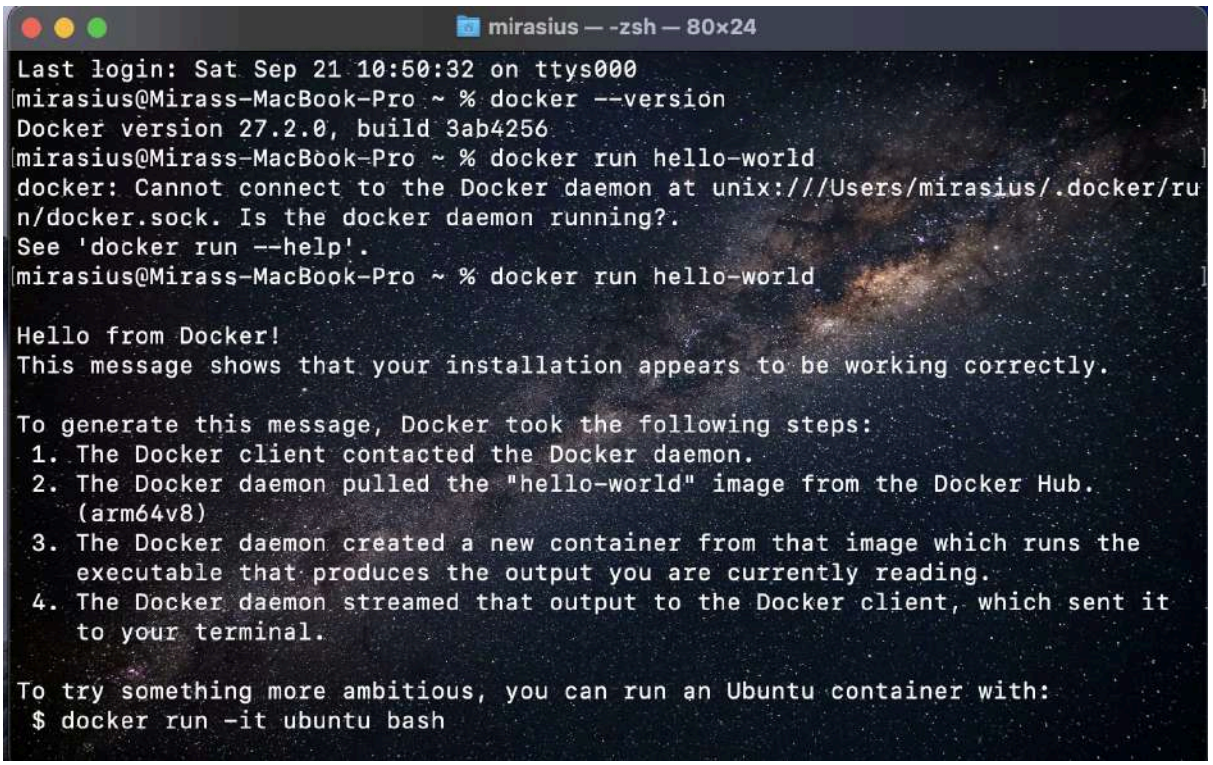
Assignment 1, Web Application Development

Done by: Assubay Miras

Exercise 1: Installing Docker

1. Objective: Install Docker on your local machine
2. Steps:



3. Questions:
   - **What are the key components of Docker (e.g., Docker Engine, Docker CLI)?**
   Docker container, image, daemon, docker CLI, docker compose
   - **How does Docker compare to traditional virtual machines?**
   It's more lightweight and doesn't require a lot configurations and repetitive work, you can write dockerfile once and it's configurations will be same every time when we do docker run
   - **What was the output of the docker run hello-world command, and what does it signify?**
   It describes steps of how this container ran, and what was done to show and generate the message into the terminal

