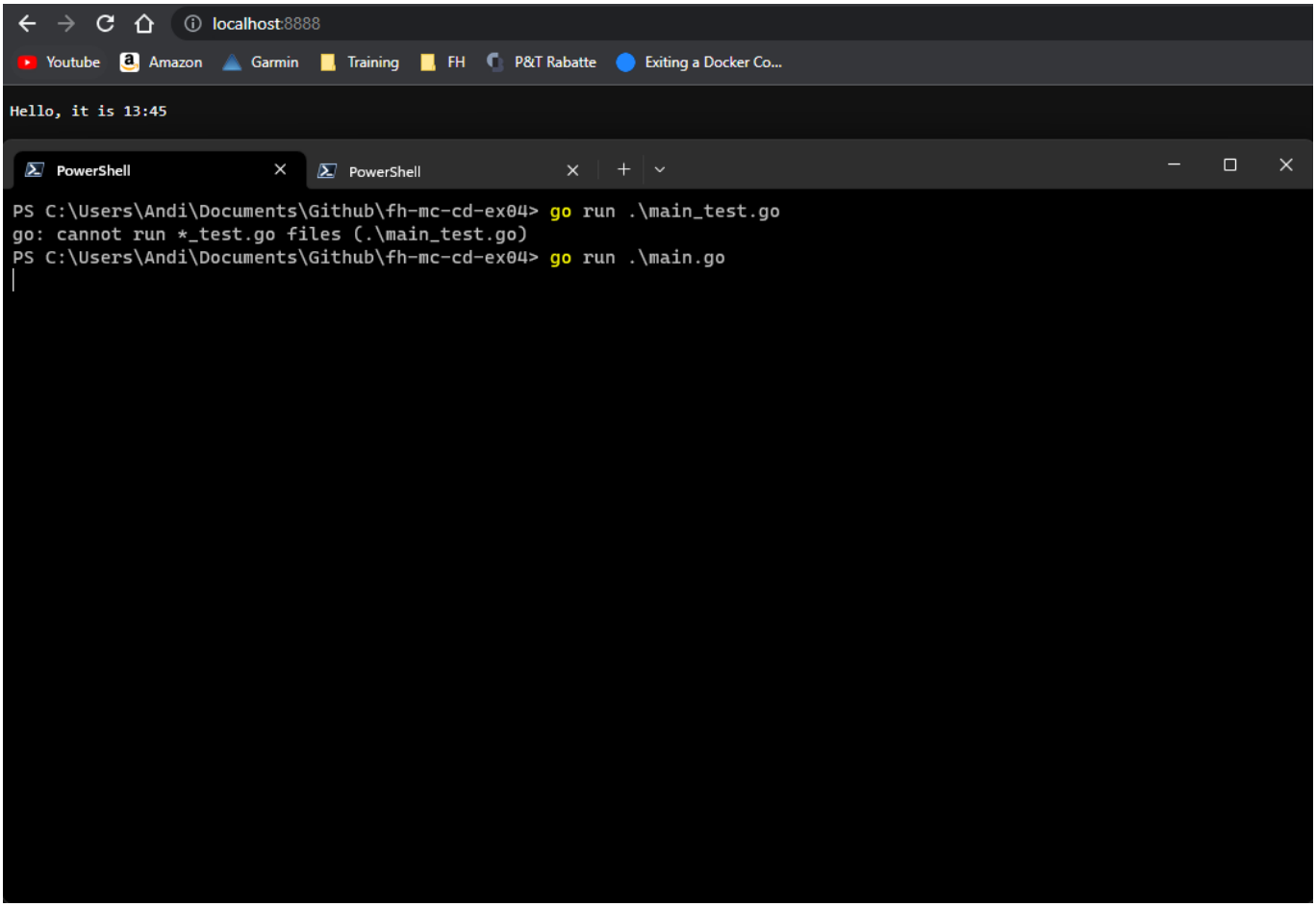
Exercise 4

Part 1

1. Check if the go program runs locally

First create a go.mod file (go init [NAME]).

```
go run main.go
```



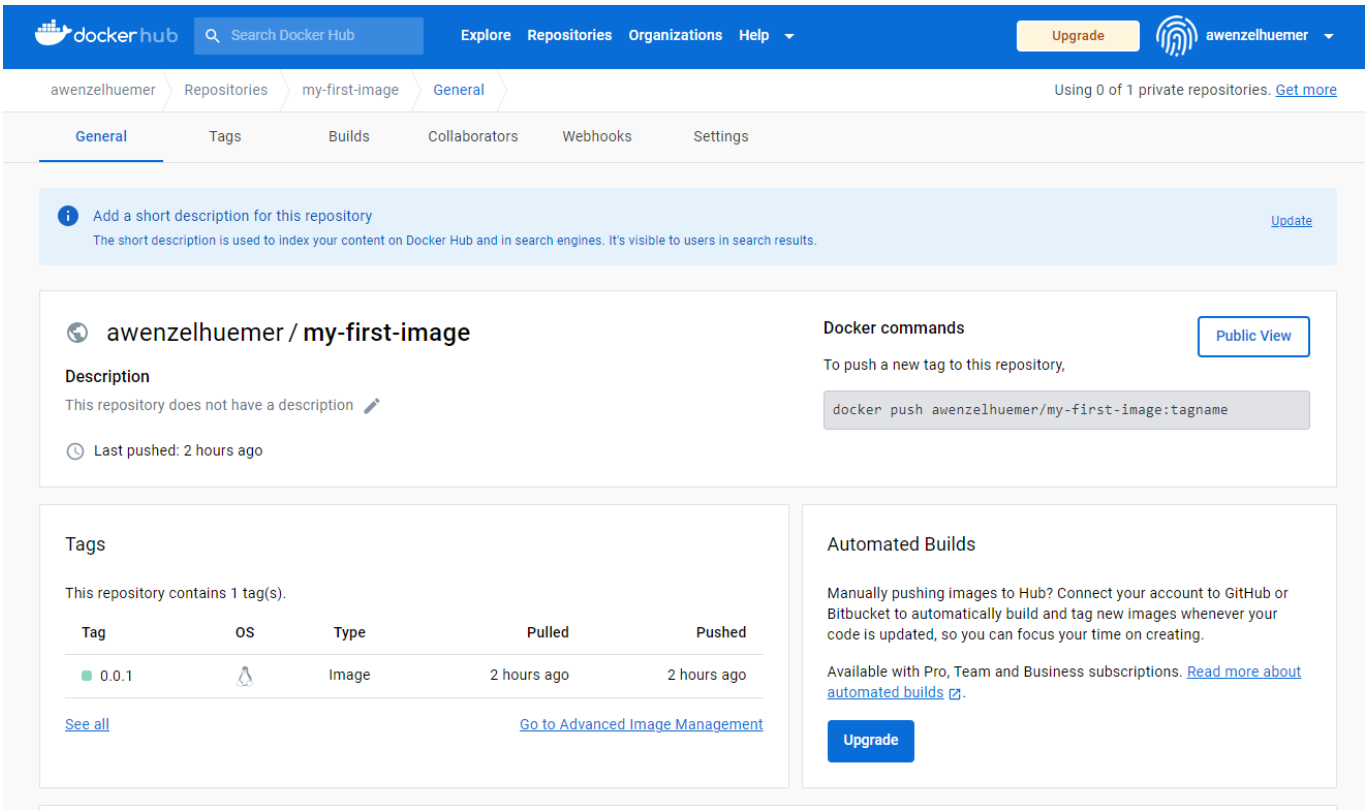
2. Modify the dockerfile in the repo

```
FROM golang:1.20-alpine

LABEL maintainer="andreas.wenzelhuemer@gmail.com"

WORKDIR /src

COPY . .

RUN ls -al
RUN go build -o myapp
```

```
RUN mv myapp /usr/

EXPOSE 8888

CMD ["/usr/myapp"]
```

