Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Инструкция по установке среды развёртывания *Microsoft SQL Server* на базе *Linux* контейнера в *Docker Desktop* для выполнения лабораторных работ

по дисциплине

«Системы управления Базами Данных»

Подготовил: студ. гр. 220604

А.И. Михнюк

Утвердил: Р.А. Шабанович

Минск 2024

**УСТАНОВКА *DOCKER DESKTOP***

Перед началом установки *Docker Desktop* требуется пройти процесс регистрации на [сайте *Docker*](https://app.docker.com/signup).

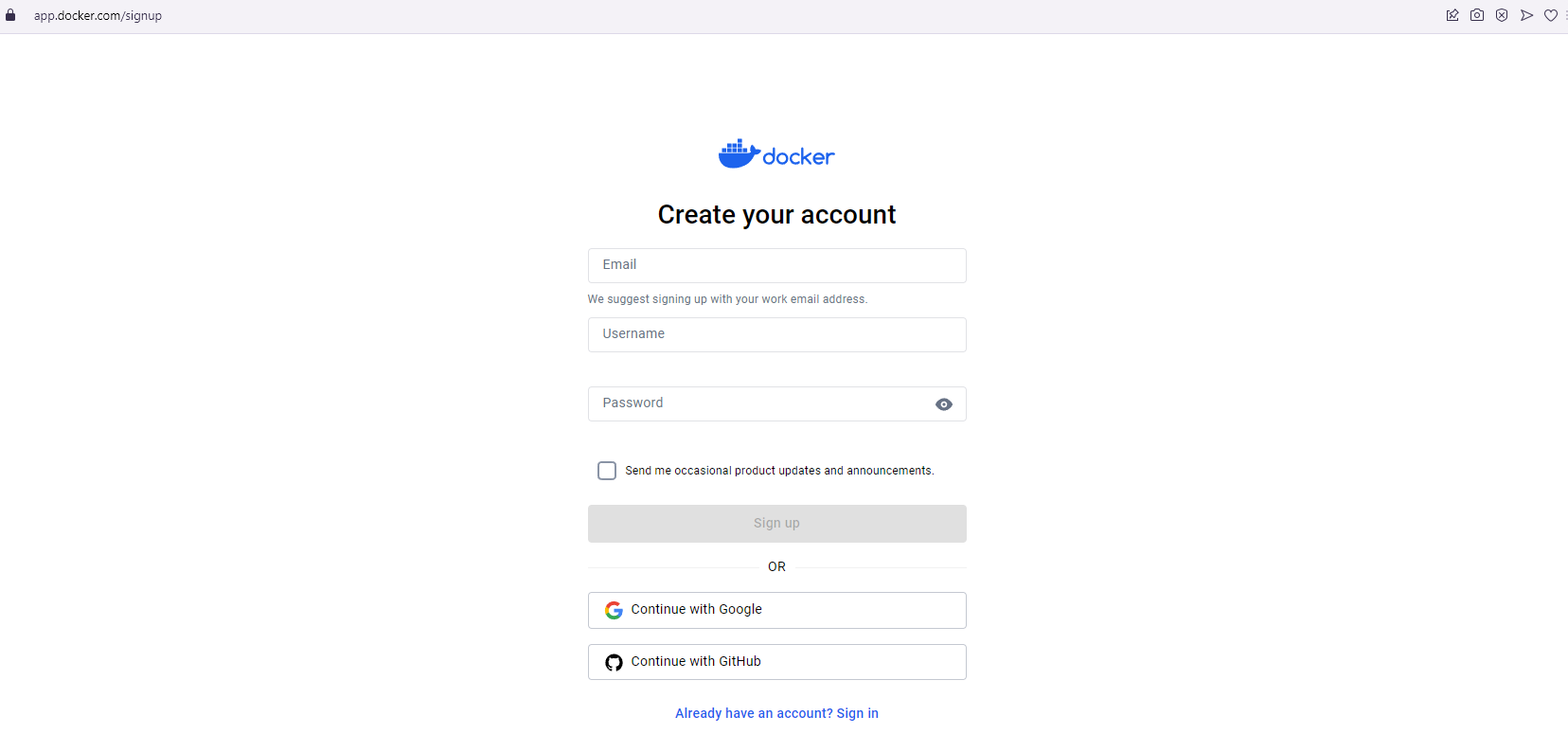


Рисунок 1 – Форма регистрации на сайте *Docker*

После успешной регистрации перед пользователем появляется плитка выбора продуктов компании *Docker*, из предложенного списка наводимся на *Docker Desktop* и нажимаем на кнопку «‎*Go to download*».