Exercise 2: Basic Docker Commands

1. Objective: Familiarize yourself with basic Docker commands.
2. Steps:

```
mirasius — -zsh — 80×24
mirasius@Mirass-MacBook-Pro ~ % docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
Digest: sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f28dfd5a9710e3
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout q
uickview nginx
mirasius@Mirass-MacBook-Pro ~ % docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
nginx         latest    195245f0c792   5 weeks ago     193MB
hello-world   latest    ee301c921b8a   16 months ago   9.14kB
mirasius@Mirass-MacBook-Pro ~ % docker run -d nginx
747217b74861d845f521c4024b624e2d5816f21bc325d5c3224544c8f1211474
mirasius@Mirass-MacBook-Pro ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS
           PORTS     NAMES
747217b74861   nginx     "/docker-entrypoint.…"   About a minute ago   Up About
a minute   80/tcp    cool_carson
mirasius@Mirass-MacBook-Pro ~ % docker stop 747217b74861
747217b74861
mirasius@Mirass-MacBook-Pro ~ %
```

3. Questions:

○ **What is the difference between docker pull and docker run?**

Docker pull downloads the image, while docker run starts container and runs it

○ **How do you find the details of a running container, such as its ID and status?**

with command **docker ps**

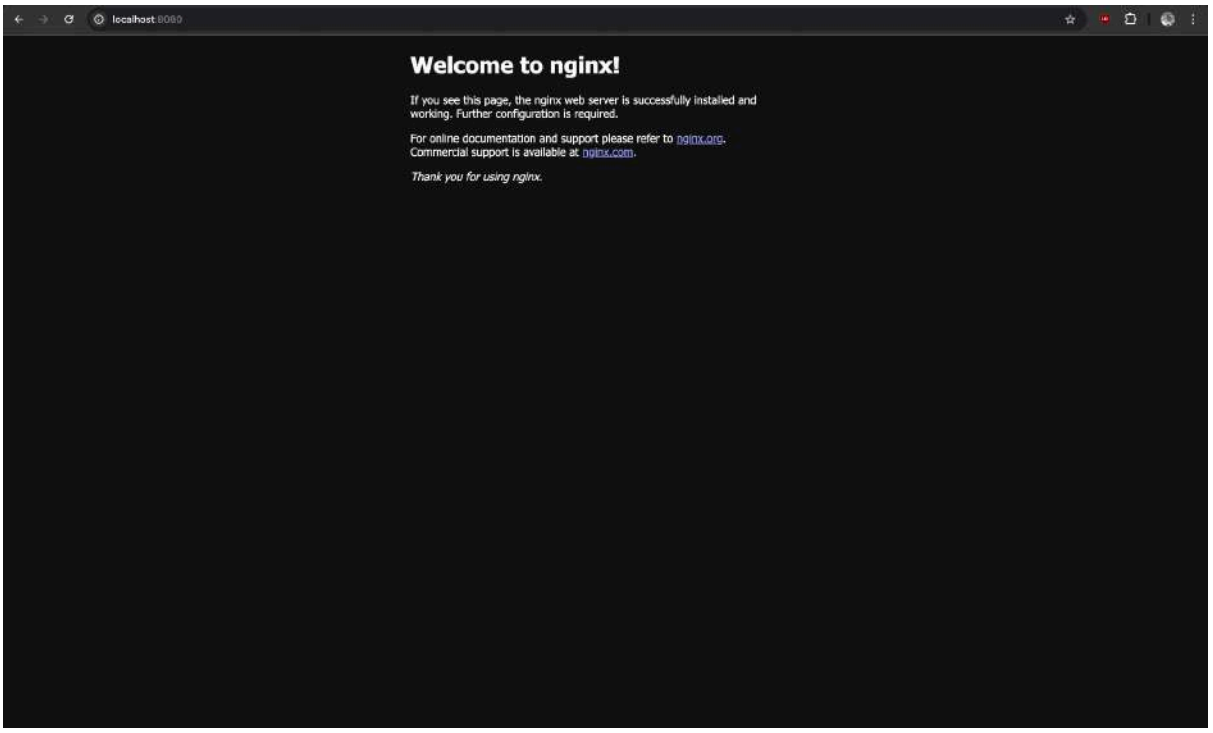
○ **What happens to a container after it is stopped? Can it be restarted?**

Image itself will be held in memory of computer or server, and it can be restarted anytime with knowing name of the image

