МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Кемеровский государственный университет»**

**Институт фундаментальных наук**

**Кафедра ЮНЕСКО по ИВТ**

**ЛАБОРАТОРНАЯ РАБОТА №5**

**ПО ДИСЦИПЛИНЕ “Автоматизация процессов развертывания и инсталляции информационных систем”**

Направление 09.03.03 – Прикладная информатика в экономике

Студента 2 курса

Башкеева Степана Дмитриевича

Преподаватель:

И.Ю. Степанов

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

Работа защищена:

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г.

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Кемерово 2022

Содержание

[Цели работы 2](#_Toc1793214634)

[Задачи 3](#_Toc1722456179)

[Описание предметной области 3](#_Toc2108959824)

[Практическая часть 4](#_Toc1897849432)

[Заключение 9](#_Toc648387724)

[Список литературы 9](#_Toc1869527046)

Цели работы

Цель данной лабораторной работы изучение документации по Docker, научиться работать с контейнерами, также нужно произвести работу с операционной системой Ubuntu.

# Задачи

* Изучить документацию по Docker.
* Научиться работать с контейнерами.
* Произвести работы с Ubuntu.

# Описание предметной области

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть развёрнут на любой Linux-системе с поддержкой контрольных групп в ядре, а также предоставляет набор команд для управления этими контейнерами. Изначально использовал возможности LXC, с 2015 года начал использовать собственную библиотеку, абстрагирующую виртуализационные возможности ядра Linux — libcontainer. С появлением Open Container Initiative начался переход от монолитной к модульной архитектуре. [1]

Ubuntu — это разрабатываемая сообществом, основанная на ядре Linux операционная система, которая идеально подходит для использования на персональных компьютерах, ноутбуках и серверах. Она содержит все необходимые программы, которые вам нужны: программу просмотра Интернет, офисный пакет для работы с текстами, электронными таблицами и презентациями, программы для общения в Интернет и много других. [2]

# Практическая часть

1. Зачем нужен Докер? Чем он отличается от виртуальной машины?

Docker — это платформа для разработки, доставки и запуска контейнерных приложений. Docker позволяет создавать контейнеры, автоматизировать их запуск и развертывание, управляет жизненным циклом. Он позволяет запускать множество контейнеров на одной хост-машине.

Виртуальные машины имеют хост-ОС и гостевую ОС внутри каждой виртуальной машины. Гостевой ОС может быть любая ОС, например, Linux или Windows, независимо от хост-ОС. Контейнеры Docker, напротив, размещаются на одном физическом сервере с операционной системой хоста, которая разделяет их между собой.

1. В чем отличие контейнера и образа?

Контейнер - это просто образ. При создании контейнера поверх образа добавляет слой, доступный для записи, что позволяет менять его по своему усмотрению. Образ - это шаблон, на основе которого создается контейнер, существует отдельно и не может быть изменен.

1. Установка Docker.

Переходим на официальный сайт docker.com и скачиваем и устанавливаем его (рис. 1-2).



Рис. 1. Скачивание Docker Desktop Installer.

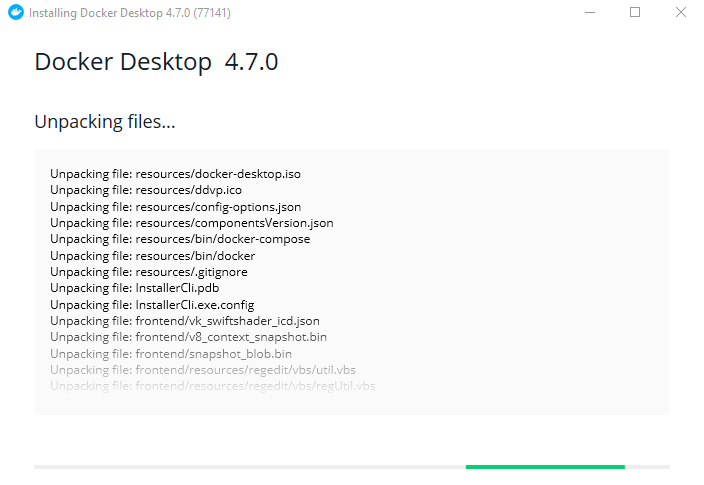


Рис. 2. Установка Docker Desktop Installer.

1. Скачать образ Ubuntu.

Чтобы скачать образ Ubuntu, необходимо в командной строке вписать команду “docker pull ubuntu” (рис. 3).

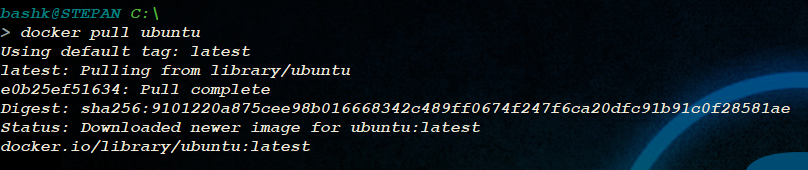


Рис. 3. Скачивание образа Ubuntu.