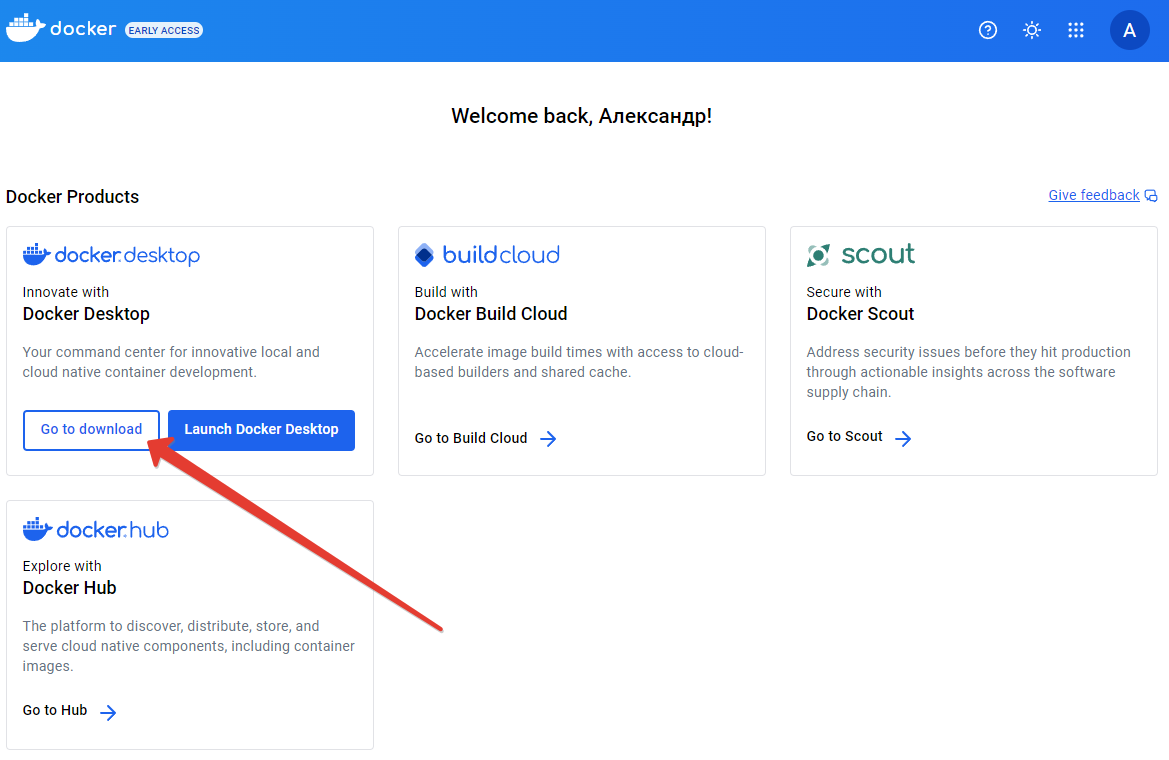


Рисунок 2 – Плитка продуктов *Docker*

После перехода на страницу нажатием клавиши «‎Go to download», открывается страница с документацией Docker, а именно – страница «‎Get Docker», на которой можно выбрать опцию «‎Docker Desktop for Windows».

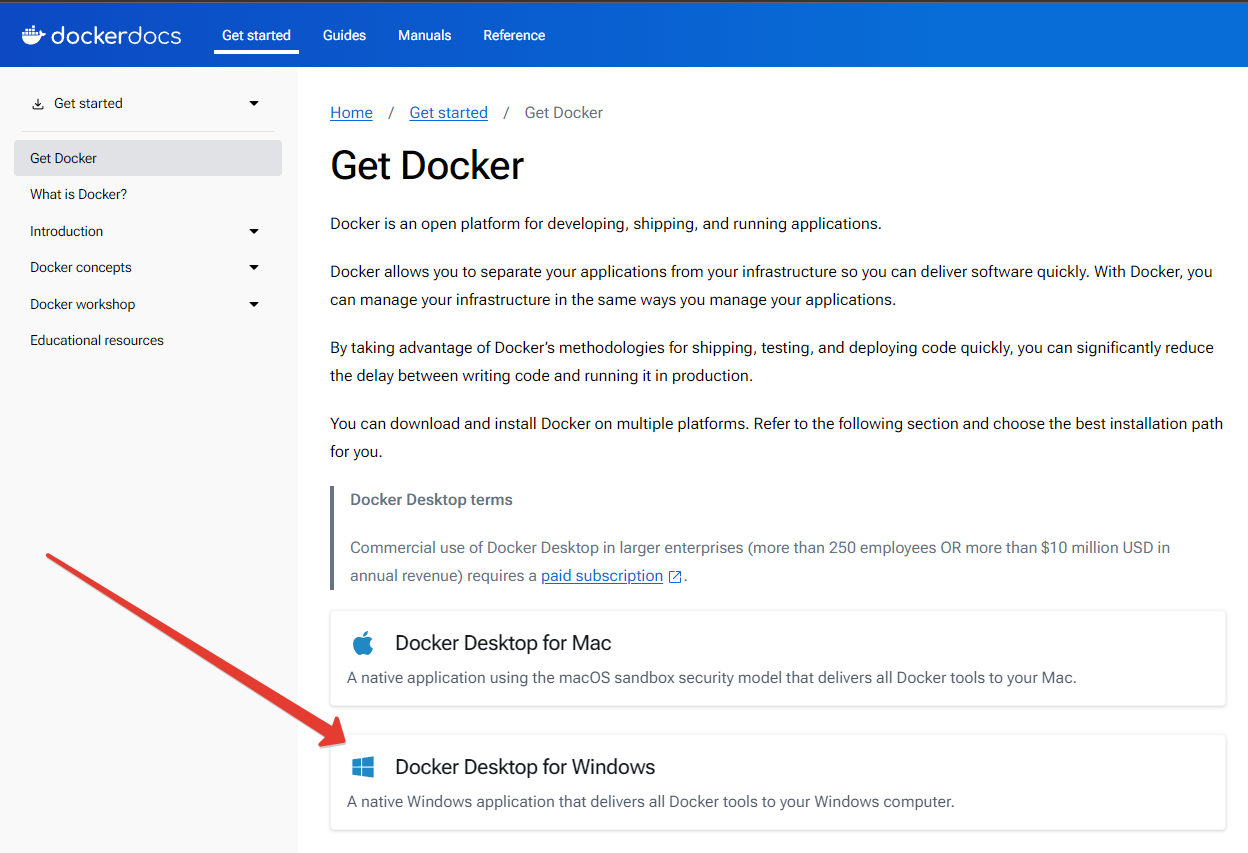


Рисунок 3 – Документация Docker

Далее появляется раздел документации «‎Install Docker Desktop on Windows», здесь происходит выбор подходящего установщика программы Docker Desktop, в зависимости от архитектуры встроенного центрального процессора – *x86*/*x64*/*Arm.*