Exercise 3: Working with Docker Containers

1. Objective: Learn how to manage Docker containers
2. Steps:

```
mirasius@Mirass-MacBook-Pro ~ % docker run -d -p 8080:80 nginx
044123d0a9e942e19ee8e93766ed24c53846975d6bcdf5a9d4716973c53b5daa
mirasius@Mirass-MacBook-Pro ~ % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS
   PORTS                    NAMES
044123d0a9e9   nginx     "/docker-entrypoint.…"   34 seconds ago   Up 34 seconds
   0.0.0.0:8080->80/tcp   nice_beaver
mirasius@Mirass-MacBook-Pro ~ % docker exec -it 044123d0a9e9 /bin/bash
root@044123d0a9e9:/# exit
exit

What's next:
    Try Docker Debug for seamless, persistent debugging tools in any container o
r image → docker debug 044123d0a9e9
    Learn more at https://docs.docker.com/go/debug-cli/
mirasius@Mirass-MacBook-Pro ~ % docker stop 044123d0a9e9
044123d0a9e9
mirasius@Mirass-MacBook-Pro ~ % docker rm 044123d0a9e9
044123d0a9e9
mirasius@Mirass-MacBook-Pro ~ %
```



3. Questions:

○ **How does port mapping work in Docker, and why is it important?**

It port forwards from port that was set inside docker image by default to the port that we can choose, for example port that was set inside docker image is already taken, and with docker we can change it

○ **What is the purpose of the docker exec command?**

It give us opportunity to go inside of the container and execute different command inside container

○ **How do you ensure that a stopped container does not consume system resources?**

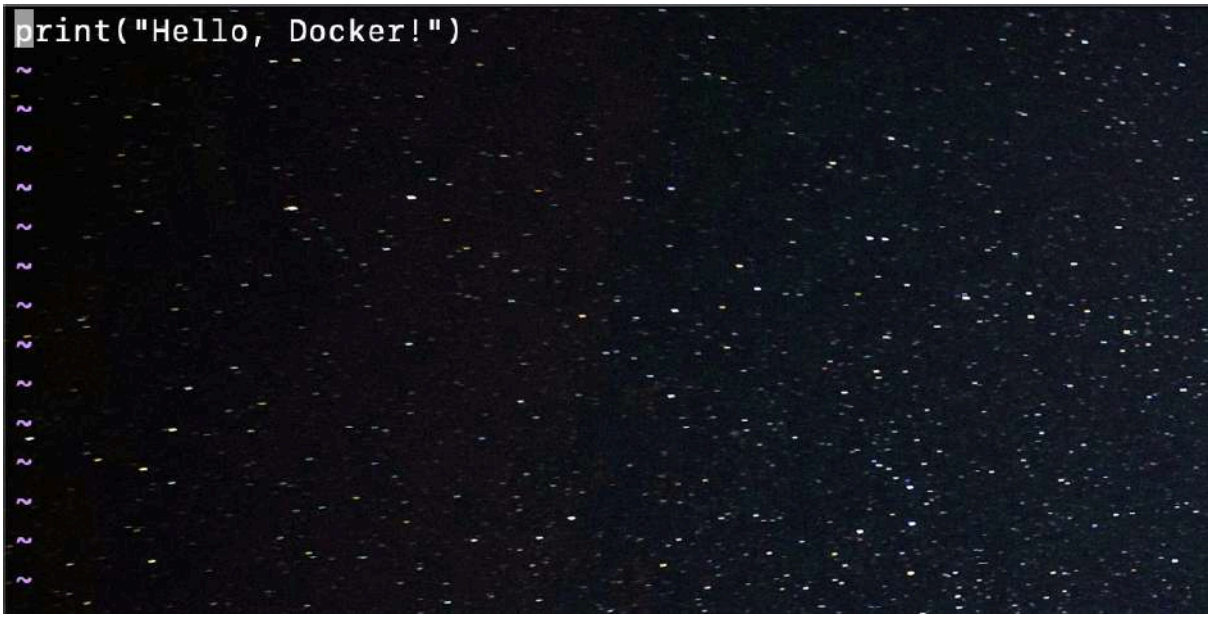
We can see it using command **docker ps** that no containers is running

