

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

Макарова Екатерина Павловна

ЛАБОРАТОРНАЯ РАБОТА 1

Проектный практикум по разработке ETL-решений

Направление подготовки

38.03.05 Бизнес-информатика

Профиль подготовки

Аналитика данных и эффективное управление

Курс обучения: 4

Форма обучения: очная

Преподаватель: кандидат технических наук,
доцент Босенко Тимур Муртазович

Москва

2025

Цель работы: освоить процесс установки и настройки Docker, научиться работать с контейнерами и образами Docker.

Задачи:

1. Установить Docker на локальный компьютер.
2. Проверить корректность установки Docker.
3. Ознакомиться с основными командами Docker CLI для работы с образами и контейнерами.
4. Выполнить индивидуальное задание.

Вариант 8 «Загрузить образ mongo, запустить контейнер, подключиться к MongoDB внутри контейнера и создать новую коллекцию»

Ход работы

1. Установка Docker Desktop на локальный компьютер

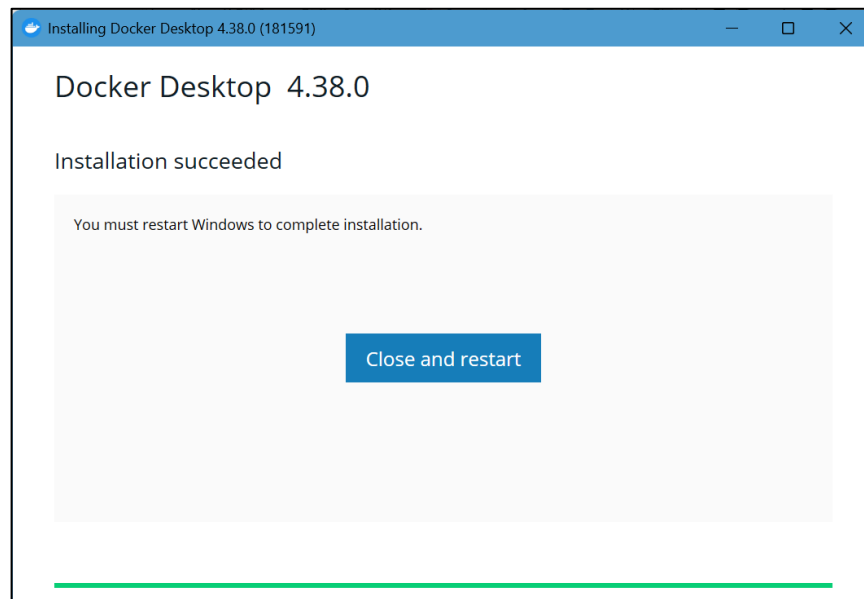


Рис. 1

Успешный запуск Docker Desktop (Рис. 2).

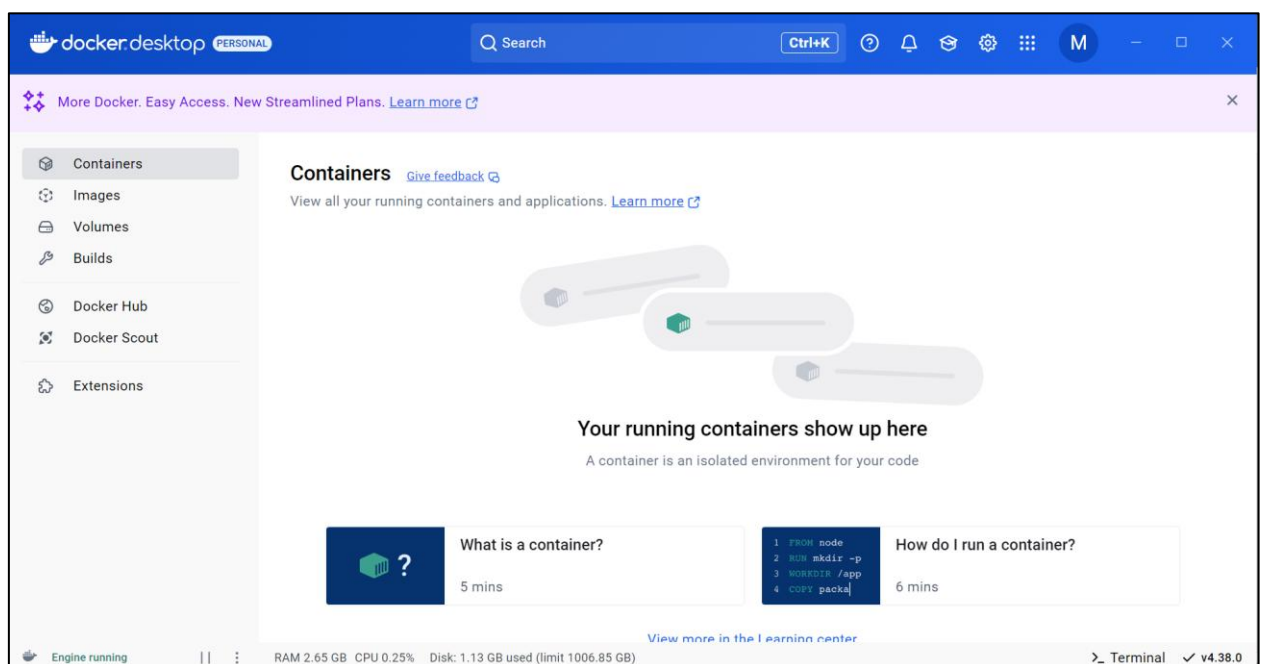


Рис. 2

2. Проверка установки:

1) Проверка версии Docker (Рис. 3).

```
PS C:\Users\Dell> docker --version
Docker version 27.5.1, build 9f9e405
PS C:\Users\Dell> |
```

Рис. 3

2) Выполнение команды на загрузку тестового образа и запуск его в контейнере (Рис. 4).

```
PS C:\Users\Dell> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Download complete
Digest: sha256:e0b569a5163a5e6be84e210a2587e7d447e08f87a0e90798363fa44a0464a1e8
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Рис. 4

3. Ознакомление с основными командами

1) Просмотреть список локальных образов Docker.

```
PS C:\Users\Dell> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	e0b569a5163a	4 weeks ago	20.4kB

Рис. 5

2) Просмотреть список запущенных контейнеров

```
PS C:\Users\Dell> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Рис. 6

3) Просмотреть список всех контейнеров, включая остановленные

```
PS C:\Users\Dell> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e9cde93f562d	hello-world	"/hello"	4 minutes ago	Exited (0) 4 minutes ago		unruffled_aryabh
ata						

Рис. 7

4. Выполнение индивидуального задания

Загрузить образ mongo, запустить контейнер, подключиться к MongoDB внутри контейнера и создать новую коллекцию.

1) Установка MongoDB в Docker

```
PS C:\Users\Dell> docker run --name my-mongo -d mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
07b085c89153: Download complete
990f24c1b076: Download complete
1aa9099a841a: Download complete
a53a70bd8c23: Download complete
8170e4b21000: Download complete
5a7813e071bf: Download complete
9c76b9a0da25: Download complete
```

Рис. 8

Команда docker вызывает утилиту docker, запускает ее подкоманду для запуска контейнера, опция --name помогает обозначить контейнер так, как вам нужно, а опция -d указывает на образ из репозитория.

2) Проверка списка запущенных контейнеров (Рис. 9).

```
PS C:\Users\Dell> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ES						
9925309afa40	mongo	"docker-entrypoint.s..."	About a minute ago	Up About a minute	27017/tcp	my-mongo

Рис. 9

3) Подключение к запущенному Docker-контейнеру (Рис. 10).

```
9925309afa40 mongo docker exec -it my-mongo bash
kate@DESKTOP-9KCDQIL:~$ docker exec -it my-mongo bash
root@9925309afa40:/# |
```

Рис. 10

4) Подключение к CLI MongoDB (Рис. 11).

```
root@9925309afa40:/# mongosh
Current Mongosh Log ID: 67b75d1ac3eb755c544ca6
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
.3.8
Using MongoDB:      8.0.4
Using Mongosh:      2.3.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
```

Рис. 11

5) Создание базы и коллекции внутри нее (Рис. 12).

```
test> use my_database
switched to db my_database
my_database> db.createCollection("kate_collection")
{ ok: 1 }
my_database> |
```

Рис. 12

6) Остановка контейнера (Рис. 13).