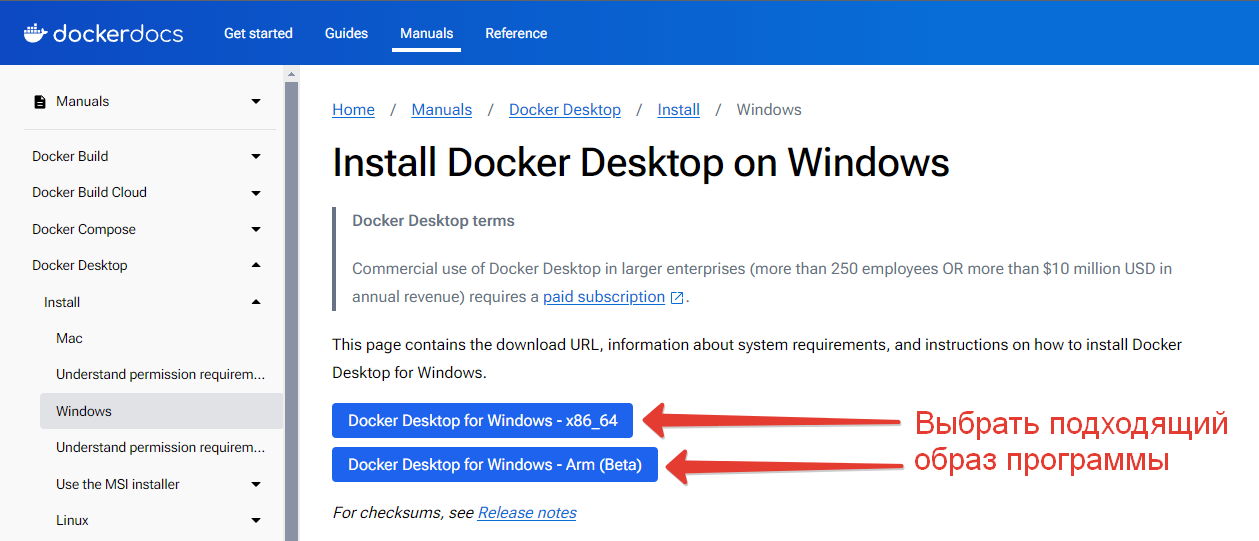


Рисунок 4 – Выбор установщика *Docker Desktop*

После успешного скачивания установщика, запускаем его и ожидаем завершение установки программы.

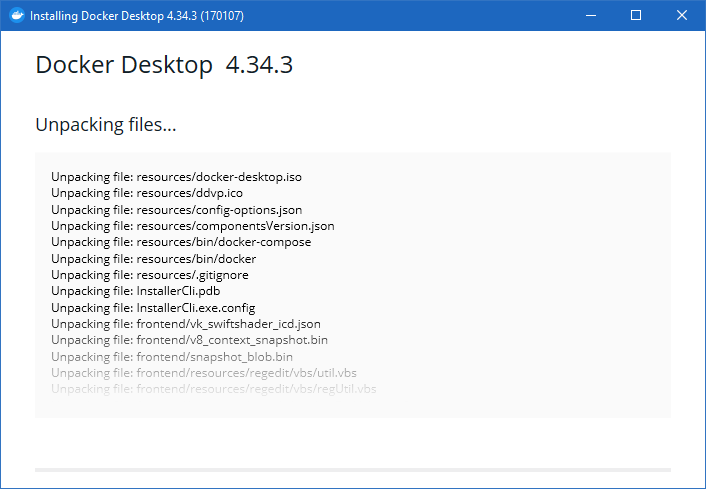


Рисунок 5 – Распаковка файлов программы

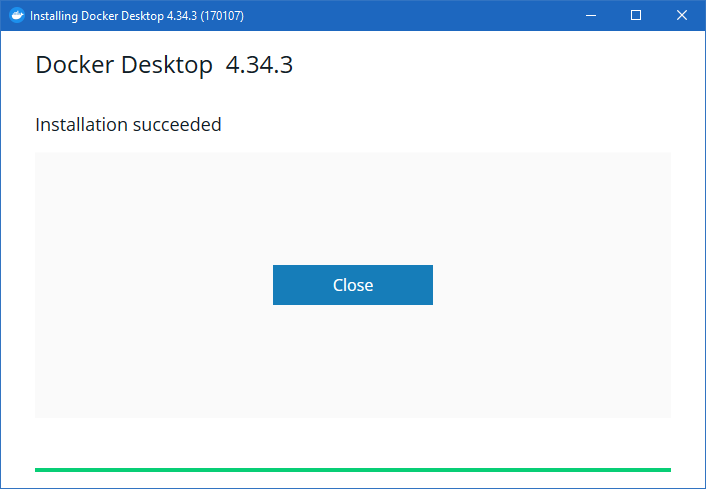


Рисунок 6 – Окно успешной установки

После успешной установки, открываем программу через ярлык на рабочем столе или через меню пуск.

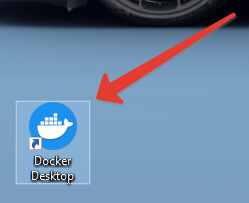


Рисунок 7 – Ярлык на рабочем столе

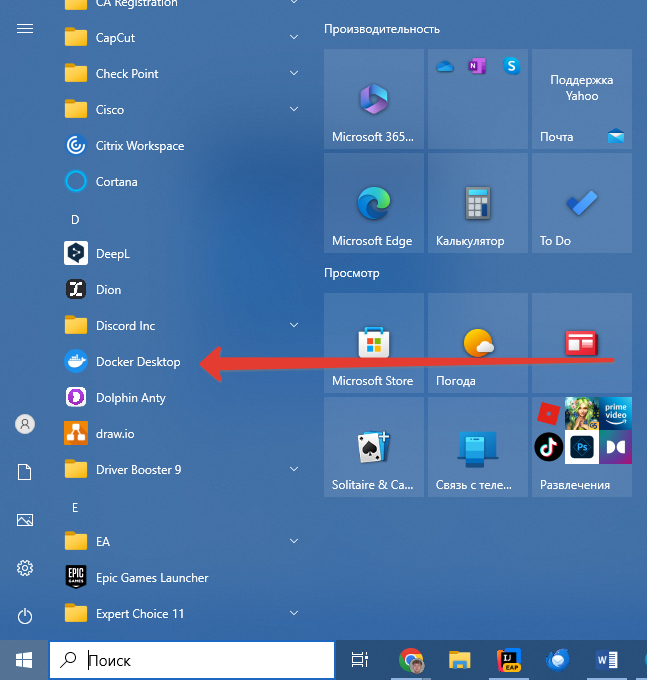


Рисунок 8 – Значок в панели пуск

Открыв *Docker Desktop* проходим локальную авторизацию нажатием на белую клавишу в правом верхнем углу экрана с названием «‎‎*Sign In*»‎.

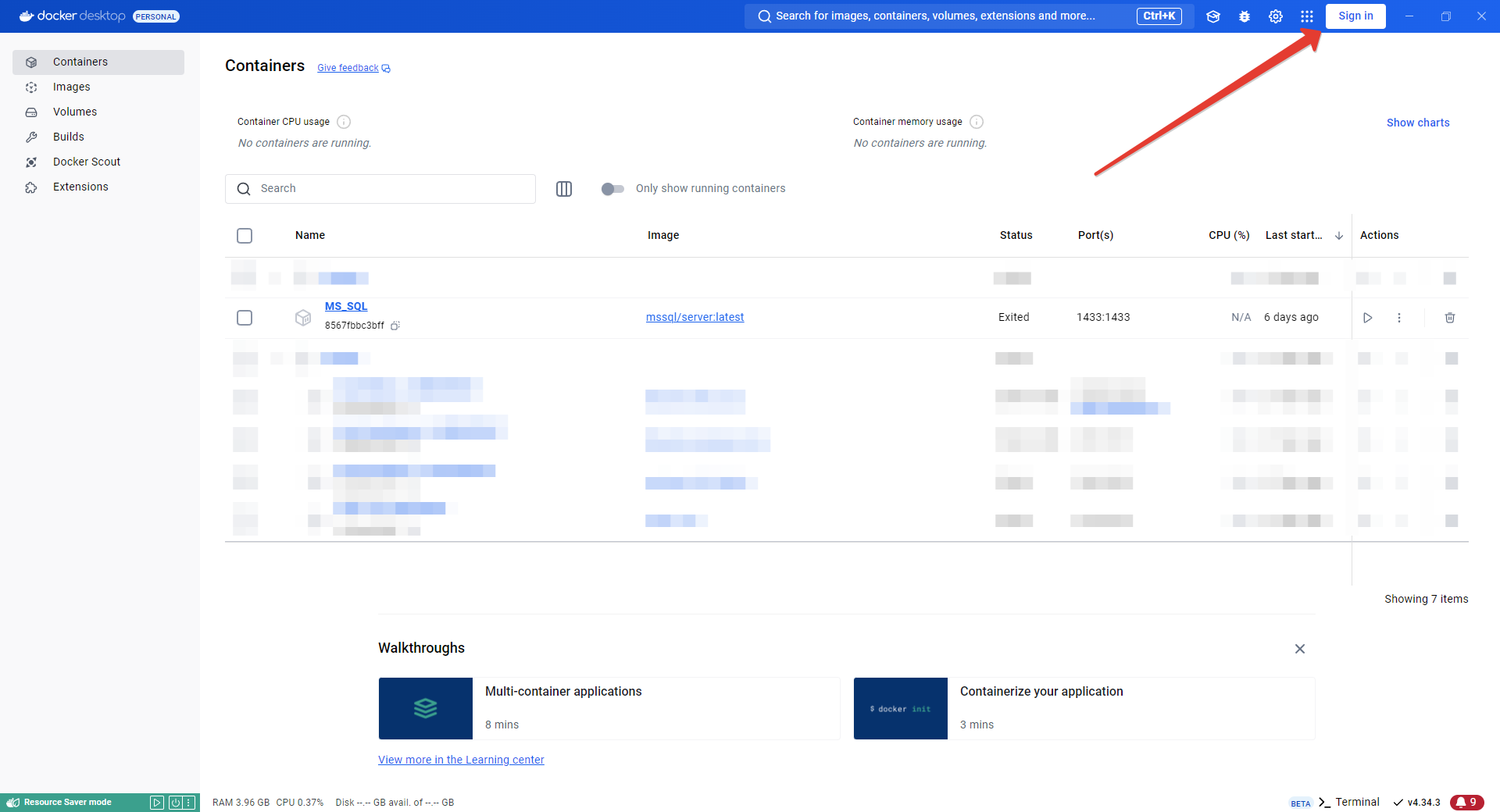


Рисунок 9 – Локальная авторизация в *Docker Desktop*

В открывшейся странице браузера заполняем поля или авторизуемся с использованием *OAuth* через *GitHub* или *Google*.

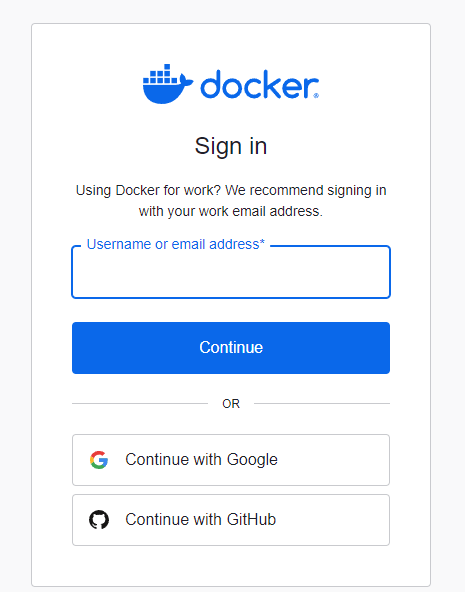


Рисунок 10 – Окно авторизации

После авторизации вас встретит нотификация об успешном входе и появится иконка вашего аккаунта *Docker*.

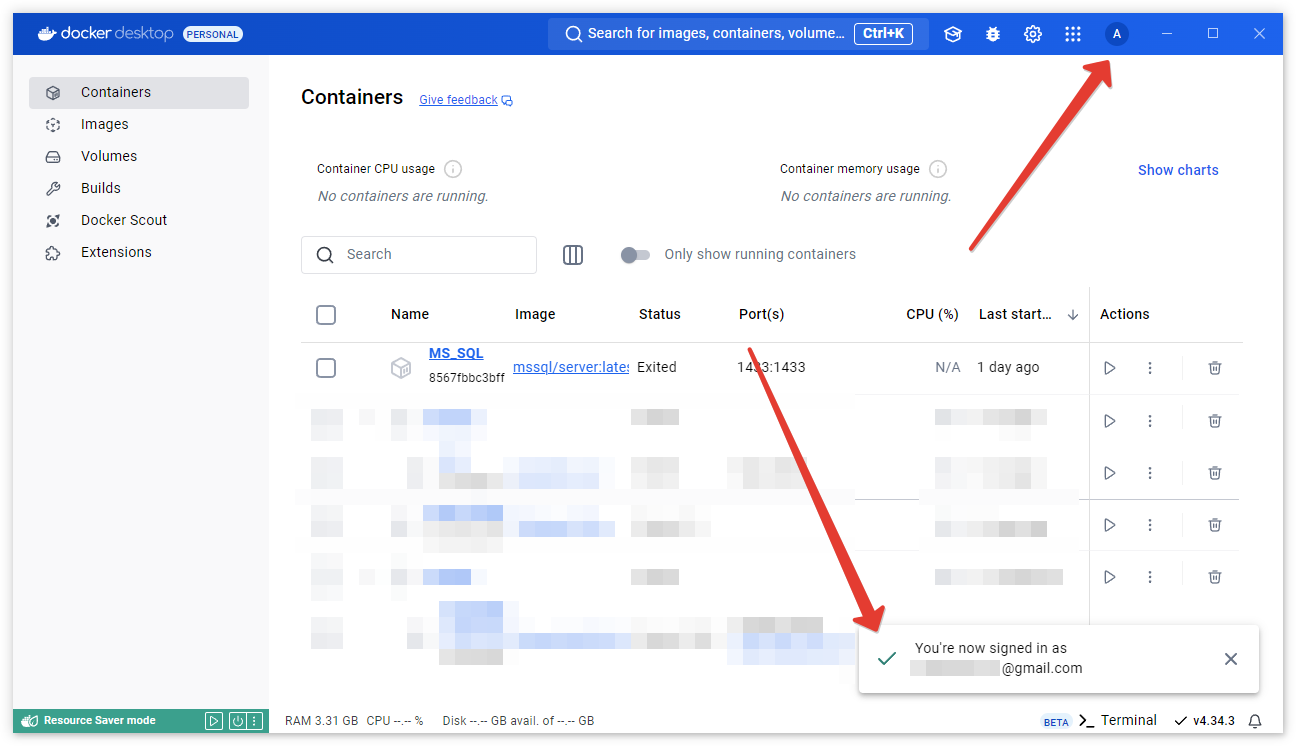


Рисунок 11 – Нотификация об успешном входе

На этом этапе подготовка окружения для развёртывания контейнерной среды завершена, теперь можно скачивать образы и запускать их в виртуальном пространстве.

**ВВЕДЕНИЕ В КОНТЕЙНЕРИЗАЦИЮ**

Начать стоит с дачи определения такому понятию, как контейнеризация. Контейнеризация — это способ упаковки приложения и всех его зависимостей в один образ, который запускается в изолированной среде, не влияющей на основную операционную систему. С помощью контейнера можно отделить приложение от инфраструктуры: не важно, в каком окружении оно будет работать, есть ли там нужные зависимости и настройки, — разработчикам достаточно создать программу и упаковать все зависимости и настройки в единый образ. Затем ее можно разворачивать и запускать на других системах. Это ускоряет процесс разработки, сокращает промежуток между написанием кода и его выкладкой.

Контейнеры и виртуальные машины (ВМ) — это методы изоляции приложений и окружений. *Docker* использует виртуализацию на уровне операционной системы, позволяя запускать приложения в изолированных контейнерах, которые делят ядро ОС с хост-системой. ВМ, напротив, создают полные виртуальные компьютеры с отдельными ОС и ресурсами.

Это означает, что *Docker*-контейнеры более легкие, потому что они не требуют дублирования всей ОС, как ВМ. Контейнеры также быстрее запускаются и имеют меньший объем, что делает их более эффективными в ресурсоемких средах. Однако ВМ предоставляют большую изоляцию, так как каждая ВМ имеет свою собственную ОС и не зависит от хост-системы.

Таким образом, *Docker* контейнеры предоставляют легковесное и быстрое решение для изоляции приложений, в то время как виртуальные машины обеспечивают более глубокую изоляцию за счет полной виртуализации.

**ЗАПУСК *LINUX* КОНТЕЙНЕРА С *MS SQL SERVER***

Контейнеры в системе *Docker* запускаются на основе образов, образ можно собрать собственноручно, описав алгоритм построения контейнера в специальном *Dockerfile*, но для обеспечения нашей потребности воспользуемся готовым образом контейнера, взятым с публичного репозитория образов *Docker Hub.*

Для запуска *Microsoft SQL Server* внутри *Linux* контейнера перейдём на [страницу верифицированного образа](https://hub.docker.com/r/microsoft/mssql-server) от компании *Microsoft*, размещённую на *Docker Hub*.

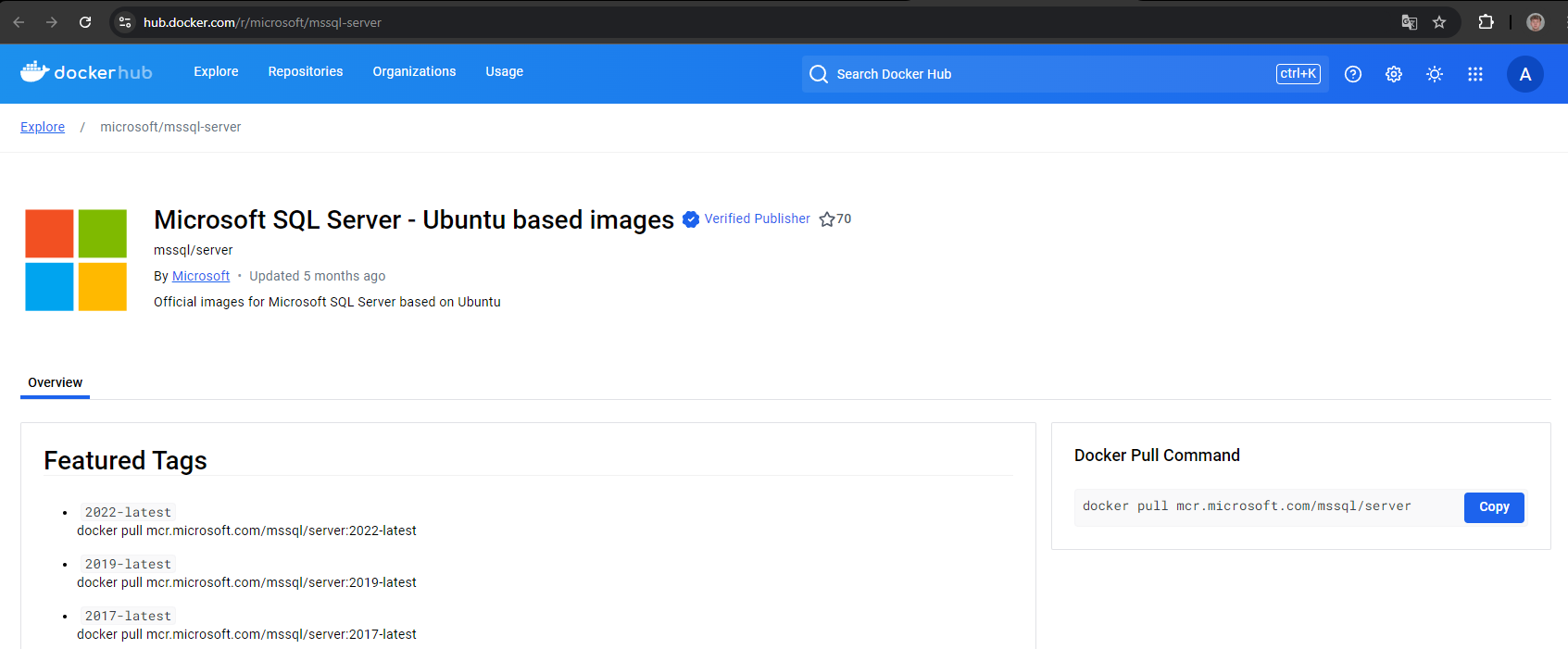


Рисунок 12 – Образ *MS SQL Server* на *Docker Hub*

С этого сайта копируем команду, которая располагается в правой части экрана рядом с синей кнопкой «‎*Copy*».

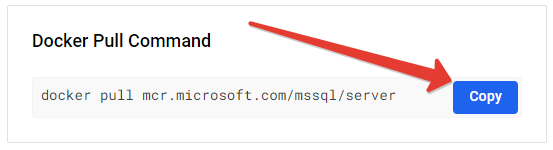


Рисунок 13 – Копирование команды

Далее откроем в приложении Docker Desktop терминал, который находится в правом нижнем углу программы.

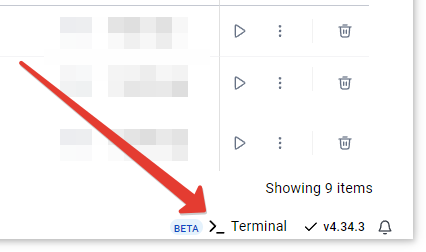


Рисунок 14 – Использование встроенного терминала

Далее в открывшееся окно вставляем скопированную команду с открытого репозитория *Docker Hub* и ожидаем окончания загрузки образа контейнера.

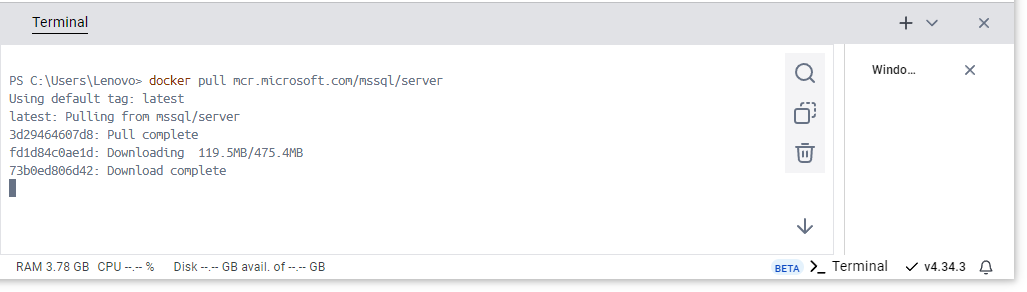


Рисунок 14 – Загрузка образа контейнера *MS SQL Server*

После успешной загрузки, скачанный образ отобразится в списке, который в свою очередь находится во вкладке с названием «‎*Images*».