3. Build a docker image based on your dockerfile

```
docker image build -f -t awenzelhuemer/my-first-image:0.0.1 ./
```



4. List all images that are stored in your local registry

```
docker images
```

```
PS C:\Users\Andi> docker images
REPOSITORY                                    TAG          IMAGE ID       CREATED          SIZE
awenzelhuemer/go-service                      latest       e454fdd25f76   5 minutes ago    339MB
awenzelhuemer/go-service                      <none>       eebfefdb6b44   11 minutes ago   339MB
awenzelhuemer/go-service                      <none>       fca2c6fd9398   25 minutes ago   318MB
<none>                                        <none>       ae73f07dae0f   46 minutes ago   318MB
quay.io/quarkus/ubi-quarkus-mandrel-builder-image   22.3-java17   413fc1ddd944   2 days ago       1.07GB
neo4j                                         latest       bbff370c3917   6 days ago       543MB
fhooe-mongo-dock-php-apache                   latest       75e277b581fa   12 days ago      681MB
postgres                                      latest       2bb008a38e7c   4 weeks ago      379MB
mongo                                         latest       9a5e0d0cf6de   6 weeks ago      646MB
ghcr.io/muchobien/pocketbase                  latest       d008fe52faf6   6 weeks ago      55.5MB
postgres                                      14           176399451347   8 weeks ago      377MB
elk-stack-dock-setup                          latest       67830e84b31c   8 weeks ago      1.29GB
elk-stack-dock-elasticsearch                  latest       9227ce144fcd   3 months ago     1.29GB
elk-stack-dock-kibana                         latest       f46724c4e069   3 months ago     717MB
elk-stack-dock-metricbeat                     latest       35f9658d367f   3 months ago     320MB
elk-stack-dock-logstash                       latest       84efc0f6fcfb   3 months ago     748MB
postgres                                      14.1-alpine  1149d285a5f5   15 months ago    209MB
mongo-express                                 latest       2d2fb2cabc8f   18 months ago    136MB
```

5. Authenticate to the container registry

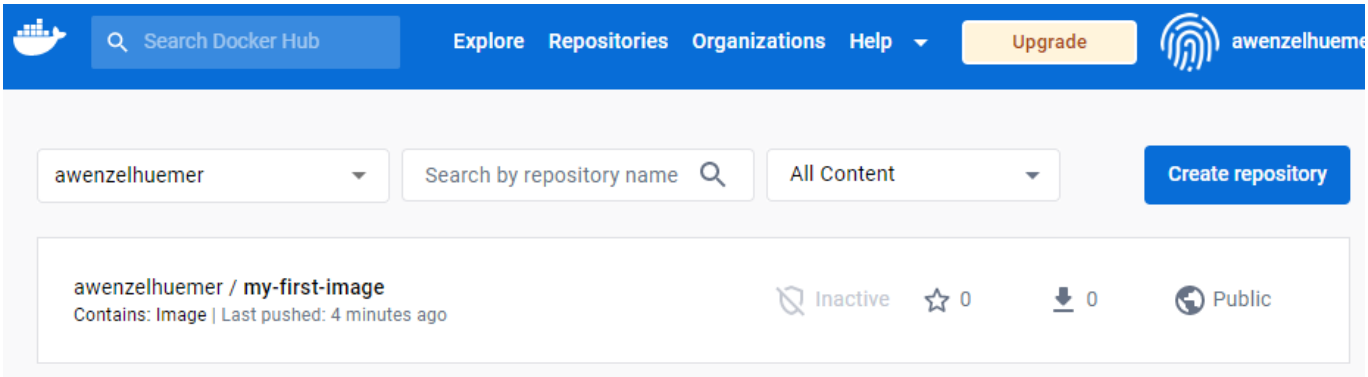
```
PS C:\Users\Andi> docker login
Login with your Docker ID to push and pull images from Docker Hub. If you do
n't have a Docker ID, head over to https://hub.docker.com to create one.
Username: awenzelhuemer
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your a
ccount.
For better security, log in with a limited-privilege personal access token.
Learn more at https://docs.docker.com/go/access-tokens/
```

