

Docker Manual

- [Docker Manual](#)
 - [System Requirements](#)
 - [Windows 10 Pro / Enterprise / Education](#)
 - [Windows 10 Home](#)
 - [MacOS 10.13 or later](#)
 - [Linux](#)
 - [Legacy Sytems \(Windows 7, 8, MacOS 64-Bit\)](#)
 - [Installation](#)
 - [Windows 10 Pro / Enterprise / Education](#)
 - [Windows 10 Home](#)
 - [MacOS 10.13 or later](#)
 - [Linux](#)
 - [Legacy Sytems \(Windows 7, 8, MacOS 64-Bit\)](#)
 - [Docker Basics](#)
 - [docker run](#)
 - [docker ps](#)
 - [docker stop](#)
 - [docker start](#)
 - [docker restart](#)
 - [docker rename](#)
 - [docker exec](#)
 - [Ports unter Docker](#)
 - [Aufräumen](#)
 - [Docker-Compose](#)
 - [Was ist Docker-Compose?](#)
 - [Wichtige Kommandos](#)
 - [Besonderheiten](#)
 - [Postgres](#)
 - [MSSQL](#)

Requirements:

- [Windows 10 Pro / Enterprise / Education:](#)
 - [Build 16299 or later](#)
 - [Hyper-V and Containers Windows feature](#)
- [Windows 10 Home:](#)
 - [Version 2004 or later](#)
 - [WSL2](#)
 - [Linux Kernel Update](#)
- [MacOS 10.13 or later:](#)

- Installer only
- Linux:
 - Depends on distro: Mainly add repo and install with packagemanager (Docker Engine)
- Legacy Systems (Windows 7, 8, MacOS 64-Bit):
 - Virtualization turned on in the BIOS
 - 64-Bit System

Installation:

- Windows 10 Pro / Enterprise / Education:
 - Check that you have **Build 19041** or higher:
 - Drücke Windows + R
 - Trage im Dialog: **winver** ein und klicke auf OK
 - Überprüfen Sie ihre Windows Version. Wenn nötig bitte updaten oder falls dies nicht möglich ist, [Docker Toolbox installieren](#):

About Windows



Windows 10

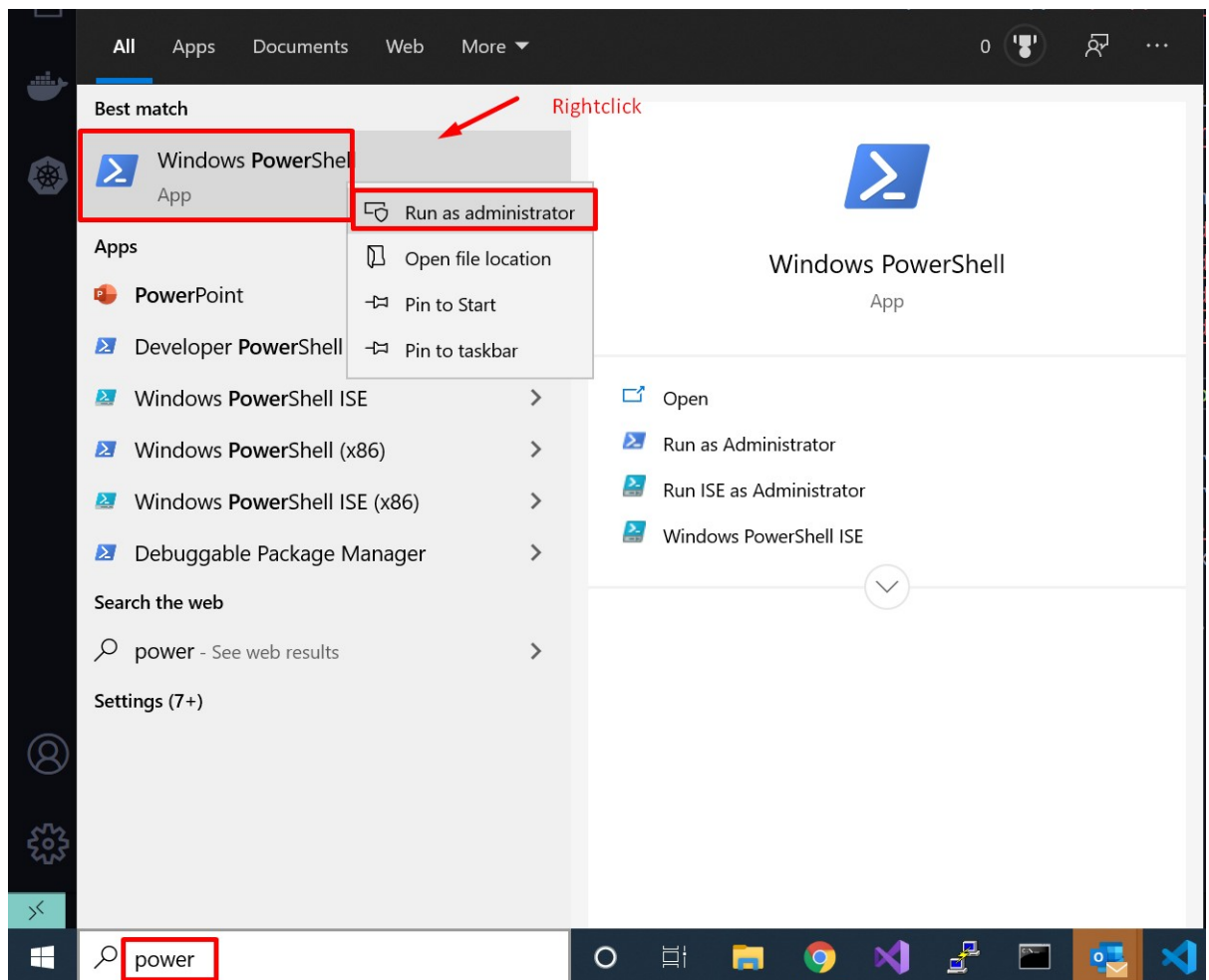
Microsoft Windows

Version 2004 (OS Build 19041.330)

© 2020 Microsoft Corporation. All rights reserved.

The Windows 10 Pro operating system and its user interface are protected by trademark and other pending or existing intellectual property rights in the United States and other countries/regions.

- Eine Powershell **als Administrator** ausführen:



- Folgende Kommandos ausführen:

- `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart`
- `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`

- restart your machine

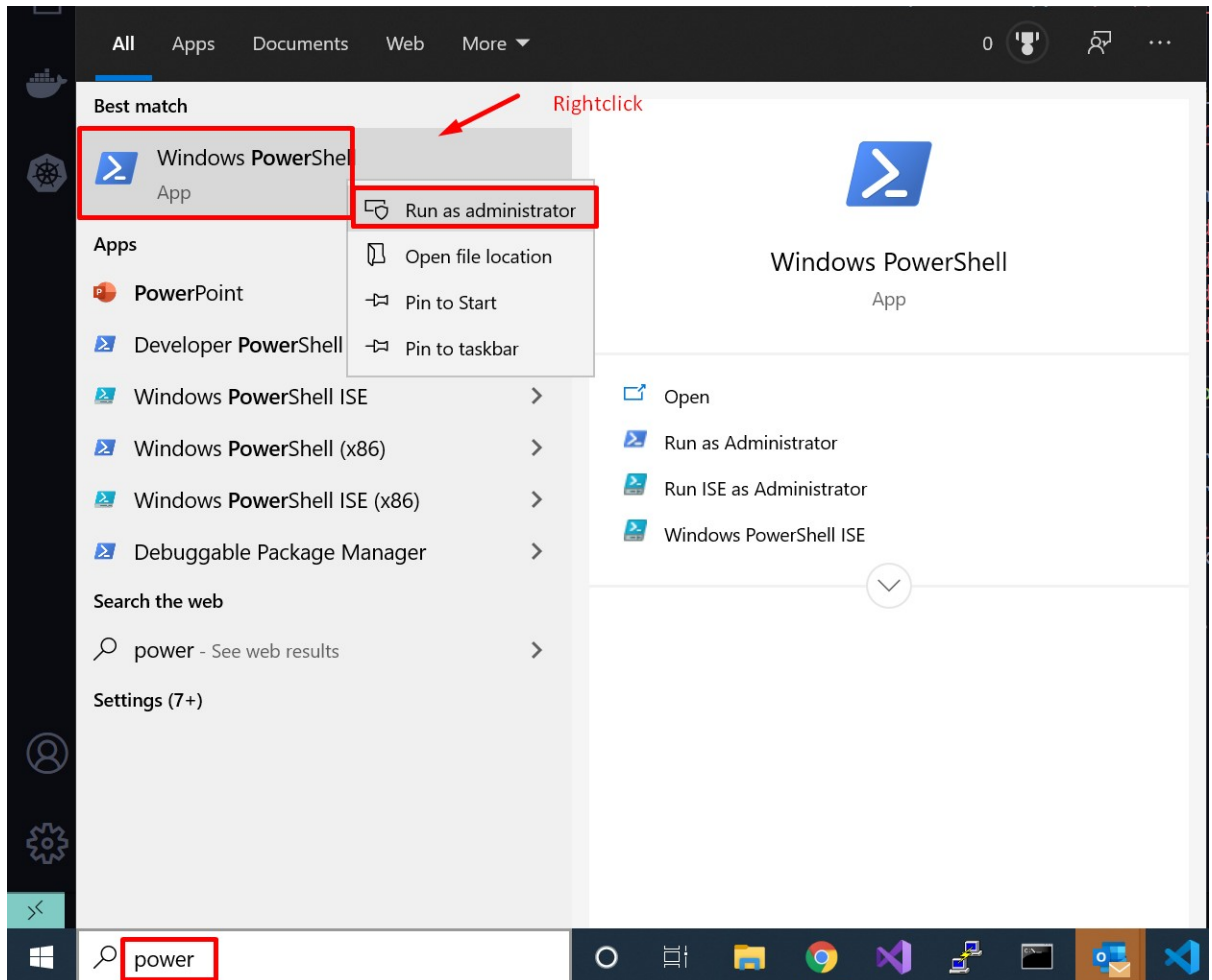
- [Download](#) Docker Desktop von DockerHub
- Führen Sie die Installation aus.
- Falls gefragt wird, stellen Sie sicher, dass `Enable Hyper-V Windows Features` ausgewählt ist.

- Windows 10 Home:

- Stellen Sie sicher, dass sie Windows **Version 2004** or neuer haben:

- Drücken Sie Windows + R.
- Trage im Dialog: `winver` ein und klicke auf OK
- Überprüfen Sie ihre Windows Version. Wenn nötig bitte updaten oder falls dies nicht möglich ist, [Docker Toolbox installieren](#):

- Eine Powershell **als Administrator** ausführen:



- Folgende Kommandos ausführen:

- `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart`
- `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`

- **DEN PC NEUSTARTEN!!!**

- [Downloaden](#) Sie das Linux Kernel Update und installieren Sie es.
- [Downloaden](#) Sie Docker Desktop von DockerHub
- Führen Sie die Installation aus.
- Falls gefragt wird, stellen Sie sicher, dass **Enable Hyper-V Windows Features** ausgewählt ist.

- MacOS 10.13 or later:

- [Downloaden](#) Sie den Installer von Docker Hub
- Führen Sie den Installer aus.
- Nach Der Installation sollte der Docker Wal in der Statusleiste vorhanden sein.



- Linux:

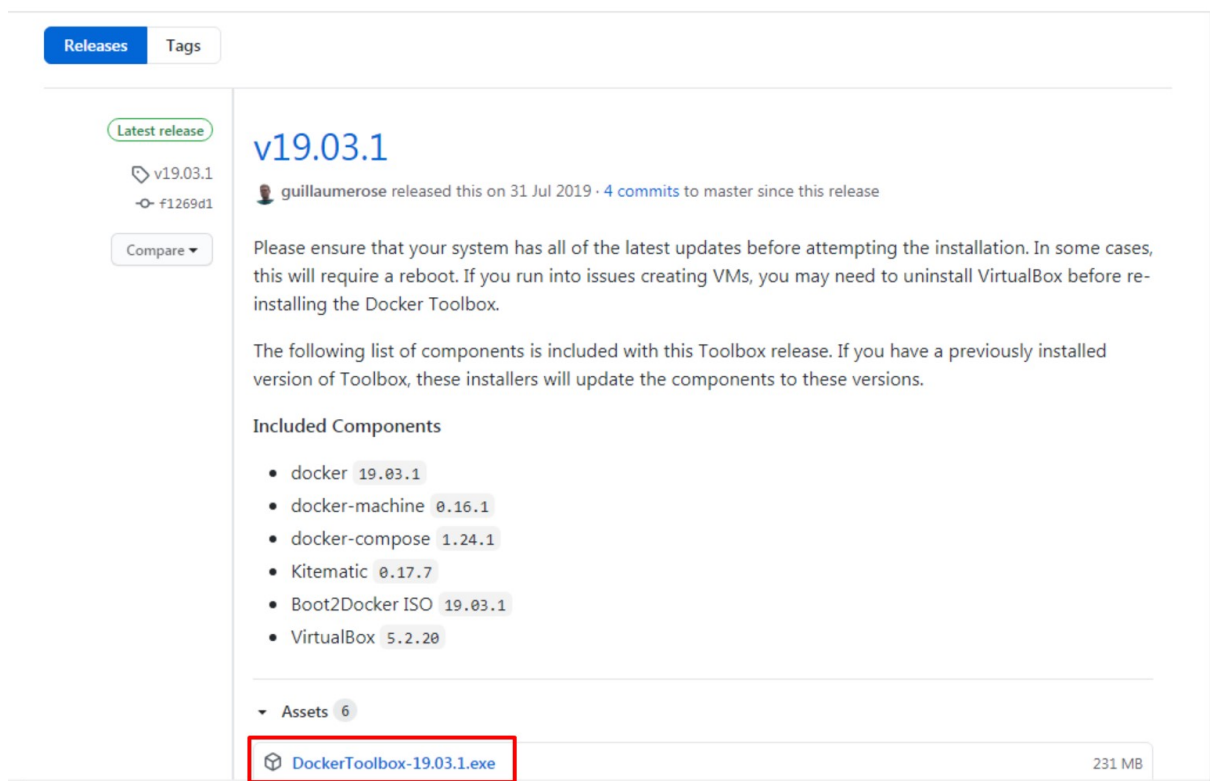
- Schauen Sie bitte in der Installationsanleitung für ihre Distro nach: Für gewöhnlich muss nur die Repo in den Paketmanager eingebunden werden und Docker dann damit installiert werden.

(Docker Engine)

- [CentOS](#)
- [Debian](#)
- [Fedora](#)
- [Ubuntu](#)
- **Additional Step:**
 - Fügen Sie ihren Nutzer die Gruppe **docker** hinzu. Dann müssen Sie die Docker Kommandos nicht als root ausführen.

- Legacy Sytems (Windows 7, 8, MacOS 64-Bit):

- Stellen Sie sicher, dass Virtualisierung in ihrem BIOS aktiviert ist. (Windows)
- Laden Sie sich die neueste Version von [Docker Toolbox](#) herunter.



(exe für Windows, pkg für MacOS)

- Falls Sie bereits VirtualBox installiert haben, stoppen Sie VirtualBox
- Führen Sie die Installation aus.
- Falls Sie VirtualBox bereits installiert haben, wählen Sie VirtualBox in der Liste der zu installierenden Komponenten ab.

Docker Basics

- Lassen Sie uns über eine paar Grundliegende Kommandos gehen.
- **docker run**
 - Um sicher zu gehen, dass die Installation erfolgreich war, können wir das Image **hello-world** ausführen.
Öffnen Sie ein Terminal und führen Sie folgendes Kommando aus:

`docker run <image-name>` hier also `docker run hello-world`

Wenn alles erfolgreich installiert wurde, dann müssten Sie sowas im Terminal sehen:

```
jan@DESKTOP-QNT1UI3:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest:
sha256:d58e752213a51785838f9eed2b7a498ffa1cb3aa7f946dda11af39286c3db9a9
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/
```

Die Ausgabe sagt ihnen folgendes:

- `Unable to find image <image_name>:<image_tag> locally`
 - Das sagt ihnen, dass Sie das Image noch nicht heruntergeladen haben und Docker versucht eine Image mit dem Namen von [Docker Hub](#) herunterzuladen.
- `Hello from Docker!...`
 - Dies ist die Ausgabe des Containers und sagt ihnen, dass Docker erfolgreich installiert wurde.

- `docker ps`

- Mit `docker ps` können Sie sich alle derzeit laufenden Container anzeigen lassen:
 - Wenn Sie jetzt `docker ps` ausführen, dann sehen Sie, dass kein Container läuft. Das hängt damit zusammen, dass der `hello-world` Container direkt nach der ausgabe beendet wird.
 - Lassen sie uns einen `nginx` Container starten.

- Führen sie das Kommando `docker run -d nginx`
 - Dies startet einen `nginx` Container im `detached` Modus
 - Die `-d` Flag ist für den `detached` Modus verantwortlich. Das bedeutet, dass der Container nicht im derzeitigen Terminal läuft, sonder sich im Hintergrund startet.
 - Sie können Docker Container auch mit der `--name <some-name>` Flag benennen. Zum Beispiel: `docker run --name nginx-container -d nginx`
- Wenn Sie nun `docker ps` erneut ausführen, dann sehen Sie, dass der `nginx` container läuft.
- `docker stop <container>`
 - Mit `docker stop <container>` können Sie einen laufenden Container löschen.
 - Ersetzen Sie `<container>` entweder mit dem vollständigen Namen oder einem Teil der ID.
 - Sie müssen nicht die komplette ID ausschreiben. Es reicht ein Teil, solange dieser eindeutig ist.
 - Wenn zum Beispiel die Container ID: `b76c9507a43e` ist, dann können Sie einfach `b76` schreiben, solange es keinen anderen Container gibt, dessen ID mit `b76` beginnt.
 - You have to fully type the container names, but you can use the `tab` key for autocompletion
 - Wenn Sie nun `docker ps` erneut ausführen, dann sehen Sie, dass der `nginx` Container nichtmehr aufgelistet wird.
 - Mit `docker ps -a` können Sie sich alle Container auflisten lassen. Dies beinhaltet auch gestoppte Container.
- `docker start <container>`
 - Dies ist das Gegenteil des Stop Kommandos und startet einen gestoppten Container wieder.
- `docker restart <container>`
 - Mit diesem Kommando kann man einen Container neustarten.
- `docker rename <container> <new-name>`
 - Mit diesem Kommando kann man einen Container umbenennen
- `docker exec <container> <command> <args>`
 - Dieses Kommando erlaubt es `<command>` mit den Argumenten `<args>` innerhalb des Containers auszuführen. Dies is zb. sinnvoll, wenn man auf das Terminal eines Containers zugreifen will:
 - `docker exec -it <container> bash`
 - Dies führt im Container `<container>` das Kommando `bash` im interactive Modus aus.
 - `-it` Diese Flagge sagt Docker, dass das Kommando im interactive Modus ausgeführt werden soll. Dies funktioniert auch mit dem `docker run` Kommando.
 - Mit `exit` können Sie das Terminal des Containers wieder beenden.
- Ports unter Docker

