ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

*Московский институт электроники и математики им. А.Н.Тихонова*

**Домашнее задание по по теме 5. Разработка защищенных программных систем**

По направлению 10.04.01 – «Информационная безопасность»

Проверил:

преп.

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

Новиков В. С. МКБ 241

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2025

# Задание

К заданию приложен архив с репозиторием приложения Pygoat. Скачайте и распакуйте его.

1. В корне репозитория расположен Dockerfile, который содержит инструкции для сборки образа контейнера с этим приложением. С помощью Hadolint (https://github.com/hadolint/hadolint) выполните сканирование Dockerfile и исправьте ошибки в файле.
2. Соберите образ контейнера.
3. С помощью Trivy (https://github.com/aquasecurity/trivy) выполните сканирование образа на уязвимости.
4. Запустите кластер Kubernetes и с помощью kube-bench (https://github.com/aquasecurity/kube-bench) выполните сканирование кластера на соответствие CIS Benchmark.
5. Запустите контейнер Pygoat в Docker.
6. \*Запустите приложение Pygoat в кластере Kubernetes.

# Этап 1. Установка и использование Hadolint

## 1. Установка:

Bash:

*sudo wget https://github.com/hadolint/hadolint/releases/download/v2.14.0/hadolint-x86\_64-linux-standalone -O /usr/local/bin/hadolint*

*sudo chmod +x /usr/local/bin/hadolint*

## 2. Проверка Dockerfile:

Bash:

*hadolint config/Dockerfile*

## 3. Результаты:

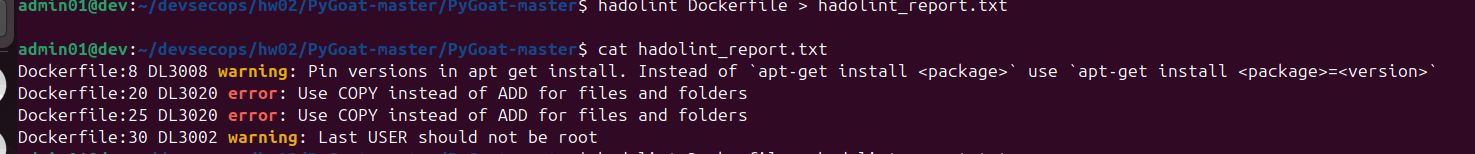


Рисунок. 1. «до исправлений»

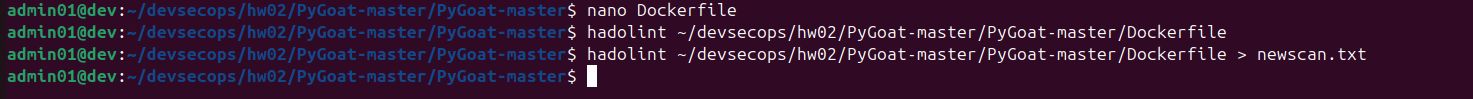


Рисунок. 2. «после исправлений»

Новый Dockerfile:

FROM python:3.9-slim-bookworm

SHELL ["/bin/bash", "-o", "pipefail", "-c"]

# set work directory

WORKDIR /app

# Набор для сборки psycopg2 (gcc+libpq-dev) и диагностики

RUN export DEBIAN\_FRONTEND=noninteractive \

&& apt-get update \

&& apt-get install -y --no-install-recommends \

build-essential=12.\* \

libpq-dev=15.\* \

dnsutils=1:9.18.\* \

&& rm -rf /var/lib/apt/lists/\*

# 13 Set environment variables

ENV PYTHONDONTWRITEBYTECODE 1

ENV PYTHONUNBUFFERED 1

#18 Install dependencies

RUN python -m pip install --no-cache-dir pip==22.0.4

RUN useradd -m -U -s /usr/sbin/nologin app

#24 copy project

COPY --chown=app:app requirements.txt /app/requirements.txt

RUN pip install --no-cache-dir -r /app/requirements.txt

COPY --chown=app:app . /app/

# install pygoat

EXPOSE 8000

USER app

WORKDIR /app/pygoat/

CMD ["gunicorn","--bind","0.0.0.0:8000","--workers","6","pygoat.wsgi"]

# Этап 2. Проверка контейнера с помощью Trivy

## Установка:

Bash:

*wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -*

*echo "deb https://aquasecurity.github.io/trivy-repo/deb $(lsb\_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list*

*sudo apt-get update*

*sudo apt-get install -y trivy*

## Сканирование:

Bash:

*trivy image pygoat:latest*

## Отчёт (ссылка на github)

<https://github.com/vit81g/Cybersecurity_HSE/blob/main/HomeWorks/DevSecOps/HW05/report/trivy_report.txt>

# Этап 3. Сборка в контейнера

Bash:

*docker build –t pygoat:latest .*

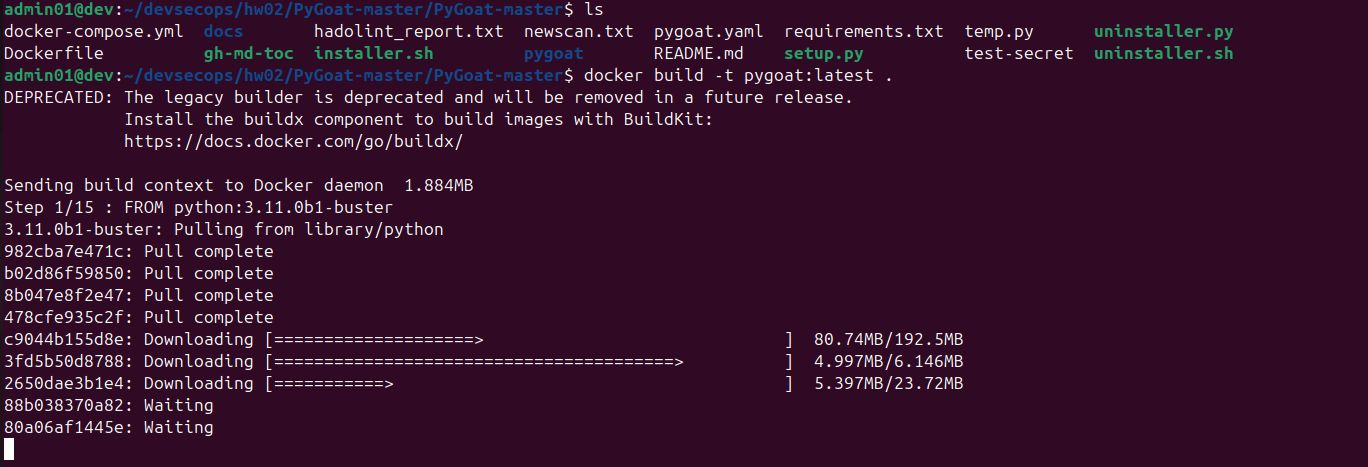


Рисунок 3. «сборка»

# Этап 4. Проверка кластера Kubernetes с kube-bench

## Запуск проверки:

Bash:

*kubectl apply -f https://raw.githubusercontent.com/aquasecurity/kube-bench/main/job.yaml*

*kubectl logs -l job-name=kube-bench*

## Результат:

== Summary total ==

57 checks PASS

14 checks FAIL

62 checks WARN

Ссылка на github:

<https://github.com/vit81g/Cybersecurity_HSE/blob/main/HomeWorks/DevSecOps/HW05/report/04_kube_check.txt>

# Этап 5. Запуск PyGoat в Docker

Сборка и запуск:

Bash:

*docker build -t pygoat:latest -f config/Dockerfile .*

*docker run -d -p 8000:8000 pygoat:latest*

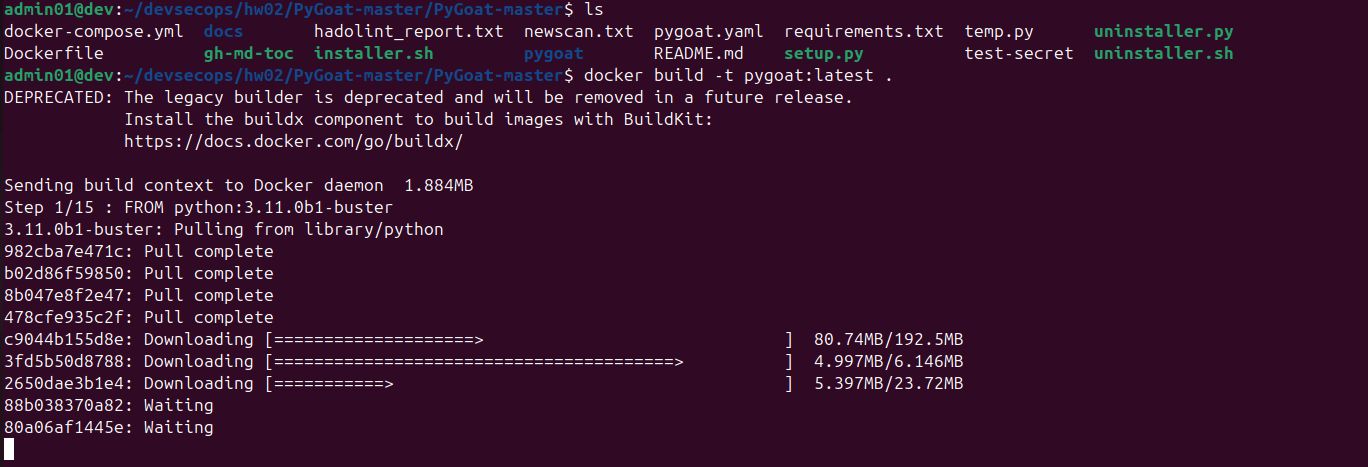


Рисунок 3 «Скриншот сборки»

