Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Высшая школа кибернетики и цифровых технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

ПО КУРСУ «ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ»

«ВВЕДЕНИЕ В DOCKER»

Выполнил: студент 2 курса группы ПО(б) – 31

Черных Данил Анатольевич

Проверила: ассистент ВШ КЦТ

Балабасова Алиса Дмитриевна

Хабаровск 2025 г.

Цель: освоить основы контейнеризации с Docker.

Задания:

1. Установить Docker на свою систему.

2. Создать Dockerfile для простого приложения (например, на C++ с компиляцией и запуском).

3. Собрать Docker-образ и запустить контейнер.

4. Опубликовать образ в Docker Hub.

5. Настроить автоматический деплой контейнера через GitHub Actions.

6. Запустить контейнер из любого публичного образа и проверить его работу.

Ход работы:

На официальном сайте Docker – docker.com, перейдем в раздел «Docker Desktop», здесь скачаем установщик для Windows – AMD64.

Установим Docker Desktop на компьютер, вместе с ним устанавливается также WLS (Подсистемы Windows для Linux). Взаимодействие с Docker осуществляется через приложение или терминал.

В папке проекта создадим Dockerfile (рисунок 1) для проекта на C++, образ будем основывать на Ubuntu:24.04, установим все необходимые программы и зададим рабочую папку по умолчанию «/app», скопируем в нее содержимое директории программы и зададим команды для компиляции и запуска программы.

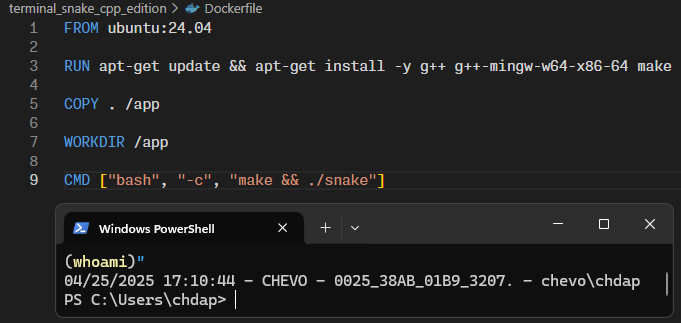


Рисунок 1 – Содержимое Dockerfile

Соберем образ, используя команду «docker build» (рисунок 2):

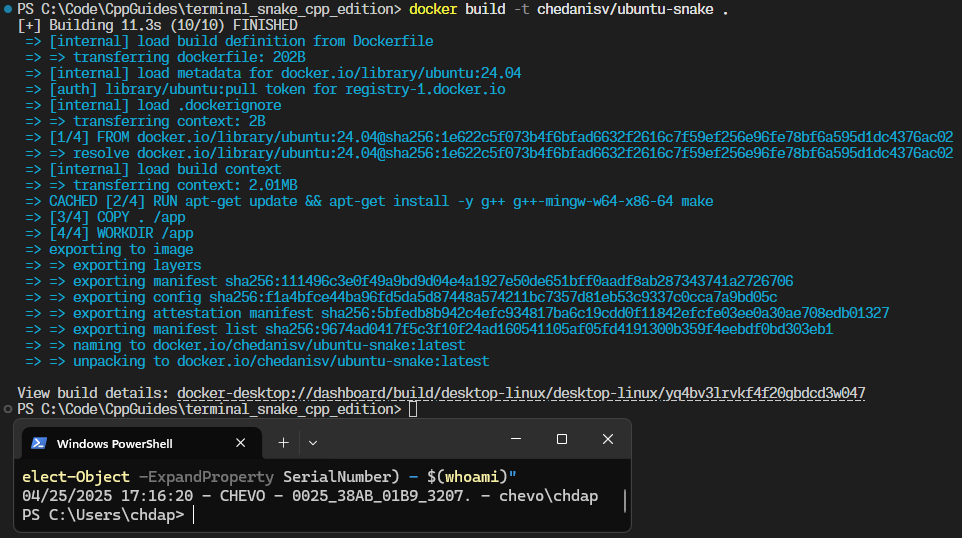


Рисунок 2 – Сборка образа

Запустим собранный образ (рисунок 3):

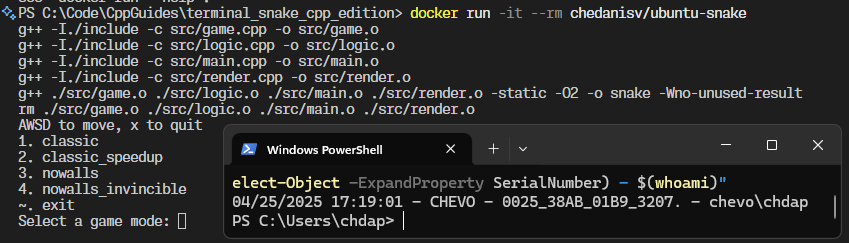


Рисунок 3 – Запуск образа, компиляция и запуск программы

Прежде чем публиковать данный образ, с помощью команды «docker tag» добавим к нему тег «dev», чтобы обозначить, что данная версия не является релизом. Релизные версии программы будут иметь тег вида «v1.2.3».

Далее авторизируемся в Docker Hub с помощью «docker login» и опубликуем образ с помощью «docker push» (рисунок 4).

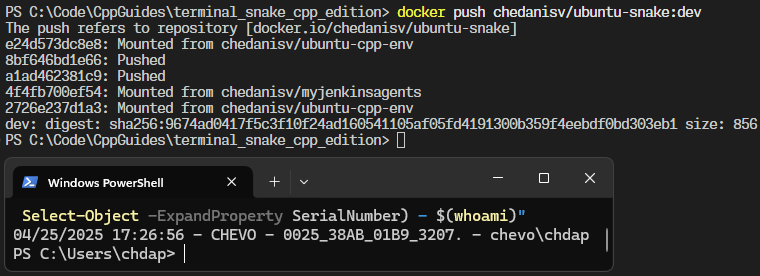


Рисунок 4 – Публикация образа на Docker Hub

С помощью GitHub Actions настроим автоматический деплой образа с новой версией программы в Docker Hub.

Создадим новый тег командой «git tag v0.2.7», запушим тег в репозиторий и посмотрим за работой GitHub Actions (рисунок 5).

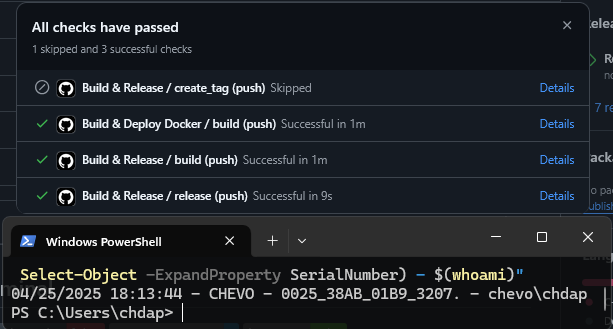


Рисунок 5 – Краткий результат сборки для Docker и GitHub

После завершения работы GitHub Actions проверим наличие новой версии на Docker Hub (рисунок 6).

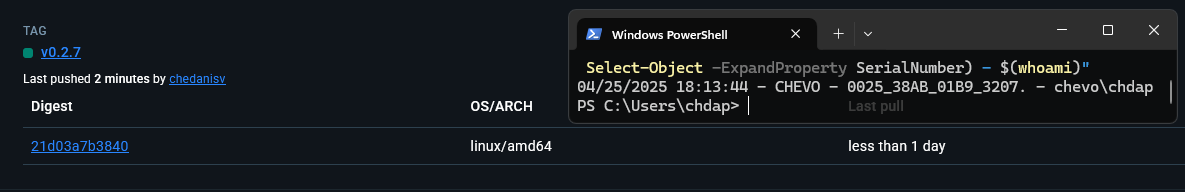


Рисунок 6 – Новая версия на Docker Hub

Проверим работоспособность созданного контейнера: скачаем его с помощью команды «docker pull» и запустим с помощью «docker run» (рисунок 7).

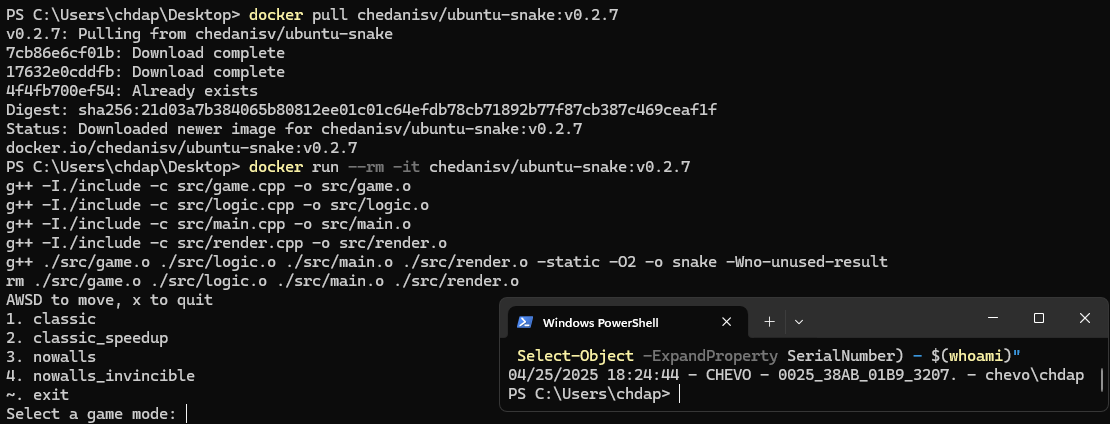


Рисунок 7 – Запуск Docker контейнера

Вывод:

В ходе выполнения работы был установлен Docker Desktop, создан Dockerfile и с его помощью собран Docker образ и запущен контейнер. Данный образ был опубликован в Docker Hub. Далее с помощью GitHub Actions была реализована автоматическая сборка и загрузка новых версий образа в Docker Hub.

Опыт, полученный в ходе выполнения данной лабораторной работы, будет полезен в будущем при работе с Docker.