Docker auf Windows installieren ohne Docker Desktop

Die DockerCLI und DockerEngine sind Freeware und OpenSource, im Business Bereich muss die Docker Desktop Anwendung erworben werden. Um Docker nutzen zu können, braucht man nicht unbedingt Docker Desktop. Auch wenn Docker Desktop einem bei der Einrichtung vieles einfacher macht. Hier soll eine Schritt für Schritt Anleitung folgen

Alle Schritte nutzen PowerShell oder meine Empfehlung „Windows Terminal Canary“

WSL (Windows-Subsystem für Linux)

Über WSL läuft eine Linux distro unter Windows.

hat man noch kein WSL installiert, kann über diesen Befehl eine Distro ausgewählt werden. Standard ist Ubuntu, ich nutze hier debian.
[crayon-67919d1f6b951326658163/]
um WSL zu installieren gibt man folgenden Befehl ein
[crayon-67919d1f6b959851962517/]
Da das Linux System CPU und RAM Resourcen verbraucht, kann man diese einschränken. Dazu wird eine .wslconfig Datei im Userverzeichnis erstellt
[crayon-67919d1f6b95a659759662/]
In Windows Explorer %UserProfile% eingeben und die erstellte .wslconfig optional bearbeiten:
[crayon-67919d1f6b95c855604689/]
jetzt **wsl** eingeben und wir befinden uns in der Kommandozeile

von Linux

Als erstes sollte man die distro aktualisieren
[crayon-67919d1f6b95e261343904/]

Optional: WSL Image verschieben

Es kann Sinnvoll sein, die WSL Images nicht im Standard Verzeichnis zu haben. Daher kann die WSL Image jederzeit umziehen.

1. Name des WSL-Images bestimmen. In meinem Fall ist es „Debian“:
   [crayon-67919d1f6b95f552426507/]
2. als tar exportieren:
   [crayon-67919d1f6b960936020205/]
3. In WSL das image lösen:
   [crayon-67919d1f6b961917866260/]
4. 2 Verzeichnisse erstellen. Mein neues Verzeichnis ist D:\Docker\wsl
   [crayon-67919d1f6b962734408496/]
5. WSL herunterfahren:
   [crayon-67919d1f6b964467850533/]
6. Import starten: wsl –import Debian
   [crayon-67919d1f6b965759291312/]
7. Neues Image starten:
   [crayon-67919d1f6b966610666402/]
8. Optional:
   [crayon-67919d1f6b968482682158/]

Docker installieren und einrichten

Zunächst werden benötigte Repositories für das Generieren von Zertifikaten installiert und dann docker selbst. Wichtig: Die Zeilen hier unten so wie die sind nacheinander, Zeile für

Zeile kopieren. Nicht mehrere Zeilen miteinander in das Terminal einfügen. Aufforderungen mit Y und Enter bestätigen
[crayon-67919d1f6b969434263377/]
[crayon-67919d1f6b96b802881312/]
Docker Daemon starten. Wenn der nicht weiter geht, Terminal schließen und neuen Terminal mit **wsl** starten
[crayon-67919d1f6b96c160559572/]
Docker müsste jetzt installiert sein. (optional) Das testen wir, indem wir ein hello-world container laufen lassen:
[crayon-67919d1f6b96d311412973/]
Damit unser aktueller User nicht immer sudo eingeben muss, können wir ihn zu der Gruppe der Docker Administratoren hinzufügen. Evtl. ist die Gruppe bereits vorhanden, dann Meldung ignorieren.
[crayon-67919d1f6b96e591424652/]

Um sicherzustellen, dass Docker bei jeden Neustart läuft, folgende Befehle ausführen:

[crayon-67919d1f6b96f487649271/]
Damit auch der dockerd nach dem Start läuft, muss man ein script einfügen. systemctl kann man mit WSL leider nicht nutzen.

Wir legen fest, dass wie sudo bei dem aktuellen Benutzer Benutzer gehandhabt werden soll. Nämlich soll für dockerd kein Passwort erfragt werden:
[crayon-67919d1f6b970842583453/]
ganz unten folgendes hinzufügen und Benutzer gegeben falls anpassen:
[crayon-67919d1f6b972461937491/]
Jetzt müssen wir die bash bearbeiten, damit nach reboot das Skript ausgeführt wird
[crayon-67919d1f6b973452846490/]
mit „Bild runter“-Taste kann man bis ganz unten blättern und

dort folgenden Code einfügen. Mit Strg+X, dann Y und Enter wieder raus:
[crayon-67919d1f6b974415024684/]
Shell neustarten:
[crayon-67919d1f6b975364807194/]
WSL beenden:
[crayon-67919d1f6b976934022039/]
und WSL reboot:
[crayon-67919d1f6b977415965492/]
Falls das nicht geht, kann ein Windows Job eingerichtet werden, der wsl automatisch bei Windows Start ausführt
[crayon-67919d1f6b978606754192/]
Unter Aufgabenplanung in Windows diesen Task als „Unabhängig von der Benutzeranmeldung“ konfigurieren. Sonst funktioniert dieser Task nicht

Linux Docker, Windows bekannt machen

Docker läuft nun auf unserer WSL Linux Maschine. Das nützt uns wenig, da wir in PowerShell nicht immer auf die WSL Linux Maschine zugreifen wollen. Wir wollen einfach docker eingeben und damit die docker Instanz auf der Linux Maschine meinen. Daher erstellen wir jetzt die Brücke.

Zuerst müssen wir das in Linux erlauben. Folgender Befehl öffnet einen Editor:
[crayon-67919d1f6b97a229381533/]
dort mit Strg+V folgendes einfügen:
[crayon-67919d1f6b97b952724460/]
Mit Strg+X , dann Y und Enter speichern

Mit **exit** beenden wir die WSL Kommandozeile

Jetzt muss unter Windows eine Umgebungsvariable für diesen Docker Host angelegt werden (Terminal muss als Admin laufen):

[crayon-67919d1f6b97c295553698/]

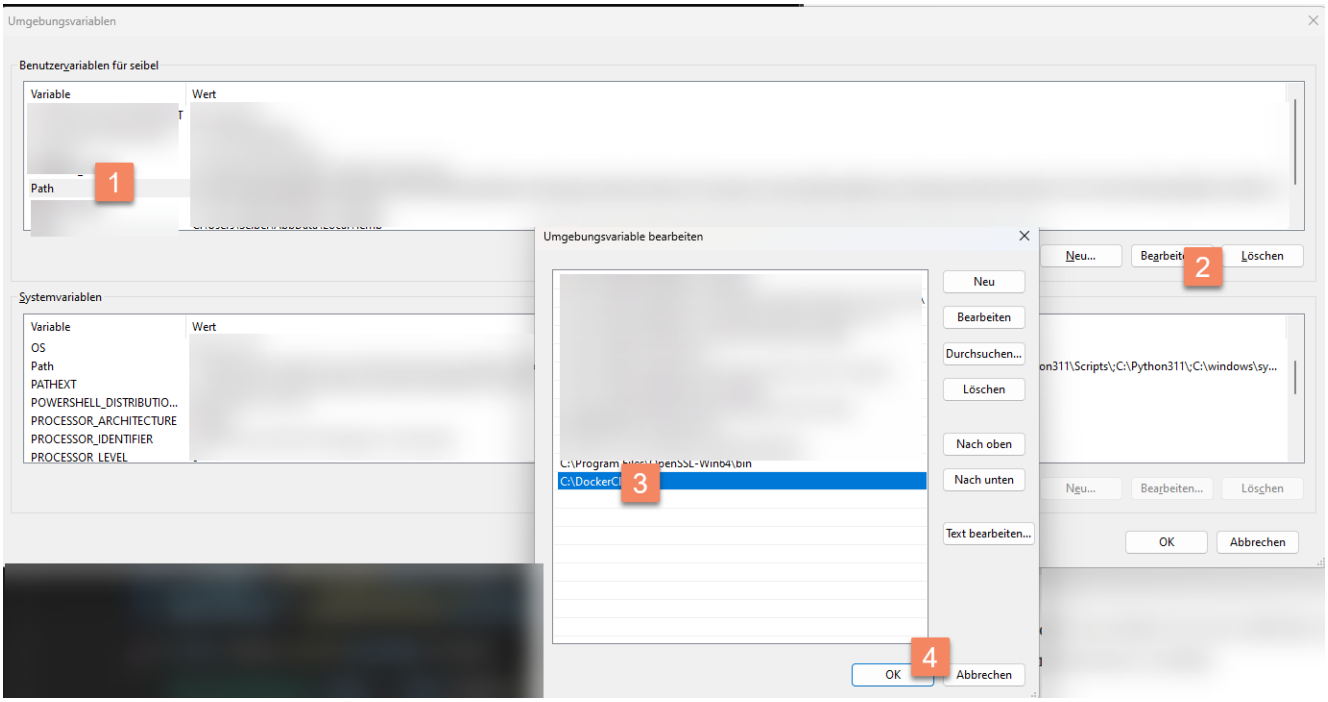
Docker CLI unter Windows installieren

Wenn man bereits choco nutzt, geht das easy mit

[crayon-67919d1f6b97d014221974/]
Ansonsten hier herunterladen:

https://download.docker.com/win/static/stable/x86_64/

und unter einem Verzeichnis z.B. C:\DockerCli ablegen. Nun muss dieses Verzeichnis in den Umgebungsvariablen unter Path eingetragen werden



über folgenden Befehl müsste Docker nun aufrufbar sein
[crayon-67919d1f6b97e623299229/]

Man muss jetzt wissen, dass es nun 3 Ebenen gibt.

Windows -> WSL Linux -> Docker Container.

> *WSL hat für Windows automatisch ein mount in dem Ordner /mnt/ eingerichtet*
>
> *Möchte man nun, dass der Container auf das Windows Verzeichnis C:\Temp bindet, so muss man stattdessen /mnt/c/Temp eingeben*

Referenz:

Docker and WSL2 without Docker Desktop | by Romain Bruyère | Medium

files folders – Change the storage location of a WSL2 – Super User

How to automatically start the Docker daemon on WSL2 – NillsF blog

Portainer installieren mit SSL Zertifikat

Portainer über docker installieren
[crayon-67919d1f6b97f252015226/]
! Funktioniert noch nicht. Das Zertifikat wird vom Browser nicht anerkannt !

Zertifikat auf der Linux WSL erzeugen und das Zertifikat nach D:\Docker kopieren:
[crayon-67919d1f6b980160327073/]

dotNet Docker veröffentlichen

1. die csproj bearbeiten:
   [crayon-67919d1f6c0a3350123891/]

2. Docker Desktop muss vorinstalliert sein und mit wsl2 laufen
   [crayon-67919d1f6c0a7094038244/]
3. Https Zertifikat für Entwicklung erstellen:
   https://learn.microsoft.com/en-us/aspnet/core/security/docker-https?view=aspnetcore-8.0#running-pre-built-container-images-with-https
   [crayon-67919d1f6c0a8488835167/]

Docker

Docker Desktop (AMD64)

Docker Desktop: The #1 Containerization Tool for Developers | Docker

In der Regel reicht auch nur die cli
[crayon-67919d1f6c2a5343221186/]
und Rancher Desktop
[crayon-67919d1f6c2a8713488146/]

Portainer

https://docs.portainer.io/start/install-ce/server/docker/wsl
[crayon-67919d1f6c2a9846848099/]
WebAdmin: https://localhost:9443/

Microsoft Sql Server

microsoft/mssql-server – Docker Image | Docker Hub
[crayon-67919d1f6c2aa518905307/]
Passwort setzen nicht vergessen

Im Container, unter volumes als bind, ein Verzeichnis einrichten, wo die backups liegen und wo die volumes liegen sollen

RabbitMQ

[crayon-67919d1f6c2ab716039958/]
WebAdmin: https://localhost:15672/

Benutzername: guest

Password: guest

Hashicorp/Vault
[crayon-67919d1f6c2ad585427865/]
hashicorp/vault – Docker Image | Docker Hub

Weboberfläche: http://localhost:8200/