# Quick And Dirty

Copy And Paste -Sammlung

- Create Registry
- Install Docker ee
- WSL2
- Private Registry - Push
- Private Registry - Search
- Private Registry - Build
- docker-compose recreate
- docker-compose install

# Create Registry

rem docker run -d -p 5000:5000 --restart=always --name registry -v
D:/DockerRegistry:/var/lib/registry registry:2

docker run -d -p 5000:5000 --restart=always --name registry registry:2

You will need to save the Docker image as a tar file:

docker save -o <path for generated tar file> <image name>
Then copy your image to a new system with regular file transfer tools such as cp, scp or rsync(preferred for big files). After that you will have to load the image into Docker:

docker load -i <path to image tar file>
PS: You may need to sudo all commands.

EDIT: You should add filename (not just directory) with -o, for example:

[Environment]::SetEnvironmentVariable("LCOW_SUPPORTED", "1", "Machine")
docker load -i d:\transfer\registry.tar
docker save -o c:/myfile.tar centos:16

Deaktivieren lässt sich die Container-Unterstützung mit:

[Environment]::SetEnvironmentVariable("LCOW_SUPPORTED", $null, "Machine")

# Install Docker ee

Manuell
http://man.hubwiz.com/docset/Docker.docset/Contents/Resources/Documents/docs.docker.com/install/windows/docker-ee.html

--versionen
https://dockermsft.blob.core.windows.net/dockercontainer/DockerMsftIndex.json

--install
```
# Stop Docker service
Stop-Service docker

# Extract the archive.
Expand-Archive docker-18.09.5.zip -DestinationPath $Env:ProgramFiles -Force

# Clean up the zip file.
Remove-Item -Force docker-18.09.5.zip

# Install Docker. This requires rebooting.
$null = Install-WindowsFeature containers

# Add Docker to the path for the current session.
$env:path += ";$env:ProgramFiles\docker"

# Optionally, modify PATH to persist across sessions.
$newPath = "$env:ProgramFiles\docker;" +
[Environment]::GetEnvironmentVariable("PATH",
[EnvironmentVariableTarget]::Machine)

[Environment]::SetEnvironmentVariable("PATH", $newPath,
[EnvironmentVariableTarget]::Machine)

# Register the Docker daemon as a service.
dockerd --register-service

# Start the Docker service.
Start-Service docker
```

---
Der einfachste Weg, um in Windows Server 2019 die Container-Funktion zu installieren besteht darin, dass der Server über eine Internetverbindung verfügt, und über diesen Weg der Download der notwendigen Komponenten bei Docker erfolgt.

Auch Container-Images oder andere Funktionen lassen sich über das Internet, zum Beispiel mit der PowerShell recht einfach auf Windows Server 2019 installieren. Zunächst sollte der Docker-Microsoft PackageManagement Provider aus der PowerShell-Gallery auf dem Server installiert werden:

mofcomp

Danach wird die aktuelle Docker-Engine installiert:

Install-Package -Name docker -ProviderName DockerMsftProvider

Die Installation muss noch bestätigt werden, danach wird Docker Enterprise auf dem Server integriert. Mit der PowerShell kann Docker in Windows Server 2019 auch aktualisiert werden:

Install-Package -Name Docker -ProviderName DockerMsftProvider -Update -Force

Start-Service Docker

Nach der Installation sollte der Server neu gestartet werden:

Restart-Computer -Force

Die erfolgreiche Installation kann ebenfalls in der PowerShell überprüft werden:

Get-Package -Name Docker -ProviderName DockerMsftProvider

Wenn die Installation erfolgreich durchgeführt wurde, steht in der PowerShell und der Eingabeaufforderung auch der Befehl "docker" zur Verfügung. Mit diesem kann die installierte Version von Docker überprüft werden:

docker --version

In einem weiteren Beitrag beschäftigen wir uns damit, wie Linux- und Windows-Container mit Docker auf Windows Server 2019 betrieben werden können.

