

Развёртывание контейнера с помощью Docker Desktop в операционной системе Windows

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается способ создания Docker контейнера в операционной системе Windows. В качестве инструмента на стороне клиента используется Docker Desktop с графическим пользовательским интерфейсом. Результатом исследования станет описание создания контейнера postgresql, начиная с установки Docker заканчивая созданием базы данных.

Ключевые слова: Docker, контейнер, postgresql.

Deploying a container using Docker Desktop on the Windows operating system

Vikhlyayev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses how to create a Docker container in the Windows operating system. Docker Desktop with a graphical user interface is used as a client-side tool. The result of the study will be a description of the creation of a postgresql container, starting with the installation of Docker and ending with the creation of a database.

Keywords: Docker, container, postgresql.

1 Введение

1.1 Актуальность

Растущая популярность контейнеризации как подхода к разработке, тестированию и развёртыванию приложений делает актуальным использование Docker Desktop в операционной системе Windows как самой популярной операционной системе для пользователей. Современные технологии разработки требуют решений, обеспечивающих гибкость, масштабируемость и совместимость между различными средами. Docker предоставляет инструменты для создания контейнеров, которые изолируют приложения и их зависимости, что позволяет избежать проблем, связанных с различиями в конфигурации среды. Это особенно важно для крупных проектов, где команды разработчиков работают в разных операционных системах и на разнородных инфраструктурах.

1.2Обзор исследований

В.Ф.Захарчёнок, А.Н.Бизюк описали процесс контейнеризации и развертывания приложений с помощью Docker и docker-compose [1]. Е.А.Миних исследовали методы и технологии оптимизации Docker [2]. С.С.Бровко, С.В.Поликарпов продемонстрировали, как контейнерная система Docker может стать инструментом в процессе обучения [3]. В.Е.Манойло разработала инфраструктуру приложений на основе Docker и Kubernetes [4]. М.С.Кузнецов, И.А.Андрианов реализовали распределённые вычисления на языке python с использованием технологии Docker [5].

1.3Цель исследования

Цель исследования – описать способ создания контейнера используя графический интерфейс Docker Desktop.

2 Материалы и методы

Для работы на стороне клиента используется программа Docker Desktop, на стороне сервера Docker engine запущенный на платформ WSL2.

3 Результаты и обсуждения

При установке Docker Desktop на Windows, можно выбрать один из двух бэкендов для работы Docker engine [6].

WSL2 (Windows Subsystem for Linux) – это интеграция Linux в Windows, которая позволяет запускать Linux-ядро внутри Windows.

Hyper-V – это гипервизор от Microsoft, который запускает полноценные виртуальные машины с изолированными системами.

Более современные версии Docker предпочитают WSL2 для интеграции. В таблице 1 представлено сравнение функциональности двух описанных систем.

Таблица 1. Сравнение функциональности бэкенда Docker для Windows

Функция	WSL 2	Hyper-V
Доступность	Windows 10/11 Home и выше	Windows 10/11 Pro, Enterprise, Education
Производительность	Быстрее (особенно файловые операции)	Медленнее
Требовательность к ресурсам	Использует меньше ресурсов (CPU, RAM), так как работает ближе к хостовой ОС.	Требует больше CPU и RAM, так как запускает полноценные виртуальные машины
Изоляция ресурсов	Ограниченная изоляция	Полное отделение виртуальные машины от хостовой системы

Стоит отметить, что преимущественно на производстве Docker контейнеры запускают непосредственно на Linux-серверах. На windows

Docker используется для тестирования и обучения, что делает более быструю и легковесную платформу WSL2 предпочтительной для выбора.

При инсталляции Docker Desktop по умолчанию устанавливается на диск «C:\». Для самостоятельного выбора директории установки следует запустить установочный файл, из командной строки применив команду «"Docker Desktop Installer.exe" install». Чтобы изменить директорию расположения файлов Docker дополнительно нужно указать флаг и путь к каталогу «--installation-dir=D:\Program\Docke\Docke». Помимо, этого указывает местоположение для дистрибутивного диска WSL. По умолчанию оно находится директории пользователя. Изменить путь можно добавив флаг «--wsl-default-data-root=D:\Program\Docke\DockeDesktopWSL».