6. Push the created image to your DockerHub account

```
docker image push awenzelhuemer/my-first-image:0.0.1
```

```
PS C:\Users\Andi\Documents\Github\fh-mc-cd-ex04> docker image push awenzelhuemer/my-first-image:0.0.1
The push refers to repository [docker.io/awenzelhuemer/my-first-image]
09636513e2b4: Pushed
403c1463e48b: Pushed
8130600bac74: Pushed
9d1f8a91962b: Pushed
b321ce0ae8c5: Pushed
0ac507e92643: Mounted from awenzelhuemer/go-service
84c52813c38c: Mounted from awenzelhuemer/go-service
0403d7f628d3: Mounted from awenzelhuemer/go-service
f1417ff83b31: Mounted from awenzelhuemer/go-service
0.0.1: digest: sha256:e47c5e6824fe26ccf0632d55da0c6e894e815f2e6e4379a643b615f7573161c9 size: 2205
```

7. Verify deployed image



Part 2

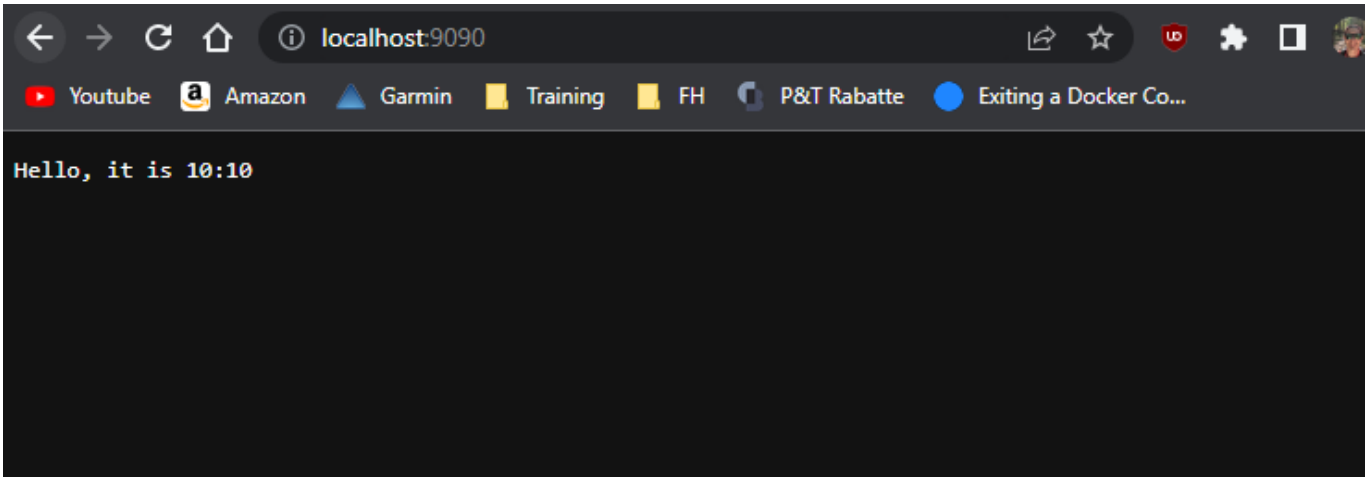
1. Create image from the provided Dockerfile

```
docker image build -t awenzelhuemer/myhello:0.0.1 ./
```

2. Run image

```
docker run --rm -p 9090:8888 awenzelhuemer/myhello:0.0.1
```

3. Check if application is running on localhost:9090



4. See your container running on your local Docker daemon

```
docker ps
```

```
PS C:\Users\Andi> docker ps
CONTAINER ID   IMAGE                      COMMAND        CREATED         STATUS         PORTS
NAMES
6de2db1ee792   awenzelhuemer/myhello:0.0.1   "/usr/myapp"   2 minutes ago   Up 2 minutes   0.0.0.0:9090->8888/tcp
compassionate_euclid
```