---

inux-Container in Windows Server 2019 betreiben

Windows-Container stellen auf Windows Server 2019 kein Problem dar. Wer Linux-Container betreiben will, benötigt das "LinuxKit". Das ist erst ab Windows Server 2016 Build 16278 auf in Windows Server 2019 verfügbar. In Windows Server 2019 kann die Unterstützung von Linux-Containern mit dem folgenden Befehl aktiviert werden:

[Environment]::SetEnvironmentVariable("LCOW_SUPPORTED", "1", "Machine")

Anschliessend muss der Dienst neu gestartet werden:

Restart-Service docker

Deaktivieren lässt sich die Container-Unterstützung mit:

[Environment]::SetEnvironmentVariable("LCOW_SUPPORTED", $null, "Machine")

Linux-Container auf Windows Server 2019 benötigen Hyper-V. Auf dem Server muss also auch Hyper-V installiert sein.

---

daemon.json

```
{
"debug": true,
"experimental": true
}
```

# WSL2

https://code.visualstudio.com/blogs/2020/03/02/docker-in-wsl2

https://docs.microsoft.com/de-de/windows/wsl/install-manual

Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

#curl.exe -L -o ubuntu-1804.appx https://aka.ms/wsl-ubuntu-1804
curl -o ubuntu-2004.appx https://aka.ms/wslubuntu2004
#Invoke-WebRequest https://aka.ms/wslubuntu2004

Rename-Item ubuntu-2004.appx ubuntu-2004.zip
Expand-Archive ubuntu-2004.zip ubuntu2004

cd ubuntu2004
.\ubuntu2004.exe

sudo apt update && sudo apt upgrade
sudo apt install ansible

Add your distro path to the Windows environment PATH using Powershell:

$userenv = [System.Environment]::GetEnvironmentVariable("Path", "User")
[System.Environment]::SetEnvironmentVariable("PATH", $userenv +
"C:\Users\Administrator\ubuntu2004", "User")
This will enable you to launch your distro from any path by typing the .exe launcher. For example using ubuntu2004.exe.

Note that this will require closing and relaunching PowerShell.

https://blog.nillsf.com/index.php/2020/06/29/how-to-automatically-start-the-docker-daemon-on-wsl2/

https://medium.com/faun/docker-running-seamlessly-in-windows-subsystem-linux-6ef8412377aa

# Private Registry - Push

https://docs.docker.com/engine/reference/commandline/push/

## Docker Push

## Beschreibung

## Push endert ein Image oder Repository in eine Registrierung

Nutzung
docker push [OPTIONS] NAME[:TAG]
Erweiterte Beschreibung
Verwenden Sie diese Datei, um Ihre Images für die Docker Hub-Registrierung oder für eine selbst gehostete Datei freizugeben.docker image push

Weitere Informationen zu gültigen Bild- und Tagnamen finden Sie in der Docker-Image-Tag-Referenz.

Das Töten des Prozesses, z. B. durch Drücken während des Laufens in einem Terminal, beendet den Push-Vorgang.docker image pushCTRL-c

Fortschrittsbalken werden während des Docker-Pushs angezeigt, die die unkomprimierte Größe anzeigen. Die tatsächliche Datenmenge, die übertragen wird, wird vor dem Senden komprimiert, sodass die hochgeladene Größe nicht von der Fortschrittsleiste widergespiegelt wird.

Registrierungsanmeldeinformationen werden von docker loginverwaltet.

Gleichzeitige Uploads
Standardmäßig schiebt der Docker-Daemon fünf Ebenen eines Bildes gleichzeitig. Wenn Sie eine Verbindung mit geringer Bandbreite haben, kann dies zu Problemen mit Timeouten führen, und Sie können dies über die Daemon-Option verringern. Weitere Informationen finden Sie in der Daemon-Dokumentation.--max-concurrent-uploads

Verwenden Sie z. B. diesen Befehl, lesen Sie den Abschnitt "Beispiele" weiter unten.

