Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Интеграция и развертывание программного обеспечения с помощью контейнеров  
**Лабораторная работа №2**

**Тема:**

«Создание Dockerfile и сборка образа»

Выполнил(а): st\_98, группа: АДЭУ-211

Преподаватель:

Москва

2025

**Вариант 11. Задача:** Создайте Dockerfile для приложения на Python, которое использует библиотеку Pandas для чтения CSV-файла и вывода его содержимого.

**Цель работы:** научиться создавать Dockerfile и собирать образы Docker для приложений.

**Ход работы:**

1. Создадим виртуальное окружение и установим библиотеку pandas для выполнения задания. Выполнения данного этапа представлено на рисунке 1.

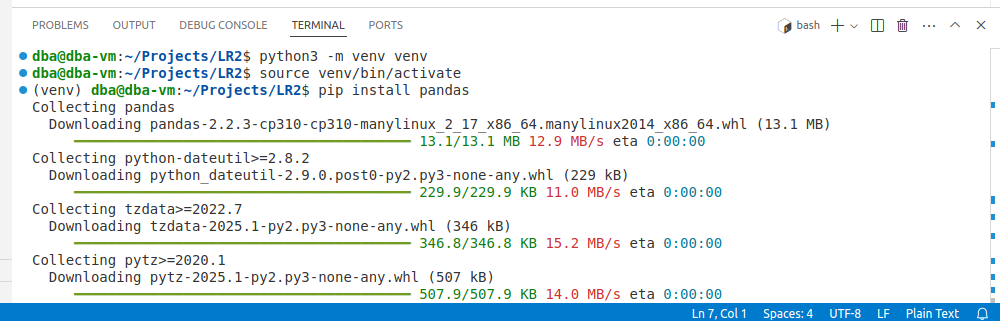


Рисунок – Создание виртуального окружения

1. Для выполнения задания создаем скрипт для чтения CSV-файла и вывода его содержимого. Код представлен на рисунке 2.

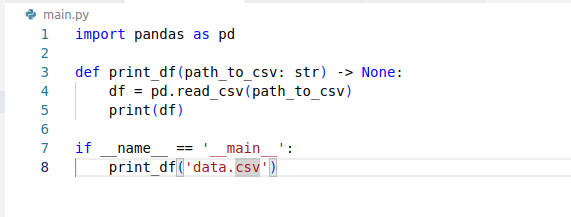


Рисунок 2 - Скрипт для чтения CSV-файла

1. Далее создаем файл requirements.txt со всеми зависимостями. Команда и файл представлены на рисунке 3.

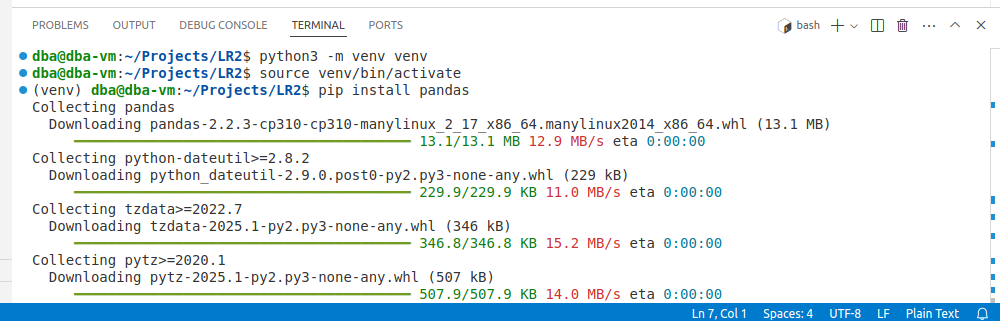


Рисунок 3 – Файл с зависимостями

1. Следующим шагом создаем Dockerfile и собираем образ. Данный процесс показан на рисунке 4.