5. Stop your container

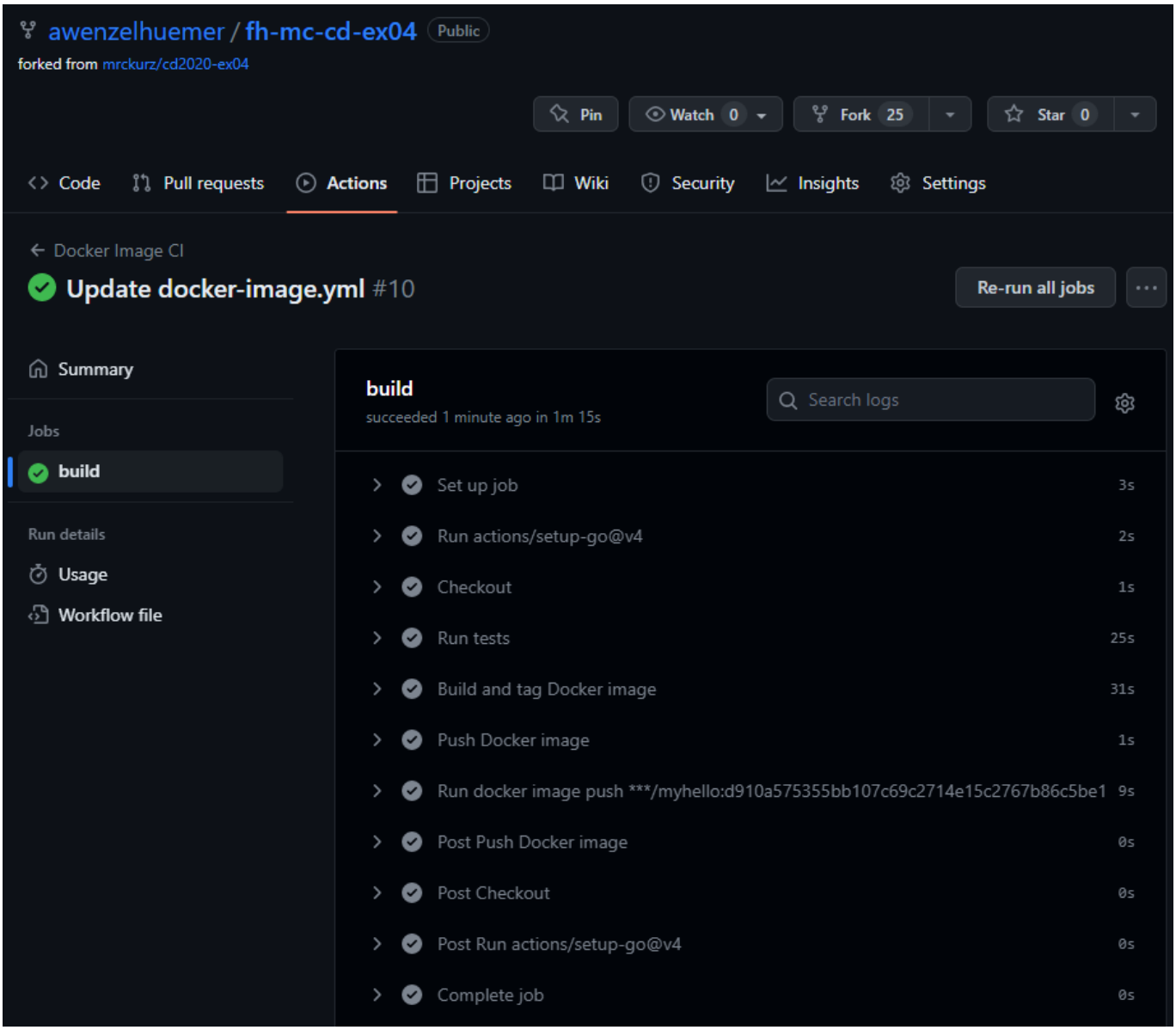
```
docker stop 6de2db1ee792
```

Part 3

Created with github actions in separate steps:

- build: Run go tests
- docker: Build image and deploy to docker hub

```
name: Docker Image CI

on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]

jobs:
  build:
    name: Build and test application
    runs-on: ubuntu-latest
    steps:
    - uses: actions/setup-go@v4
      with:
        go-version: 'stable'
    - name: Checkout
      uses: actions/checkout@v3
    - name: Run tests
      run: go test -v

  docker:
    name: Build and push Docker image
    needs: build
    runs-on: ubuntu-latest
    steps:
    - name: Checkout
      uses: actions/checkout@v3
    - name: Build and tag Docker image
      run: docker image build -t awenzelhuemer/myhello:${{ github.sha }} -t
awenzelhuemer/myhello:latest ./
    - name: Push Docker image
      uses: docker/login-action@v2
      with:
        username: ${{ secrets.DOCKER_USERNAME }}
        password: ${{ secrets.DOCKER_TOKEN  }}
    - run: docker image push -a awenzelhuemer/myhello
```