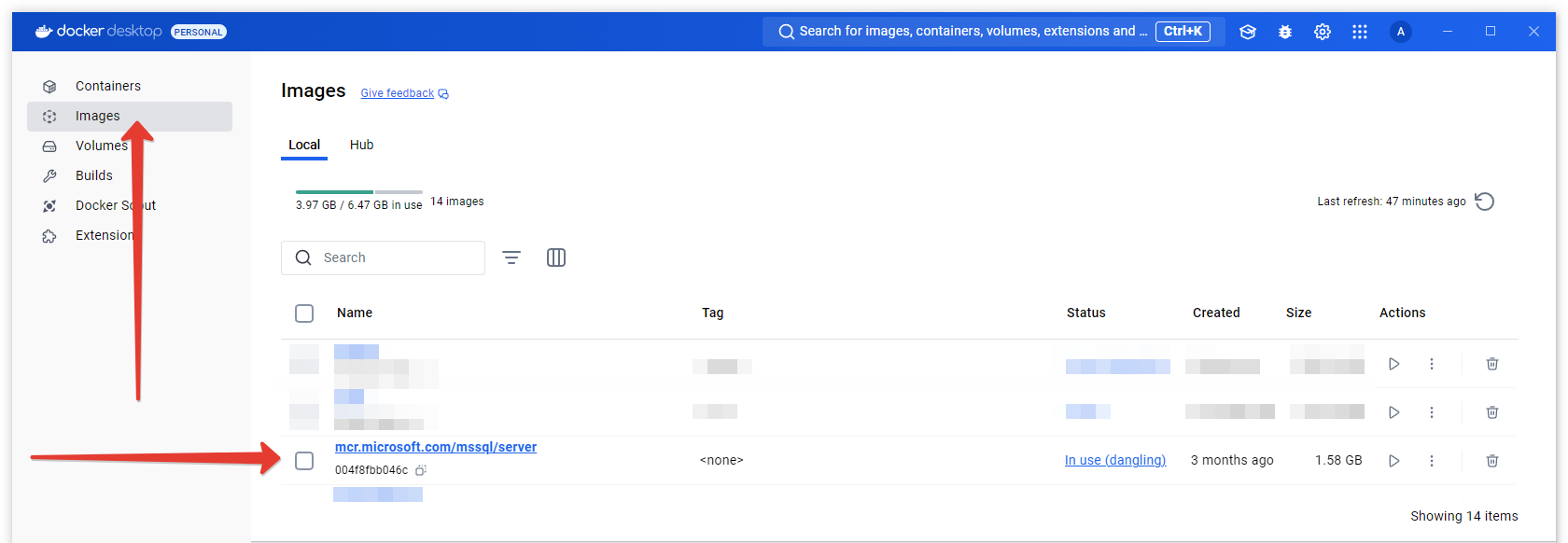


Рисунок 15 – Проверка наличия образа

Далее возвращаемся на сайт *Docker Hub* и ознакамливаемся с документацией контейнера. В ней есть раздел «‎*How to use this image*», в этом разделе указаны варианты команд для запуска контейнров MS SQL Server разных версий и с разным набором параметров. Но в нашем случае нужно ввести в консоль следующую команду.

Фрагмент кода:

*docker run -e "ACCEPT\_EULA=Y" -e "MSSQL\_SA\_PASSWORD=12345678aA%" -e "MSSQL\_PID=Developer" -p 1433:1433 --name MS\_SQL\_SERVER --hostname MS\_SQL\_SERVER -d mcr.microsoft.com/mssql/server:2022-latest*

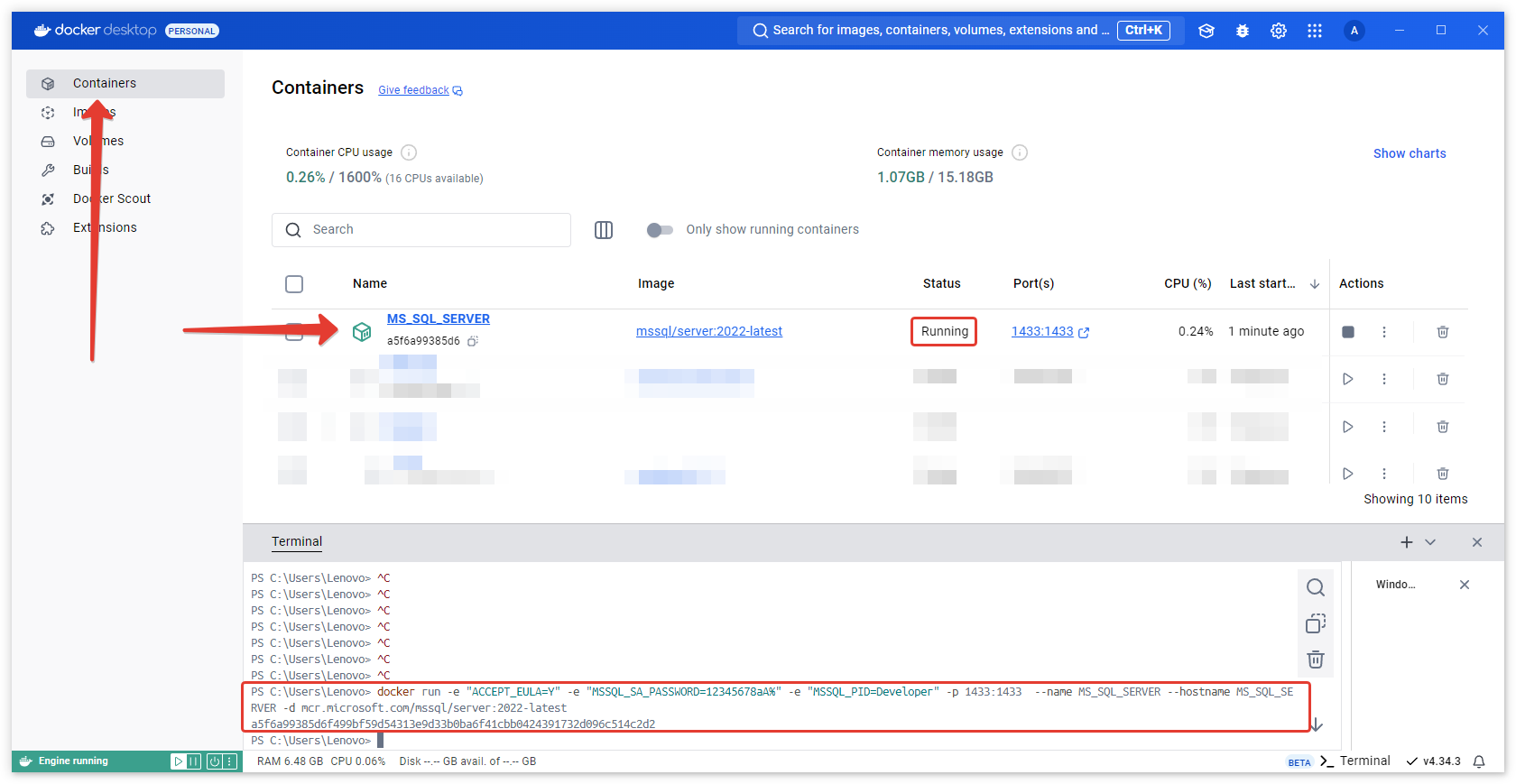


Рисунок 16 – Развёртывание контейнера с *MS SQL Server*

После выполнения указанной команды, во вкладке «‎*Containers*» появится контейнер в статусе «‎*Running*». Это будет значит, что развёртывание *MS SQL Server* в контейнере на базе *Linux* прошло успешно и теперь можно выполнять подключение к базе данных.

**ПОДКЛЮЧЕНИЕ К БД ЧЕРЕЗ *SQL MANAGEMENT STUDIO***

Для подключения к *MS SQL Server* открываем утилиту для администрирования баз данных – *Microsoft SQL Server Management Studio*.

Перед пользователем открывается окно «‎Соединение с сервером», в котором нужно выбрать тип сервера «‎Ядро СУБД», в поле «‎Имя сервера» указать «‎*localhost*», далее установить значение селектора «‎Проверка подлинности *SQL Server*», в качестве имени для входа установить «‎*sa*», а пароль указать тот, который был использован в команде из предыдущего раздела, если вы ничего не меняли, то по умолчанию будет следующее значение – «‎*12345678aA%*».

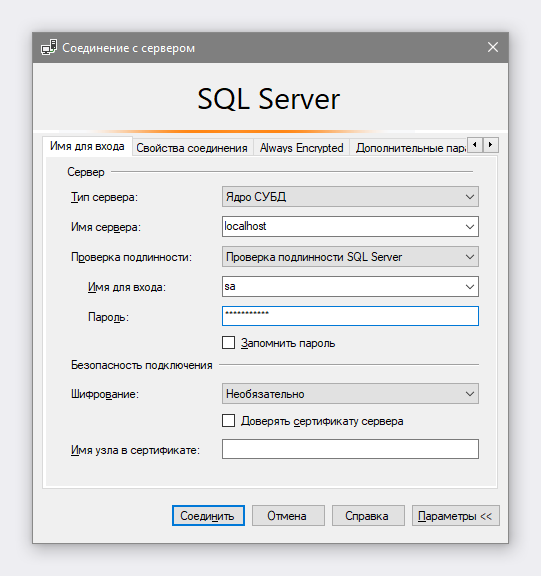


Рисунок 17 – Окно «‎Соединение с сервером».

На этом работа по подключению к MS SQL Server на базе Linux контейнера через *Microsoft SQL Server Management Studio* является успешно завершённой.

**РАБОТА С ФАЙЛОВОЙ СИСТЕМОЙ КОНТЕЙНЕРА**

В рамках выполнения лабораторных работ нередко случается такая ситуация, что требуется поместить какие-то файлы в хранилище сервера БД, далее будет описан процесс переноса файлов с *Windows* в контейнер *Linux* для работы с файлами через утилиту для администрирования БД.

Для того, чтобы поместить локальный файл в контейнер, исполним следующую команду, в которой на месте «‎*example.txt*» указывается абсолютный путь к перемещаемому файлу, к примеру: «‎*E:\BSUIR\ AdventureWorksDW2012*»..

Фрагмент кода:

*docker cp example.txt MS\_SQL\_SERVER:/var/opt/mssql*

После того, как загрузка файла завершена, нашей СУБД нужно выдать права на доступ к данному файлу. Для этого нужно подключиться к контейнеру с правами *root*-пользователя и выдать доступ пользователю *mssql* на работу с перемещённым файлом. Описанные действия реализуются при помощи следующего набора команд, которые исполняются поочерёдно, друг за другом.

Фрагмент кода:

*docker exec -it -u root MS\_SQL\_SERVER bash*

*chown mssql:mssql /var/opt/mssql/example.txt*

*chmod 755 /var/opt/mssql/example.txt*

После выполнения предоставленных команд, через *SQL Management Studio* при помощи исполнения *SQL* запросов можно иметь доступ к файлам, расположенным в хранилище *Linux* контейнера.

Стоит учитывать при формировании *SQL* запросов, что *Linux* имеет отличное от *Windows* строение путей, поэтому в большинстве своём все ваши файлы будут располагаться по следующему пути: «/‎*var/opt/mssql/*».

**ЗАКЛЮЧЕНИЕ**

В данной инструкции была подробно рассмотрена установка среды развертывания *Microsoft SQL Server* на базе *Linux* контейнера в *Docker Desktop*, что позволяет эффективно использовать возможности контейнеризации для выполнения лабораторных работ по дисциплине – системы управления базами данных.

Инструкция начинается с описания процесса установки *Docker Desktop*, который предоставляет удобный интерфейс для управления контейнерами и образами, а также необходимые инструменты для работы с контейнерами. Далее были описаны основы контейнеризации, которая является ключевым понятием в современном подходе к разработке и развертыванию приложений.

Запуск *Linux* контейнера с *Microsoft SQL Server* был описан чётко в качестве итеративных действий, что позволяет студентам быстро и без лишних затруднений настроить рабочую среду. Подключение к базе данных осуществлялось через SQL *Server Management Studio* (*SSMS*) и обеспечило возможность работы с данными в привычной графической среде, что значительно упрощает взаимодействие с базой данных.

Также была рассмотрена работа с файловой системой контейнера, что позволяет более гибко управлять данными и конфигурацией базы данных. Эта часть инструкции подчеркивает важность понимания внутренней структуры контейнера на базе операционной системы Linux и методов работы с файловыми ресурсами.

Данная инструкция предоставляет все необходимые шаги и рекомендации для успешного развертывания и работы с *Microsoft SQL Server* в *Docker*, что делает её полезным ресурсом для студентов и специалистов, стремящихся освоить современные технологии контейнеризации и управления базами данных.