Местоположение для контейнеров указывает флаг «--windows-containers-default-data-root=D:\Program\Docke\Containers» Эти три флага следует указать если недостаточно место на диске « C:\ » или по другой причине.

В Docker образы (images) – это шаблоны для создания контейнеров. Они содержат всё необходимое: операционную систему, приложение и зависимости. Имеется официальная облачная платформа для хранения, распространения и управления Docker-образами называемая Docker Hub [7]. Она позволяет искать и загружать готовые образы, а также публиковать свои. В Docker Hub, образы могут иметь разные метки, которые помогают пользователю быстро оценить их надежность и происхождение.

Образ имеет зеленую метку "Verified Publisher", что означает, что его автор был официально верифицирован Docker. Эта метка присваивается организациям или компаниям, чьи образы прошли проверку и подтверждены как надежные для использования.

Образ с синей меткой "Official Image" указывает на то, что это официальный образ, поддерживаемый командой Docker или его сообществом. Эти образы тщательно проверяются на соответствие стандартам качества и безопасности.

Имеются официальные образы, созданные и поддерживаемые разработчиками продуктов или сообществом Docker. Они проходят строгие проверки качества, что делает их надежным выбором (рис.1).

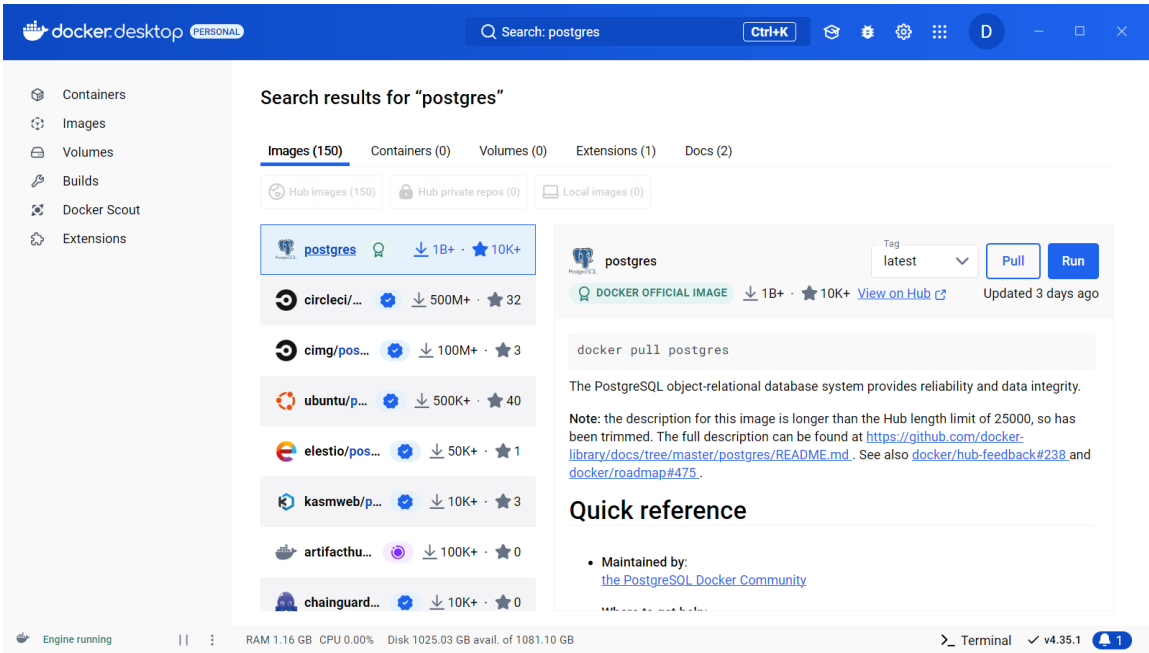


Рис. 1. Docker hub в окне Docker Desktop

Для примера загружен официальный образ postgresql. Чтобы скачать образ нужно нажать «Pull», чтобы скачать и запустить «Run». После скачивания образ появится в списке (рис.2).

☐	Name	Tag	Image ID	Created	Size	Actions
☐	nginx	stable-alpine3.20-perl	b4019c2eb4ff	3 months ago	125.95 MB	
☐	postgres	latest	163763c8afd2	3 days ago	615.36 MB	

Showing 2 items

Рис. 2. Список образов

На основании одного образа можно запустить несколько контейнеров, каждый будет работать в своей Linux системе. Создавать и запускать контейнер лучше отдельно вручную, иначе все параметры будут установлены по умолчанию. Самым необходимым является проброс портов между контейнером и реальной операционной системой. Для PostgreSQL прямо использовать порт 5432, внутри собственной операционной системы Linux уже настроено использование данного порта. Но для внешней системы это нужно сделать вручную. Также конкретно для postgresql при запуске необходимо присвоить переменной «POSTGRES_PASSWORD» пароль суперпользователя, иначе программа завершится с ошибкой и контейнер тут же прекратит работу. В Docker, контейнеры работают только, когда запущен процесс, если процесс завершает работу Docker останавливает контейнер (рис.3).

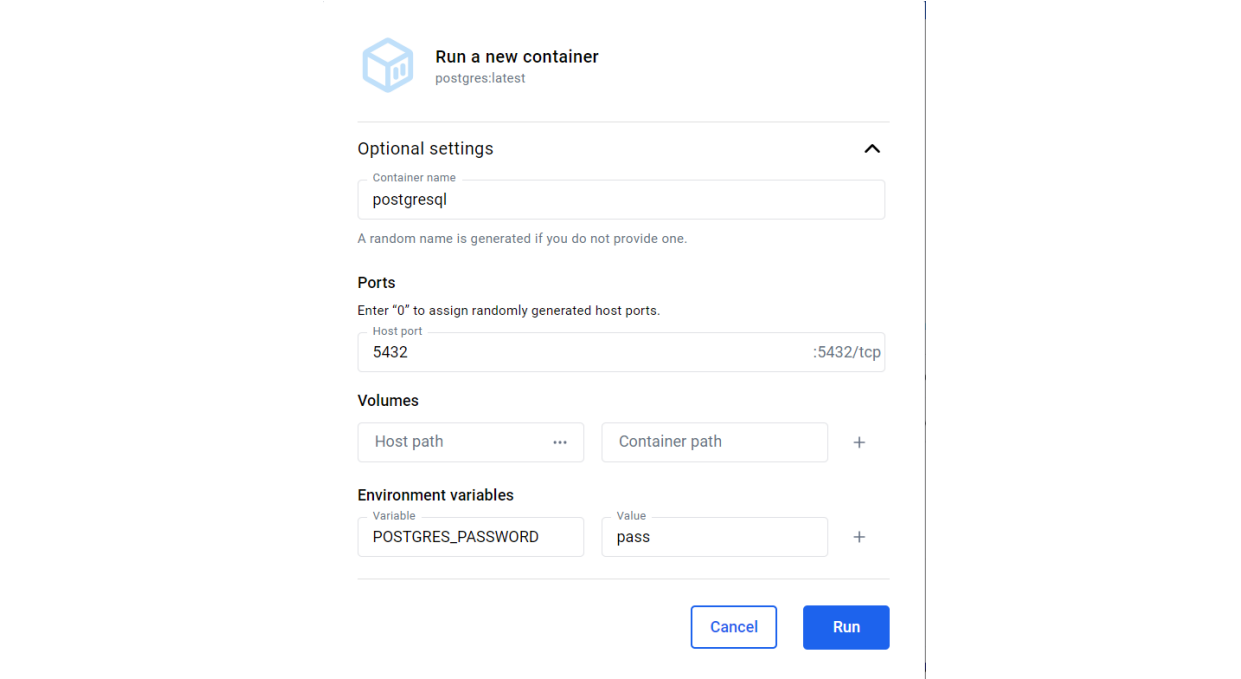


Рис. 3. Создание контейнера

После запуска, в списке контейнеров появится информация о новом контейнере. После перезагрузки, контейнер сохранит все данные, которые были добавлены в него во время прошлой работы (рис.4).

