МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ образовательное учреждениевысшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации



**ОТЧЁТ по Лабораторной работе №4**

**по дисциплине «ПРОГРАММИРОВАНИЕ»**

**«Основы работы с Docker: Введение в контейнеризацию»**

|  |  |
| --- | --- |
| Факультет: АВТФ  Группа: АБс-222  Студент: Гатауллин Д.Р. | Преподаватель: Архипова А.Б. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**Цели и задачи работы**: Изучение принципов работы Dockerfile и Docker Compose и практические применение возможностей развёртывания приложений в контейнеризованной среде.

**Теоретическая часть**:

Docker — это платформа контейнеризации с открытым исходным кодом, с помощью которой можно автоматизировать создание приложений, их доставку и управление. Платформа позволяет быстрее тестировать и выкладывать приложения, запускать на одной машине требуемое количество контейнеров.

Основные концепции Docker включают:

* Образы Docker (Docker Images): Образ Docker — это легковесный, автономный, исполняемый пакет программного обеспечения, который включает все, что необходимо для запуска кода: системную библиотеку, системные инструменты, код, среду выполнения, настройки.
* Контейнеры Docker (Docker Containers): Контейнер — это среда выполнения для образа Docker. Он представляет собой полностью изолированное окружение, в котором работает приложение.
* Dockerfile: Dockerfile — это текстовый документ, который содержит все команды, которые пользователь может вызвать в командной строке для создания образа.
* Docker Compose: Docker Compose — это инструмент для определения и запуска многоконтейнерных приложений Docker. С его помощью вы можете использовать файл YAML для настройки служб вашего приложения.
* Docker Hub: Docker Hub — это облачный сервис для совместного использования и версионирования образов Docker. Это место, где вы можете создавать свои образы Docker и делиться ими с другими.

**Ход выполнения работы**:

1. Для начала посмотрел, что из себя представляет докер, его возможности и примеры его применений на практике, посмотрев небольшой обучающий курс

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

1. Затем я установил с официального сайта непосредственно докер и запустил первоначальную страничку докера в виде контейнера, при этом знакомился с основными командами докера

docker version – просмотр текущей версии

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

docker run -d -p 80:80 docker/getting-started – запуск в фоновом режиме с связанным портом 80

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

docker info – просмотр количество контейнеров и основной информации

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

docker images – просмотр всех образов

docker ps -a – просмотр всех контейнеров

Изображение выглядит как снимок экрана, текст, Шрифт

Автоматически созданное описание

docker kill – убить контейнер (принудительно остановить)

docker rm – удалить контейнер

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. На этом этапе я запускаю среду выполнения команд Python при помощи интерактивного режима докера

docker run -it –name Python python – запуск в интерактивном режиме образа python в контейнере Python

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. В этом пункте я рассматривал возможность внедрения в среду разработки VCode самого докера

FROM – из какого образа

COPY – откуда (хост машина) и куда (виртуальная машина) пойдут сами файлы для контейнера

WORKDIR – рабочая директория виртуальной машины

RUN – команда для компиляции нашего проекта (в данном случае .c файла)

CMD – команда для запуска нашего скомпилированного файла

EXPOSE – порт виртуальной машины

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

1. Далее я знакомлюсь с docker compose

version – указывает версию compose

services – хранилище образов

php, db... – названия (любое)

image – название образа

restart – функция, позволяющая перезапускать (или ничего не делать) при «форс-мажоре» контейнеры

ports – связь между портами хост системы и виртуальной

остальное – дополнительные настройки

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

1. И в завершении я делаю загрузку и выгрузку своего образа

docker build -t gatikgubkabob/myiso ./php – команда, которая строит образ с названием gatikgubkabob/myiso из папки php

docker push gatikgubkabob/myiso:latest – выгрузка своего образа

docker pull gatikgubkabob/myiso – загрузка своего образа

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Текст программы:**

==========================================================

<https://github.com/SpongebobGatik/GataullinDocker>

==========================================================

**Вывод**: Docker представляет собой мощный инструмент, который позволяет разработчикам упаковывать приложения вместе со всеми их зависимостями в стандартизованные блоки, называемые контейнеры. Это обеспечивает консистентность на всех этапах разработки и развертывания, что в свою очередь увеличивает производительность, упрощает управление и улучшает безопасность.