Dockerfile

Exercise 1: Creating a simple dockerfile

1. Objective: Write a Dockerfile to containerize a basic application.

2. Steps:



```
FROM python:latest

WORKDIR /app

COPY app.py .

ENTRYPOINT ["python", "app.py"]
```

```
Start a build
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker build -t hello-docker .
[+] Building 40.7s (9/9) FINISHED                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                          0.0s
 => => transferring dockerfile: 119B                                          0.0s
 => [internal] load metadata for docker.io/library/python:latest              2.8s
 => [auth] library/python:pull token for registry-1.docker.io                 0.0s
 => [internal] load .dockerignore                                             0.0s
 => => transferring context: 2B                                               0.0s
 => [1/3] FROM docker.io/library/python:latest@sha256:7859853e7607927aa1      3.5s
 => => resolve docker.io/library/python:latest@sha256:7859853e7607927aa10     0.0s
 => => sha256:7859853e7607927aa1d1b1a5a2f9e580ac90c2b66fe 9.72kB / 9.72kB     0.0s
 => => sha256:9c1a7e7e41337f687fa7e5196aada121fd4897cb32f 2.33kB / 2.33kB     0.0s
 => => sha256:dbb5b79c2668bb64927f07f13fcb3d0c506f2ab8c30 5.86kB / 5.86kB     0.0s
 => => sha256:364d19f59f69474a80c53fc78da91f85553e16e8 23.59MB / 23.59MB     14.6s
 => => sha256:56c9b9253ff98351db158cb67898486656b8d54f41 49.59MB / 49.59MB    6.3s
 => => sha256:843b1d8321825bc8302752ae093026f13bd15e6e 64.00MB / 64.00MB     18.7s
 => => extracting sha256:56c9b9253ff98351db158cb67898486656b8d54f411c00373    1.9s
 => => sha256:a348c2a8d94613f34ce7d0ac4fd4e51800d8f4 202.65MB / 202.65MB     32.5s
 => => extracting sha256:364d19f59f69474a80c53fc78da91f85553e16e8be6e2886     0.6s
 => => sha256:353de1168lb2d36db971d2402e6deb187a2651bc3f 6.24MB / 6.24MB     17.4s
 => => sha256:799b63efcab5e4f11215dd6a8c4f59d117ae572e 23.65MB / 23.65MB     22.3s
 => => sha256:238a68c3d761a41b503e61aed92cb67f43c456139d427d 251B / 251B     19.2s
 => => extracting sha256:843b1d8321825bc8302752ae093026f13bd15e6eaf3cfed3    1.8s
 => => extracting sha256:a348c2a8d94613f34ce7d0ac4fd4e51800d8f4a8a0bc93e     4.1s
 => => extracting sha256:353de1168lb2d36db971d2402e6deb187a2651bc3f1c4554b    0.2s
 => => extracting sha256:799b63efcab5e4f11215dd6a8c4f59d117ae572e0f8b0b0b    0.5s
 => => extracting sha256:238a68c3d761a41b503e61aed92cb67f43c456139d427d47    0.0s
 => [internal] load build context                                             0.0s
 => => transferring context: 5.7B                                             0.0s
 => [2/3] WORKDIR /app                                                        0.3s
 => [3/3] COPY app.py .                                                       0.0s
 => exporting to image                                                        0.0s
 => => exporting layers                                                       0.0s
 => => writing image sha256:2b146ffb1fb94faff4dcb78c8c3eb06fa8968620afd63     0.0s
 => => naming to docker.io/library/hello-docker                               0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/wapwzhinfuypw1pcxs49z9k76

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker run hello-docker
Hello, Docker!
mirasius@Mirass-MacBook-Pro WebAppAssignment1 %
```

3. Questions:

○ **What is the purpose of the FROM instruction in a Dockerfile?**

It's the base image from which our new image will be build from

○ **How does the COPY instruction work in Dockerfile?**

It copies files from our computer to the container

○ **What is the difference between CMD and ENTRYPOINT in Dockerfile?**

ENTRYPOINT is immutable, and we can only add something at the end of command, while CMD is customizable and some arguments can be changed

Exercise 2

```
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker run hello-docker-alpine
Hello, Docker!
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
hello-docker-alpine latest    5068ffe3e573   14 seconds ago   53.5MB
hello-docker2       latest    18231dd4a57a   3 minutes ago    1.02GB
```

```
FROM python:3.9-alpine

WORKDIR /app

COPY . .

ENTRYPOINT ["python", "app.py"]

~
~
~
~
```

Exercise 3

MultiStage:

```
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker build -t hello-go-multistage .
[+] Building 17.1s (15/15) FINISHED                                docker:desktop-linux
 => [internal] load build definition from Dockerfile                                    0.0s
 => => transferring dockerfile: 240B                                                    0.0s
 => [internal] load metadata for docker.io/library/alpine:latest                        2.8s
 => [internal] load metadata for docker.io/library/golang:1.18-alpine                   3.0s
 => [auth] library/alpine:pull token for registry-1.docker.io                           0.0s
 => [auth] library/golang:pull token for registry-1.docker.io                           0.0s
 => [internal] load .dockerignore                                                       0.0s
 => => transferring context: 557B                                                       0.0s
 => [builder 1/4] FROM docker.io/library/golang:1.18-alpine@sha256:77f25 13.7s
 => => resolve docker.io/library/golang:1.18-alpine@sha256:77f25981bd57e6 0.0s
 => => sha256:77f25981bd57e60a51016bf3be89c901aec90453fd0 1.66kB / 1.65kB 0.0s
 => => sha256:670182396fe94e1b50ea107b66d4b2d158b2cd7c76f 1.16kB / 1.16kB 0.0s
 => => sha256:6867eb9d8bd48c0f43214b09485a3dbf916437bd6ae 5.03kB / 5.03kB 0.0s
 => => sha256:a9eaa45ef418e883481a13c7d84fa9904f2ec56789c 3.26MB / 3.26MB 1.1s
 => => sha256:da3aa66ae7ca151fbd33126b15fa4d01bd5ceb3 286.26kB / 286.26kB 0.7s
 => => sha256:72d1fefe56e2997391a3dee0aa70cb676557be 110.45MB / 110.45MB 10.5s
 => => sha256:a13fd264aa8ef04a29b827fe2454ac2bb9c52f70bf0b0ea 156B / 156B 1.1s
 => => extracting sha256:a9eaa45ef418e883481a13c7d84fa9904f2ec56789c02a87 0.1s
 => => extracting sha256:da3aa66ae7ca151fbd33126b15fa4d01bd5ceb35d53de36a 0.0s
 => => extracting sha256:72d1fefe56e2997391a3dee0aa70cb676557be6201cca145 3.0s
 => => extracting sha256:a13fd264aa8ef04a29b827fe2454ac2bb9c52f70bf0b0ea0 0.0s
 => [internal] load build context:                                                      0.0s
 => => transferring context: 140B                                                       0.0s
 => [stage-1 1/3] FROM docker.io/library/alpine:latest@sha256:beefdbd8a1d 0.0s
 => => resolve docker.io/library/alpine:latest@sha256:beefdbd8a1da6d2915 0.0s
 => => sha256:beefdbd8a1da6d2915566fde36db9db0b524eb737fc 1.85kB / 1.85kB 0.0s
 => => sha256:9cee2b382fe2412cd77d6d437d15a33de8de37381e2177 528B / 528B 0.0s
 => => sha256:c157a86ed455142fd79bff5dce951fd5fb8d0c6e4b 1.47kB / 1.47kB 0.0s
 => [stage-1 2/3] WORKDIR /app                                                          0.0s
 => [builder 2/4] WORKDIR /app                                                          0.1s
 => [builder 3/4] COPY . .                                                              0.0s
 => [builder 4/4] RUN go mod init helloapp && go build -o helloapp                      0.2s
 => [stage-1 3/3] COPY --from=builder /app/helloapp .                                   0.0s
 => exporting to image                                                                  0.0s
 => => exporting layers                                                                 0.0s
 => => writing image sha256:7dc2dadc85c86cfd8ffbfa2aed488850a1a2514a8c4ed 0.0s
 => => naming to docker.io/library/hello-go-multistage                                  0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/f4owljhjfqqo49dexjih1vuwf

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker run hello-go-multistage
Hello, World!
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % vim Dockerfile
```

```
FROM golang:1.18-alpine AS builder

WORKDIR /app

COPY . .

RUN go mod init helloapp && go build -o helloapp

FROM alpine:latest

WORKDIR /app

COPY --from=builder /app/helloapp .

CMD ["./helloapp"]

~
~
~
```

Singlestage build

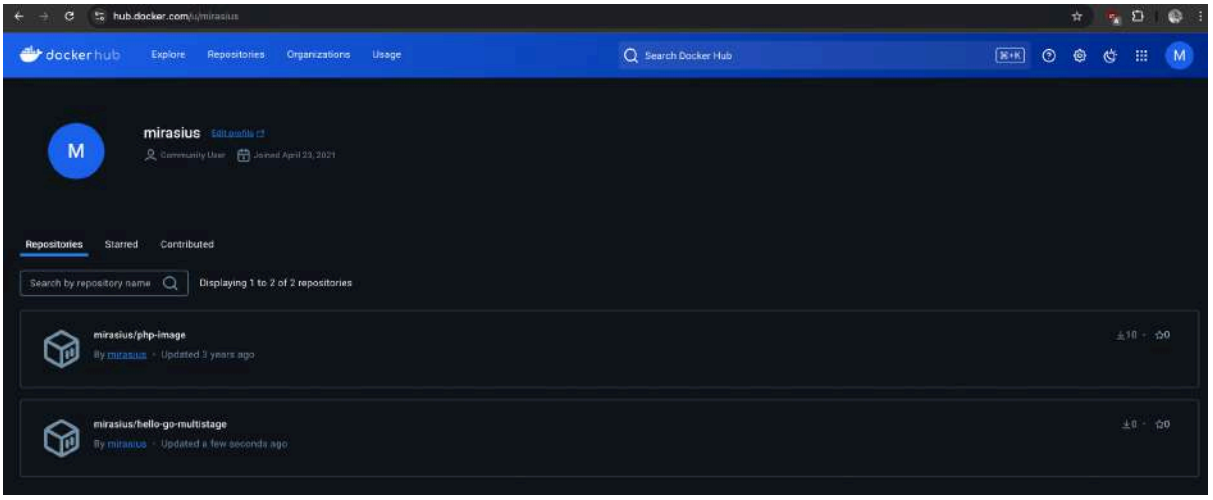
```
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker build -t hello-go-singlestage .
[+] Building 0.8s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 156B
 => [internal] load metadata for docker.io/library/golang:1.18-alpine
 => [internal] load .dockerignore
 => => transferring context: 567B
 => [1/4] FROM docker.io/library/golang:1.18-alpine@sha256:77f25981bd57e60a510165f9ba89c901a
 => [internal] load build context
 => => transferring context: 53B
 => CACHED [2/4] WORKDIR /app
 => CACHED [3/4] COPY . .
 => CACHED [4/4] RUN go mod init helloapp && go build -o helloapp
 => exporting to image
 => => exporting layers
 => => writing image sha256:c8b9fa7e9bb8447a913a0a7884ca81dd01ab401a647a3ab04ff4625a6233a92
 => => naming to docker.io/library/hello-go-singlestage

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/v10bbl6wgja

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker run hello-go-singlestage
Hello, World!
```

```
FROM golang:1.18-alpine

WORKDIR /app

COPY . .

RUN go mod init helloapp && go build -o helloapp

CMD ["./helloapp"]

~
~
```

Image sizes:

```
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker images
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
hello-go-multistage     latest    76c2dadce5c8   3 minutes ago    10.6MB
hello-go-singlestage    latest    c8b9fa7e9bb8   3 minutes ago    330MB
```

Exercise 4:




```
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker tag hello-go-multistage m
irasius/hello-go-multistage
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker login
Authenticating with existing credentials...
Login Succeeded
mirasius@Mirass-MacBook-Pro WebAppAssignment1 % docker push mirasius/hello-go-mu
ltistage
Using default tag: latest
The push refers to repository [docker.io/mirasius/hello-go-multistage]
bb856d5cb452: Pushed
be3c60720aac: Pushed
16113d51b718: Mounted from library/alpine
latest: digest: sha256:1439d0ede89aff5dbc0f4ce567e46447281dff8cf460ac248564b8286
453a7b1 size: 944
mirasius@Mirass-MacBook-Pro WebAppAssignment1 %
```