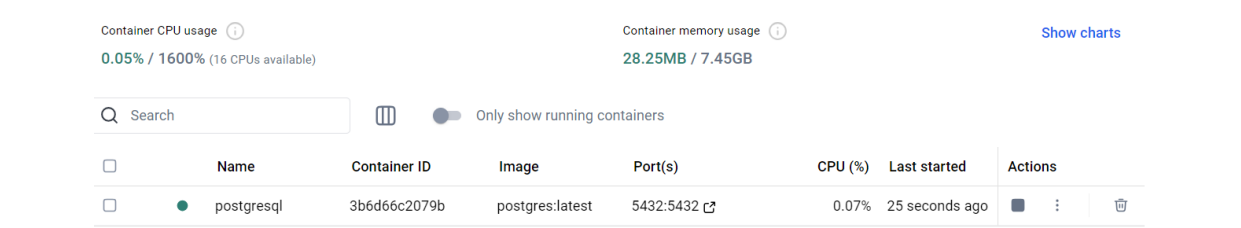


Рис. 4. Список контейнеров

Docker Desktop предоставляет всю необходимую информацию и доступ к управлению контейнера. Зайдя в контейнер можно посмотреть логи и файловую систему, а также подключиться к интерактивному терминалу и управлять системой (рис.5).

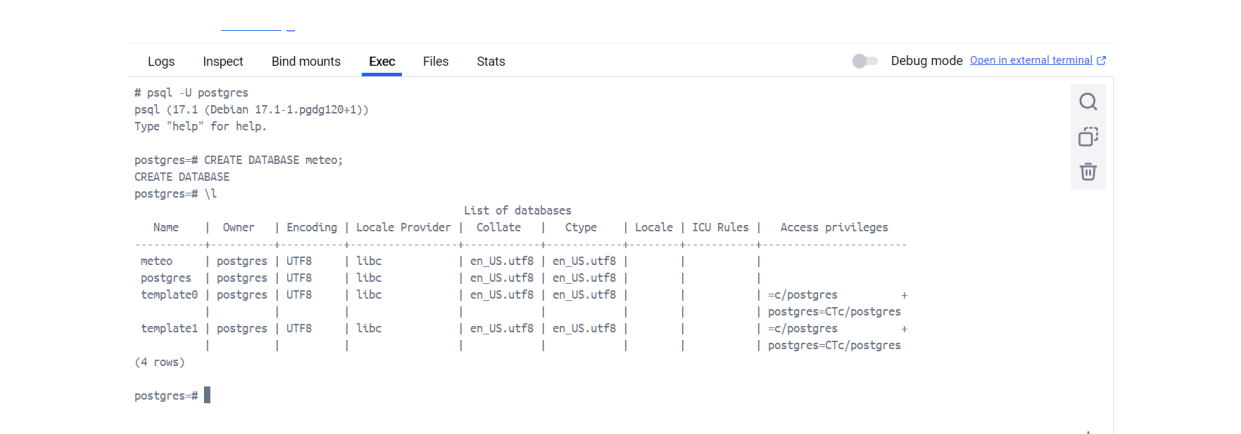


Рис. 5. Интерактивный терминал

В результате исследования был описан способ загрузки Docker Desktop на операционную систему Windows. Описан способ установки директорий хранения программы и контейнеров. Рассмотрен пример скачивания образа с удалённого депозитария и создания на его основе контейнера. Также приведён пример обращения к процессу контейнера.

Библиографический список

1. Захарчёнок В.Ф., Бизюк А.Н. Контейнеризация и развертывание приложений с помощью Docker и docker-compose // В сборнике: Материалы докладов 56-й международной научно-технической конференции преподавателей и студентов. В двух томах. Витебск, 2023. С. 80-82.
2. Миних Е.А. Методы и технологии оптимизации Docker // Аллея науки. 2018. Т. 3. № 4 (20). С. 931-934.
3. Бровко С.С., Поликарпов С.В. Контейнерная система Docker как инструмент процесса обучения // Информационные технологии. Радиоэлектроника. Телекоммуникации. 2015. № 5-1. С. 144-147.
4. Манойло В.Е. Разработка инфраструктуры приложений на основе Docker и Kubernetes // Научно-технические инновации и веб-технологии. 2023. № 1. С. 17-20.
5. Кузнецов М.С., Андрианов И.А. Реализация распределённых вычислений на языке python с использованием технологии Docker // В сборнике: Современные инновации в науке и технике. Сборник научных трудов 10-й Всероссийской научно-технической конференции с международным участием. Курск, 2020. С. 186-189.
6. Документация по установке Docker на операционную систему Windows URL: <https://docs.docker.com/desktop/setup/install/windows-install> (дата обращения: 20.11.2024).
7. Удалённый репозиторий образов Docker URL: <https://hub.docker.com> (дата обращения: 20.11.2024).