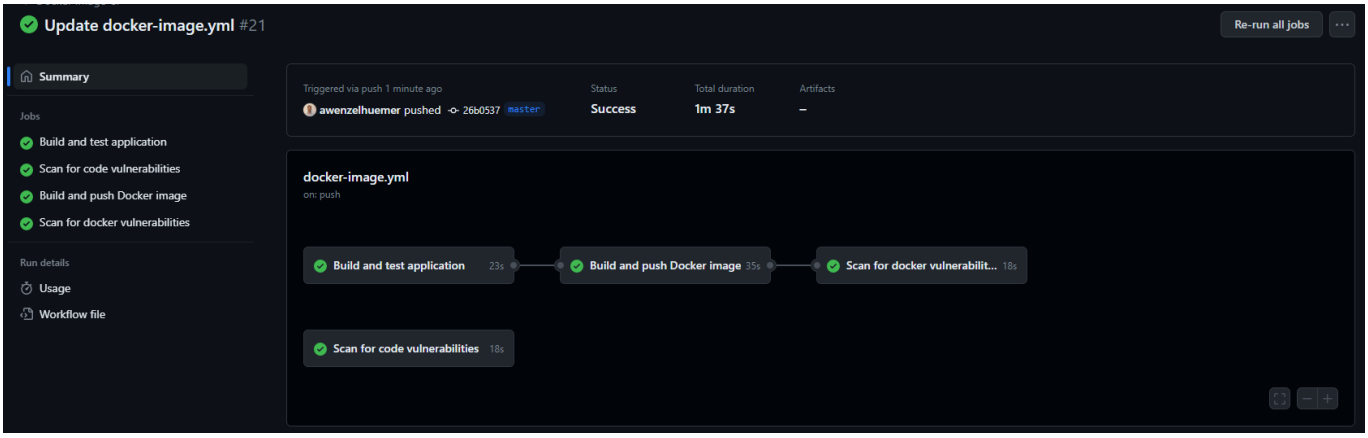

```
docker image pull awenzelhuemer/myhello
```

```
PS C:\Users\Andi\Documents\Github\fh-mc-cd-ex04> docker pull awenzelhuemer/myhello
Using default tag: latest
latest: Pulling from awenzelhuemer/myhello
f56be85fc22e: Already exists
85791d961cd3: Already exists
9a18547b54a1: Already exists
55e4c593b212: Already exists
823b3e814265: Pull complete
921601d28df2: Pull complete
3ad45adb2eda: Pull complete
7275103ff430: Pull complete
Digest: sha256:b679316d7a939f4a1f3f8597e15a6424169da68798c08f5403f33d261b22d0e6
Status: Downloaded newer image for awenzelhuemer/myhello:latest
docker.io/awenzelhuemer/myhello:latest
```

Part 4

Scan docker and code for vulnerabilities.

```
  trivy-docker:
    name: "Scan for docker vulnerabilities"
    needs: docker
    runs-on: ubuntu-latest
    steps:
      - name: Run scanner
        uses: aquasecurity/trivy-action@master
        with:
          image-ref: "awenzelhuemer/myhello:latest"
          format: "table"
          exit-code: "1"
          ignore-unfixed: true
          skip-files: "*.go"
          vuln-type: "os,library"
          severity: "CRITICAL"

  trivy-repo:
    name: "Scan for code vulnerabilities"
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v3
      - name: Run scanner
        uses: aquasecurity/trivy-action@master
        with:
          scan-type: "fs"
          scan-ref: "."
          format: "sarif"
          output: "trivy-results.sarif"
          severity: "CRITICAL,HIGH"
```



Log for docker vulnerabilities: