Государственное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

09.03.02 320008

**«Системное программное обеспечение»**

**ОТЧЁТ**

По лабораторным работам №2, №3



Преподаватель: Кулемин А.Н.

Студент: Стрельников М. А.

Группа: Фт — 320008

Екатеринбург 2024

Цель работы: создание контейнера с использованием Docker, освоение

сопутствующих навыков. Тестирование и мониторинг его работы. Создание репозитория.

Выполнение работы:

Был создан репозиторий https://github.com/m3kskssssssss/Atomny\_Horse. В работе мы запустим этот репозиторий используя Docker

Для создания Docker image нам нужен специальный файл, который не имеет расширения и называется Dockerfile.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рис. 3 – Файл Dockerfile

* **FROM node:18**: Эта команда указывает, что базовым образом для вашего контейнера будет образ Node.js версии 18.
* **WORKDIR /app**: Эта команда устанавливает рабочую директорию для всех последующих команд.
* **COPY package\*.json ./** : Эта команда копирует файлы package.json и package-lock.json из локального контекста сборки текущую рабочую директорию контейнера (/app).
* **RUN npm install**: Эта команда выполняет установку всех зависимостей, указанных в package.json. Она создает папку node\_modules в контейнере, где будут храниться все установленные пакеты.
* **COPY . .**: Эта команда копирует все файлы и папки из локального контекста сборки в текущую рабочую директорию контейнера (/app). Это включает в себя исходный код приложения и все другие необходимые файлы.
* **RUN npm run build**: Эта команда выполняет скрипт build, указанный в package.json. Этот скрипт компилирует приложение, создавая оптимизированные статические файлы, которые будут развернуты на веб-сервере.
* **RUN npm install --save-dev gh-pages**: Эта команда устанавливает пакет gh-pages как зависимость для разработки. Этот пакет используется для развертывания приложения на GitHub Pages.
* **RUN npm run deploy**: Эта команда выполняет скрипт deploy, указанный в package.json. Этот скрипт использует gh-pages для развертывания содержимого папки сборки на GitHub Pages.

Далее работа будет производиться в стандартной cmd консоли Windows и такого приложения как Docker desktop.

В cmd терминале вводим команду:

docker build -t m3kskssssssssssss/atomny\_horse .

Изображение выглядит как текст, снимок экрана, веб-страница, программное обеспечение

Автоматически созданное описание

Рис. 4 – Сборка образа Docker

И запускаем образ командой docker run -d m3kskssssssssssss/atomny\_horse

Изображение выглядит как текст, снимок экрана, прямоугольный, кроссворд

Автоматически созданное описание

Рис. 5 – Проверка работы веб-приложения

После успешного тестирования работоспособности наш docker контейнер надо выложить на Docker Hub. Docker Hub – это хранилище огромного количества образов, куда каждый пользователь может выложить свой образ, и любой другой человек – скачать его.

Для того чтобы выложить свой docker-образ нужно сперва залогиниться в своем аккаунте в Docker Hub командой docker login. После этого даем имя и командой: docker push m3kskssssssssssss/atomny\_horse:latest публикуем

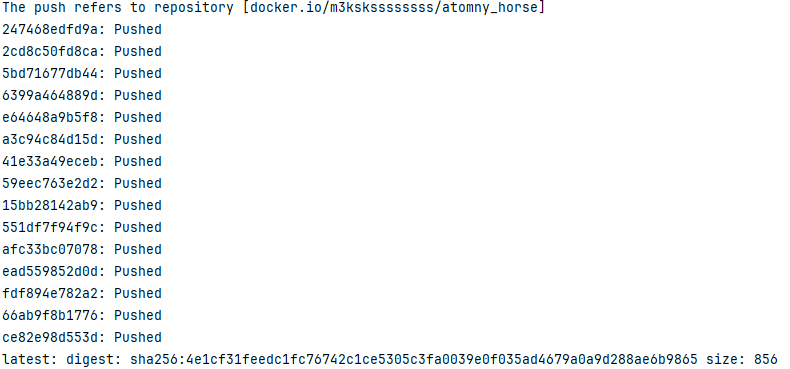


Рис. 6 – Публикуем образ на Docker Hub

Всё прошло успешно. Теперь можно заходить в аккаунт и проверять, отобразился ли там наш контейнер.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис. 7 – Интерфейс Docker Hub

Как мы видим, контейнер успешно опубликован, и теперь каждый может скачать его себе на компьютер, или увидеть на сайте по ссылке: <https://hub.docker.com/repositories/m3kskssssssss>

Теперь нам надо провести мониторинг работы системы. Будем мониторить наше веб-приложение docker desktop во вкладке stats.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Рис.8 – мониторинг веб приложения

Заключение

В данной лабораторной работе мы освоили работу с контейнерами, а именно с созданием и запуском изображений, их мониторингом и загрузкой на Docker Hub. Все цели лабораторной работы были успешно выполнены.