Результат: Приложение доступно по адресу <http://localhost:8000>

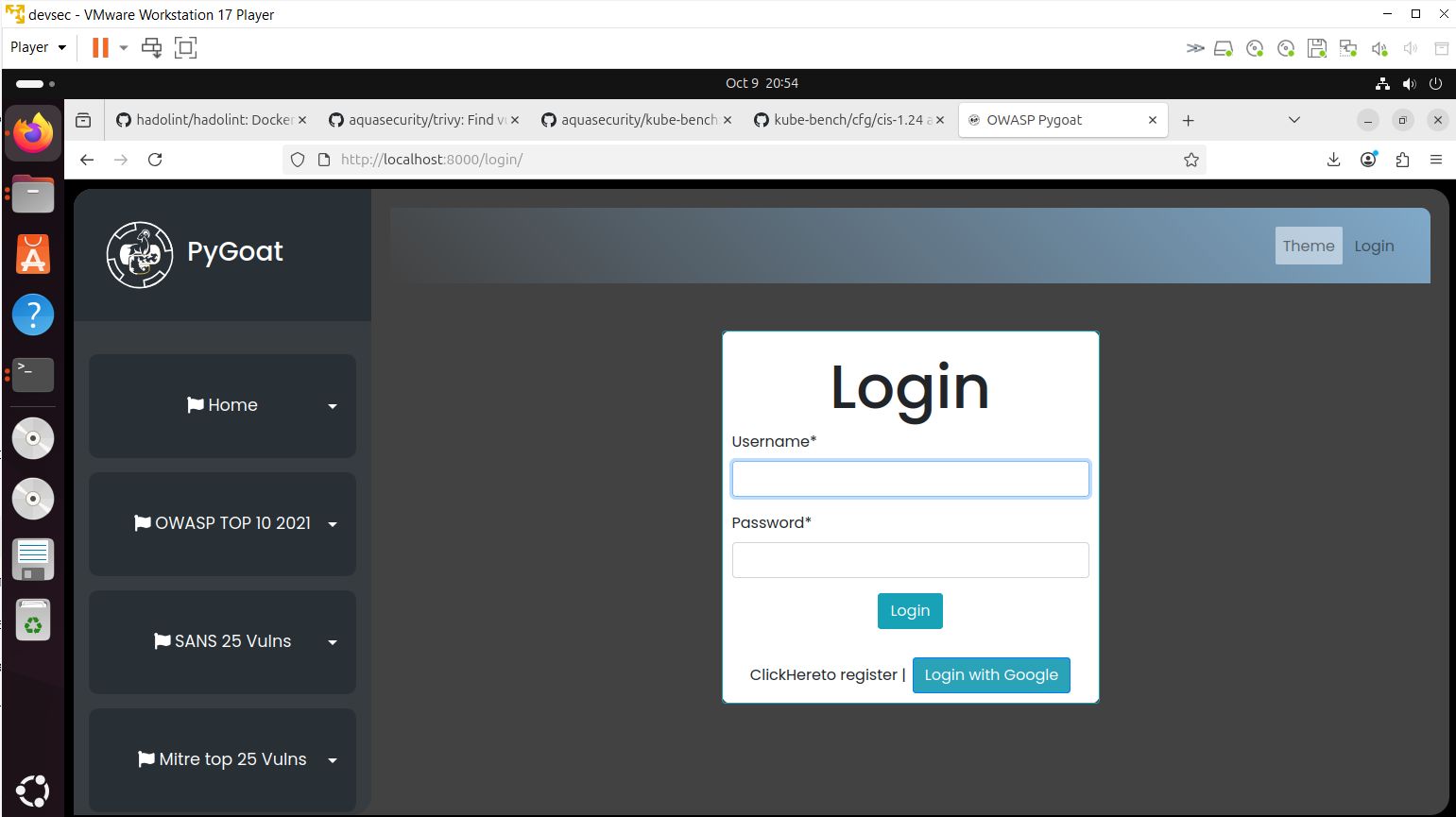


Рисунок 4. «работа приложения»

# Этап 6. Развёртывание PyGoat в Kubernetes

## Манифесты Deployment и Service:

pygoat-deployment.yaml:

<https://github.com/vit81g/Cybersecurity_HSE/blob/main/HomeWorks/DevSecOps/HW05/config/pygoat-deployment.yaml>

pygoat-service.yaml:

<https://github.com/vit81g/Cybersecurity_HSE/blob/main/HomeWorks/DevSecOps/HW05/config/pygoat-service.yaml>

## Применение:

Bash:

*kubectl apply -f config/pygoat-deployment.yaml*

*kubectl apply -f config/pygoat-service.yaml*

*kubectl get pods*

*kubectl get svc*

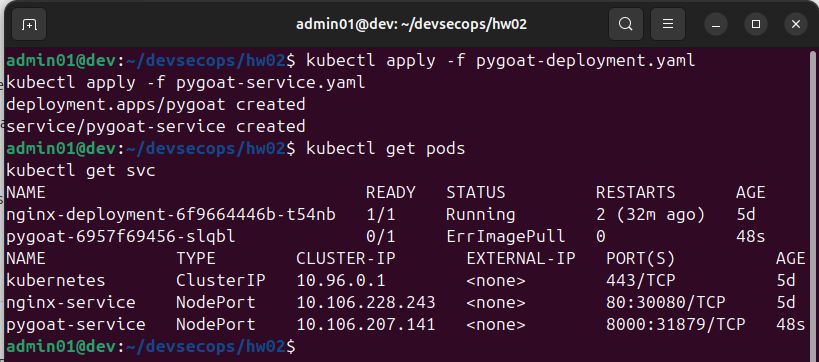


Рисунок 5. «проверка»

NodePort Kubernetes: 31879

(это автоматически назначенный порт, через который доступ к сервису открывается снаружи узла)

http://10.106.207.141:31879