- In Docker sind Container von dem Host System isoliert. Sie laufen in ihrem eigenen Netzwerk. Aus diesem Grund kann man ohne weiteres nicht auf die Ports eines Containers zugreifen. Man kann aber Ports eines Containers auf die Ports des Host-System Mappen
 - Docker Container **EXPOSE** Ports die dann auf die Ports des Host-Systems gemapped werden können
 - Wir können mit der Flag **-p <local_port>:<container_port>** in Kombination mit dem **docker run** Kommando ausführen.
 - Lassen Sie uns den Port **80** des **nginx** container auf den Port **81** unseres Systems mappen. Falls der **nginx** Container noch läuft, beenden Sie diesen.
 - Nun starten wir ihn wieder, aber mappen die Ports wie oben beschrieben:
 - **docker run -d -p 81:80 nginx**
 - Lassen Sie uns einmal über das Kommando gehen:
 - Wir starten einen neuen Container mit dem **nginx** Image
 - Mit der **-d** Flagge sagen wir, dass wir den Container im Hintergrund (detached) laufen lassen wollen.
 - Wir wollen den Container Port **80** als Port **81** auf unserem Host-System erreichbar machen.
 - Wenn wir nun **docker ps** erneut ausführen, sehen wir, dass der Port **81** des Host-Systems, auf Port **80** des Containers gemapped wurde.
 - Öffnen Sie einen Browser und besuchen Sie **http://localhost:81**.
 - Sie sollten die **Default nginx** Seite sehen.
- Alles wieder aufräumen
 - Lassen Sie sich zunächst alle laufenden Container anzeigen:
 - Dies machen Sie mit **docker ps**
 - Nun stoppen Sie alle laufenden Container mit **docker stop**, wie oben beschrieben.
 - Überprüfen Sie mit **docker ps**, dass keine Container mehr laufen. Die Liste sollte leer sein, da das Kommando ja nur laufende Container anzeigt.
 - Nun lassen Sie sich die gestoppten Container mit **docker ps -a** anzeigen.
 - Nun können Sie mit **docker rm <container>** jeden Container löschen.
 - Dieses Kommando ist dem **docker stop** sehr ähnlich. Der Unterschied zum **docker stop** Kommando ist jedoch der, dass **stop** die Container nicht löscht, sondern nur stoppt. Er kann also wieder erneut gestartet werden. **rm** (ReMove) hingegen löscht den Container (und damit gespeicherte Daten)
 - @Todo: Possibility to link commands? eg: Delete all containers at once

Docker-Compose

- Was ist Docker-Compose?

Docker-Compose ist ein tool, dass es erlaubt, mehrer Docker Container zu einer Applikation zu bündeln und diese dann zu starten. Die Konfiguration findet über die **docker-compose.yml** statt. Diese enthält Anweisungen, wie Docker-Compose die Container zur Verfügung stellen soll.

Darin enthalten sind Dinge wie:

- Welche Images
- In welcher Reihenfolge soll es gestartet werden

- Networking
- Volumes...
- Wichtige Kommandos:
 - `docker-compose up`
 - Mit `docker-compose up` kann eine multi-container Applikation gestartet werden. Das Tool guckt im derzeitigen Verzeichnis nach der `docker-compose.yml`.
 - `-d` Flag: Startet die Container im `detached` / Hintergrund Modus
 - `docker-compose down`
 - Mit `docker-compose down` kann eine multi-container Applikation gestoppt werden. Das Tool guckt wie bei `docker-compose up` im derzeitigen Verzeichnis nach einer `docker-compose.yml`
 - `-v` Flag: Gibt an, dass die in der `docker-compose.yml` definierten Volumes mit gelöscht werden sollen.
 - `docker-compose ps`
 - Mit `docker-compose ps` lassen sich die von Docker-Compose erstellten Container auflisten.

Besonderheiten:

- **Postgres:**
 - Das Postgres Image erlaubt es SQL Skripte beim erstellen auszuführen. Dazu müssen dann die `.sql` Dateien in `/docker-entrypoint-initdb.d/` gemountet werden. Alternativ kann auch das Image per DOCKERFILE erweitert werden. [Beispiel für MSSQL](#)
- **MSSQL:**
 - Das gleiche geht auch beim MSSQL Image aber dort ist es leider mit ein wenig Arbeit verbunden [Github Issue](#).
Hierbei wird über das Command Attribut in der docker-compose ein shell skript ausgeführt, das intern `sqlcmd` nutzt um ein sql skript auszuführen.
[Beispiel für MSSQL](#)