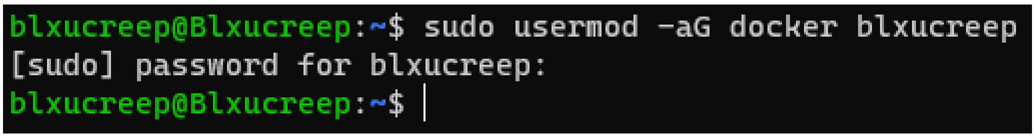
Bonus difference between ADD and COPY

‘COPY’ is used to, logically, copy our files in the current directory into the container. But ‘ADD’ can, in addition to the exact same things than ‘COPY’, download files from other sources, like an URL for example.

Bonus run without sudo rights

In order to do that, we need to add the user we wish him to run the container into the Docker group, permitting running the container without sudo rights.

The command: '**sudo usermod -aG docker blxucreep**'.



```
blxucreep@Blxucreep:~$ sudo usermod -aG docker blxucreep
[sudo] password for blxucreep:
blxucreep@Blxucreep:~$ |
```

(I had some issues with sudo on my laptop but not on my computer at home.)

Bonus run a secure scan

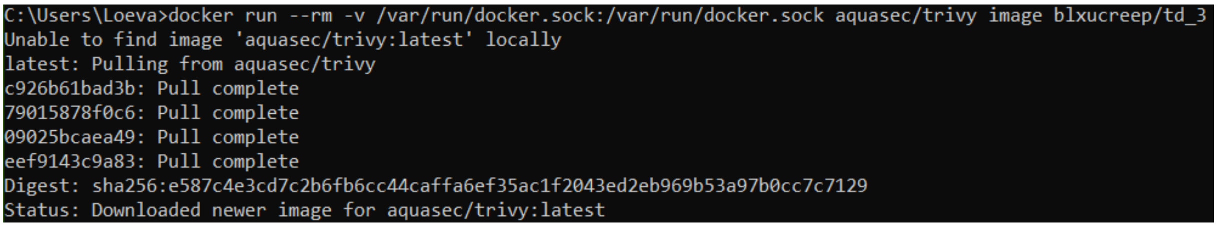
We can use different tools to run a secure scan on the container, here I took 'trivy'. First, it downoalds the image, then updates the database, and finally we have the output of the scan.

We don't forget to remove the container when the check is done ('--rm').

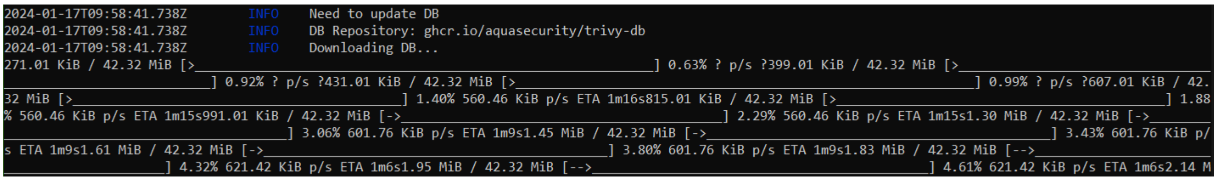
'-v /var/run/docker.sock:/var/run/docker.sock' mounts the Docker socket from the host into the container, allowing 'trivy' to communicate with the Docker daemon.

Finally, we provide the name of the image we want to check ('blxucreep/td_3').

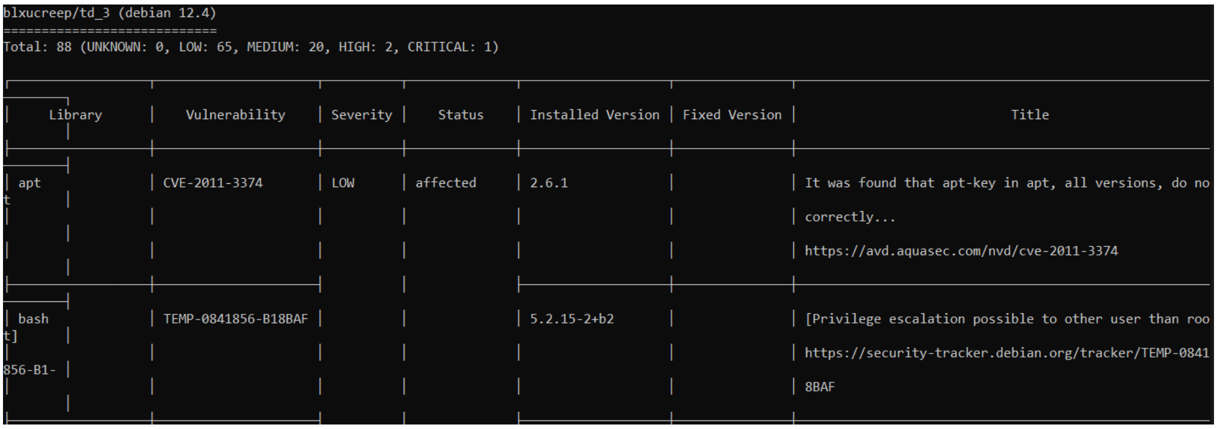
The command: '**docker run --rm -v /var/run/docker.sock:/var/run/docker.sock aquasec/trivy image blxucreep/td_3**'.



```
C:\Users\Loeva>docker run --rm -v /var/run/docker.sock:/var/run/docker.sock aquasec/trivy image blxucreep/td_3
Unable to find image 'aquasec/trivy:latest' locally
latest: Pulling from aquasec/trivy
c926b61bad3b: Pull complete
79015878f0c6: Pull complete
09025bcaea49: Pull complete
eef9143c9a83: Pull complete
Digest: sha256:e587c4e3cd7c2b6fb6cc44caffa6ef35ac1f2043ed2eb969b53a97b0cc7c7129
Status: Downloaded newer image for aquasec/trivy:latest
```



```
2024-01-17T09:58:41.738Z        INFO    Need to update DB
2024-01-17T09:58:41.738Z        INFO    DB Repository: ghcr.io/aquasecurity/trivy-db
2024-01-17T09:58:41.738Z        INFO    Downloading DB...
271.01 KiB / 42.32 MiB [>______________________________________________________________] 0.63% ? p/s ?399.01 KiB / 42.32 MiB [>____________________
___________________________] 0.92% ? p/s ?431.01 KiB / 42.32 MiB [>______________________________________________________] 0.99% ? p/s ?607.01 KiB / 42.
32 MiB [>______________________________________________________] 1.40% 560.46 KiB p/s ETA 1m16s815.01 KiB / 42.32 MiB [>______________________________________] 1.88
% 560.46 KiB p/s ETA 1m15s991.01 KiB / 42.32 MiB [->___________________________________________________] 2.29% 560.46 KiB p/s ETA 1m15s1.30 MiB / 42.32 MiB [->_______
__________________________________________] 3.06% 601.76 KiB p/s ETA 1m9s1.45 MiB / 42.32 MiB [->_________________________________________] 3.43% 601.76 KiB p/
s ETA 1m9s1.61 MiB / 42.32 MiB [->_______________________________________________] 3.80% 601.76 KiB p/s ETA 1m9s1.83 MiB / 42.32 MiB [-->_______________
__________________________] 4.32% 621.42 KiB p/s ETA 1m6s1.95 MiB / 42.32 MiB [-->_____________________________________________] 4.61% 621.42 KiB p/s ETA 1m6s2.14 M
```



```
blxucreep/td_3 (debian 12.4)
============================
Total: 88 (UNKNOWN: 0, LOW: 65, MEDIUM: 20, HIGH: 2, CRITICAL: 1)

┌                    ┬                        ┬          ┬               ┬                     ┬                 ┬
─────────┐
│     Library       │     Vulnerability      │ Severity │    Status     │ Installed Version │ Fixed Version │                    Title
         │
├                    ┼                        ┼          ┼               ┼                     ┼                 ┼
─────────┤
│ apt               │ CVE-2011-3374          │ LOW      │ affected      │ 2.6.1               │                 │ It was found that apt-key in apt, all versions, do no
t        │
│                   │                        │          │               │                     │                 │ correctly...
         │
│                   │                        │          │               │                     │                 │ https://avd.aquasec.com/nvd/cve-2011-3374
         │
├                    ┼                        ┤          │               ├                     ┼                 ┼
─────────┤
│ bash              │ TEMP-0841856-B18BAF │          │               │ 5.2.15-2+b2         │                 │ [Privilege escalation possible to other user than roo
t]       │
│                   │                        │          │               │                     │                 │ https://security-tracker.debian.org/tracker/TEMP-0841
856-B1-  │
│                   │                        │          │               │                     │                 │ 8BAF
         │
├                    ┼                        ┼          ┼               ┼                     ┼                 ┼
```

Bonus public GitHub repository

Every practical works are pushed on GitHub and I am going to make commits regularly.

Here is the link: https://github.com/Blxucreep/containerization-technologies.