1. Просмотр списка всех загруженных образов.

Для просмотра списка всех загруженных образов, необходимо вписать команду “docker images” в командной строке. Результат работы выполнения команды проиллюстрирван на рисунке 4.

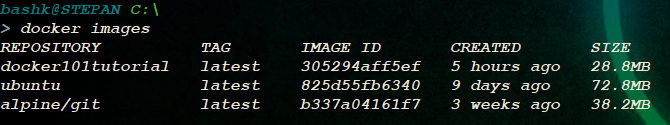


Рис. 4. Список загруженных образов.

1. Работа с контейнером.

Вначале создадим контейнер для дальнейшей работы с ним (рис. 5).

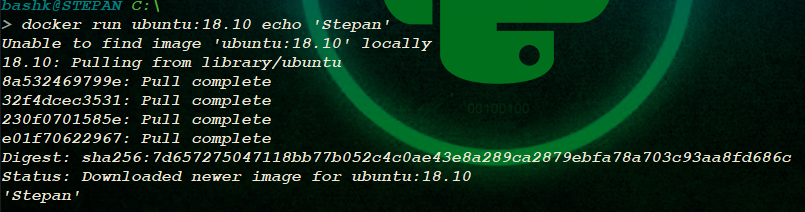


Рис. 5. Создание контейнера.

С помощью команды “docker ps -a” можно посмотреть какие контейнеры активны, также можем посмотреть информацию о каждом контейнере. Например, можно узнать ID контейнера, он понадобится для дальнейшей работы. В нашем случаи ID = d406b09793c1 (рис. 6). [3]

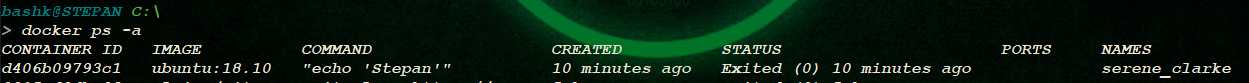


Рис. 6. Проверка контейнера.

Для остановки контейнера, необходимо вписать команду “docker stop (ID контейнера)” (рис. 7).

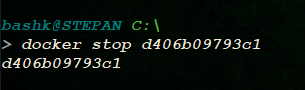


Рис. 7. Остановка контейнера.

Чтобы удалить контейнер, нужно вписать команду “docker rm (ID контейнера)” (рис. 8).

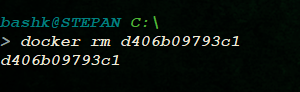


Рис. 8. Удаление контейнера.

1. Создание DockerFile и отображения пирамидки.

Для начала создадим отдельную папку, в которой будем хранить файлы “DockerFile” и “cli.php”. В файле “cli.php” пишем алгоритм отображения пирамидки (рис. 9-10).

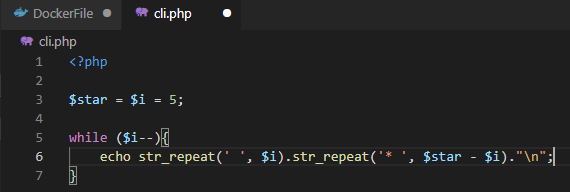


Рис. 9. Файл “cli.php”.

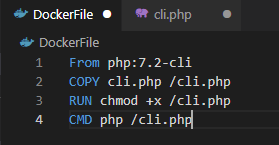


Рис. 10. Файл “DockerFile”.

Далее переходим в папку с этими файлами через командную строку, для это нужно вписать команду “cd C:\(путь к папке)” (рис. 11).



Рис. 11. Обращение к папке с файлами.

Затем необходимо вписать команду “docker build . --tag pyramid”, чтобы создать пирамидку (рис. 12).



Рис. 12. Создание пирамидки.

Теперь вписываем команду “docker run pyramid”, которая отобразит пирамидку (рис. 13).

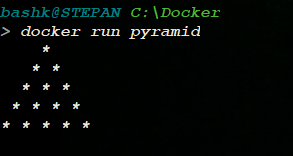


Рис. 13. Отображение пирамидки.

# Заключение

В результате лабораторной работы была изучена документация Docker, также был получен опыт работы с контейнерами и с DockerFile, с помощью которого была отображена пирамидка. Были произведены работы с операционной системой Ubuntu.

# Список литературы

1. Docker [Электрон. сервис] \ URL - <https://habr.com/ru/company/ruvds/blog/438796/>
2. Русскоязычная документация по Ubuntu [Электрон. сервис] \ URL - <https://help.ubuntu.ru/manual/введение>
3. Docker документация [Электрон. сервис] \ URL - <https://dker.ru/docs/>