```
PS C:\Users\Dell> docker stop my-mongo
my-mongo
```

Рис. 13

Выводы по работе

В ходе выполнения лабораторной работы были освоены ключевые аспекты работы с Docker и MongoDB, что позволило успешно реализовать поставленные задачи.

Установка Docker Desktop и проверка его работоспособности подтвердили готовность среды для работы с контейнерами. Были изучены базовые команды CLI, такие как:

- `docker images` — управление образами;
- `docker ps` — мониторинг контейнеров;
- `docker run`, `docker stop`, `docker exec` — запуск, остановка и взаимодействие с контейнерами.

Задание по запуску контейнера MongoDB и созданию коллекции выполнено успешно. На практике применены:

- Проброс портов через `-p 27017:27017` для доступа к MongoDB с хоста;
- Использование команд MongoDB CLI (`mongosh`) для создания базы данных и коллекций.

На примере работы подтверждены ключевые преимущества контейнеризации:

- Изоляция: MongoDB работала в независимой среде, не затрагивая локальную систему.
- Портативность: образ MongoDB из репозитория Docker Hub запускался без дополнительных настроек.
- Эффективность: контейнер стартовал за секунды, потребляя минимум ресурсов.

Контрольные вопросы

1. Что такое Docker и для чего он используется?

Docker — это платформа для разработки, развертывания и управления приложениями в контейнерах. Контейнеры — это изолированные среды, которые включают код приложения, его зависимости и настройки. Docker используется для:

- Быстрого развертывания приложений в любом окружении (локально, в облаке, на сервере).
- Изоляции процессов и ресурсов (например, для микросервисной архитектуры).
- Автоматизации сборки и тестирования через CI/CD.

2. Какие преимущества дает использование контейнеров Docker по сравнению с виртуальными машинами?

- Эффективность ресурсов: контейнеры используют ядро хостовой ОС, а не эмулируют отдельную ОС, как виртуальные машины. Это снижает нагрузку на CPU, RAM и дисковое пространство.
- Скорость: контейнеры запускаются за секунды (вместо минут у VM).
- Портативность: образы Docker работают одинаково на любой ОС (Linux, Windows, macOS).
- Масштабируемость: легко запускать множество контейнеров на одном хосте.
- Упрощенное управление: инструменты Docker (Compose, Swarm, Kubernetes) упрощают оркестрацию.

3. Что такое образ Docker и как он связан с контейнерами?

Образ (Image) — это шаблон только для чтения, содержащий файловую систему, зависимости и настройки приложения.

Контейнер — это запущенный экземпляр образа. Контейнеры создаются из образов и изолированы от хоста и других контейнеров.

Связь образа с контейнером:

- Из одного образа можно создать множество контейнеров.
- Изменения в контейнере не влияют на исходный образ.

4. Какие основные команды Docker CLI вы узнали в ходе выполнения лабораторной работы?

- `docker run -d --name my-mongo -d mongo` - запуск контейнера в фоне;
- `docker ps` - список активных контейнеров;
- `docker ps -a` — показывает все контейнеры, независимо от их состояния;
- `docker stop my-mongo` — остановка контейнера;
- `docker images` — список локальных образов.

5. Как можно настроить маршрутизацию портов при запуске контейнера Docker?

1) Используя флаг `-p` при запуске контейнера, чтобы связать порт хоста с портом контейнера (Рис. 14).

```
kate@DESKTOP-9KCDQIL:~$ docker run -d --name my-mongo -p 27017:27017 mongo:latest  
56247ba619d6e9b6126eded76ec0e579d07bc61502b52f77e5b463735b268fc6
```

Рис. 14

2) Используя Docker Compose

В файле `docker-compose.yml` указать порты в секции `ports`:

```
kate@DESKTOP-9KCDQIL: ~/P × + v
version: '3.8'

services:
  mongo:
    image: mongo:latest
    ports:
      - "27017:27017"
    environment:
      MONGO_INITDB_ROOT_USERNAME: admin
      MONGO_INITDB_ROOT_PASSWORD: secret
    volumes:
      - mongo-data:/data/db # Сохранение данных

volumes:
  mongo-data:
  ~
  ~
  ~
```

Рис. 15