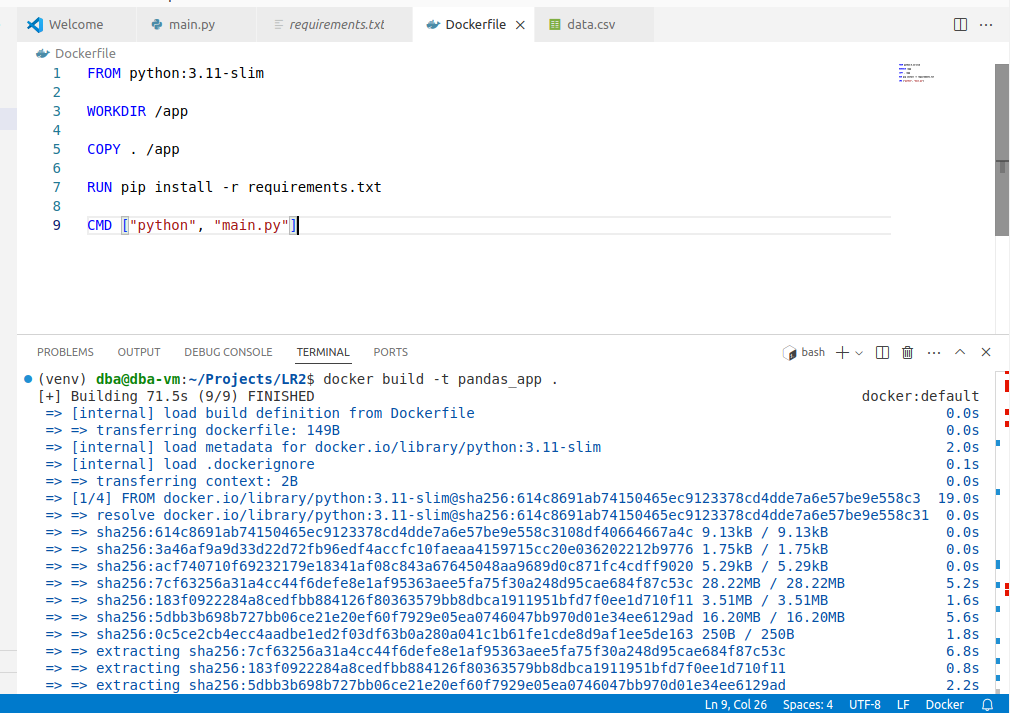


Рисунок 4 – Сборка образа

1. Запускаем данный контейнер и просматриваем вывод. На рисунке 5 можно заметить, что загруженный файл data.csv был успешно прочитан и его результат выведен в терминале.

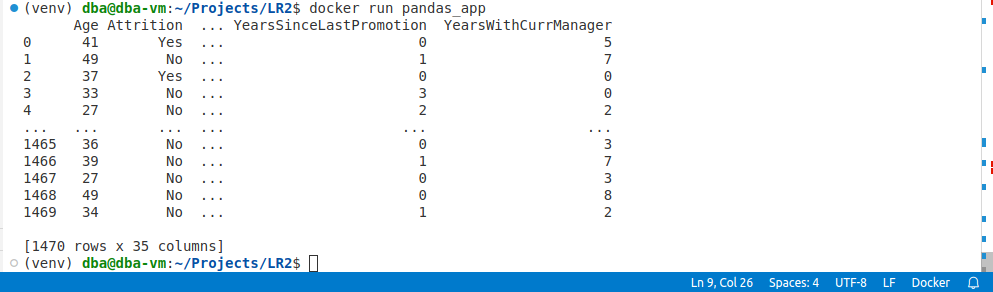


Рисунок 5 – Результаты выполнения скрипта

**Вывод:**

В ходе выполнения задания был создан Dockerfile для приложения на Python, которое использует библиотеку Pandas для чтения и вывода содержимого CSV-файла. Основные этапы работы включали:

* Написан Dockerfile, в котором указаны базовый образ Python, установка необходимых зависимостей (библиотеки Pandas) и копирование исходного кода приложения в контейнер.
* С помощью команды docker build был собран Docker-образ на основе созданного Dockerfile.
* Контейнер был запущен с использованием команды docker run, и приложение корректно прочитало CSV-файл, выведя его содержимое в терминал.
* Убедились, что приложение работает как ожидалось.

**Контрольные вопросы:**

1. Что такое Dockerfile и для чего он используется?

Dockerfile — это текстовый файл, содержащий набор инструкций для создания Docker-образа. Он описывает, как настроить среду внутри контейнера, какие зависимости установить, какие файлы скопировать и какие команды выполнить. Dockerfile используется для автоматизации процесса создания образов, что делает их воспроизводимыми и переносимыми.

1. Какие основные инструкции используются в Dockerfile?

* **FROM:** указывает базовый образ, на основе которого будет создаваться новый образ. Например:
* **WORKDIR:** задаёт рабочую директорию внутри контейнера, где будут выполняться последующие команды.
* **COPY:** копирует файлы или директории из хоста в контейнер.
* **RUN:** выполняет команды внутри контейнера во время сборки образа. Обычно используется для установки зависимостей.
* **CMD:** указывает команду, которая будет выполнена при запуске контейнера. Может быть переопределена при запуске контейнера.
* **EXPOSE:** указывает порты, которые будут открыты в контейнере.

1. Как выполняется сборка образа Docker с использованием Dockerfile?
2. Создается Dockerfile в корневой директории проекта
3. Выполняется команда для сборки образа: docker build
4. Как запустить контейнер из собранного образа?

После сборки образа необходимо запустить контейнер с помощью команды: docker run

1. Каковы преимущества использования Dockerfile для создания образов Docker?

* Автоматизация: Dockerfile позволяет автоматизировать процесс создания образов, что упрощает развертывание приложений.
* Воспроизводимость: Образы, созданные с помощью Dockerfile, всегда одинаковы, что исключает проблемы с различиями в окружениях.
* Переносимость: Docker-образы можно использовать на любой системе, где установлен Docker.
* Масштабируемость: Dockerfile позволяет легко создавать и обновлять образы для различных версий приложений.
* Изоляция: Контейнеры, созданные из образов, изолированы от хостовой системы, что повышает безопасность и стабильность.
* Управление зависимостями: Все зависимости приложения описываются в Dockerfile, что упрощает управление ими.