Optionen
Name, Kurzschrift Standard Beschreibung
--all-tags , -a Drücken Sie alle markierten Bilder in das Repository
--disable-content-trust true Überspringen der Bildsignatur
--quiet , -q Unterdrücken ausführlicher Ausgabe
Beispiele
Push en ein neues Image an eine Registrierung
Speichern Sie zuerst das neue Image, indem Sie die Container-ID (mit Docker-Container ls)suchen

und dann an einen neuen Imagenamen übertragen. Beachten Sie, dass nur beim Benennen von Bildern zulässig sind:a-z0-9-_.

$ docker container commit c16378f943fe rhel-httpd:latest
Übertragen Sie das Bild nun mithilfe der Image-ID in die Registrierung. In diesem Beispiel befindet sich die Registrierung auf dem Host nament und lauscht auf Port . Markieren Sie dazu das Bild mit dem Hostnamen oder der IP-Adresse und dem Port der Registrierung:registry-host5000

$ docker image tag rhel-httpd:latest registry-host:5000/myadmin/rhel-httpd:latest

$ docker image push registry-host:5000/myadmin/rhel-httpd:latest
Überprüfen Sie, ob dies funktioniert hat, indem Sie:

$ docker image ls
Sie sollten beides sehen und aufgelistet.rhel-httpdregistry-host:5000/myadmin/rhel-httpd

Drücken Sie alle Tags eines Bildes
Verwenden Sie die Option (oder ), um alle Tags eines lokalen Bildes zu drücken.-a--all-tags

Im folgenden Beispiel werden mehrere Tags für ein Image erstellt und alle diese Tags an Docker Hub übertragen.

$ docker image tag myimage registry-host:5000/myname/myimage:latest
$ docker image tag myimage registry-host:5000/myname/myimage:v1.0.1
$ docker image tag myimage registry-host:5000/myname/myimage:v1.0
$ docker image tag myimage registry-host:5000/myname/myimage:v1
Das Bild ist jetzt unter mehreren Namen getaggt:

$ docker image ls

REPOSITORY TAG IMAGE ID CREATED SIZE
myimage latest 6d5fcfe5ff17 2 hours ago 1.22MB
registry-host:5000/myname/myimage latest 6d5fcfe5ff17 2 hours ago 1.22MB
registry-host:5000/myname/myimage v1 6d5fcfe5ff17 2 hours ago 1.22MB
registry-host:5000/myname/myimage v1.0 6d5fcfe5ff17 2 hours ago 1.22MB
registry-host:5000/myname/myimage v1.0.1 6d5fcfe5ff17 2 hours ago 1.22MB
Beim Drücken mit der Option werden alle Tags des Bildes gedrückt:--all-tagsregistry-host:5000/myname/myimage

$ docker image push --all-tags registry-host:5000/myname/myimage

The push refers to repository [registry-host:5000/myname/myimage]
195be5f8be1d: Pushed
latest: digest:
sha256:edafc0a0fb057813850d1ba44014914ca02d671ae247107ca70c94db686e7de6 size: 4527
195be5f8be1d: Layer already exists
v1: digest: sha256:edafc0a0fb057813850d1ba44014914ca02d671ae247107ca70c94db686e7de6

size: 4527
195be5f8be1d: Layer already exists
v1.0: digest: sha256:edafc0a0fb057813850d1ba44014914ca02d671ae247107ca70c94db686e7de6 size: 4527
195be5f8be1d: Layer already exists
v1.0.1: digest: sha256:edafc0a0fb057813850d1ba44014914ca02d671ae247107ca70c94db686e7de6 size: 4527
Übergeordneter Befehl
Befehl Beschreibung
Docker Der Basisbefehl für die Docker CLI.

# Private Registry - Search

http://registry:5000/v2/_catalog

# Private Registry - Build

https://github.com/StefanScherer/dockerfiles-windows/blob/main/registry/Dockerfile

dockerfile

```
FROM golang as build

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]

ENV DOCKER_BUILDTAGS include_oss include_gcs
ENV DISTRIBUTION_VERSION v2.6.2

RUN mkdir src\github.com\docker ; \
    cd src\github.com\docker ; \
    git clone -q https://github.com/docker/distribution ; \
    cd distribution ; \
    git checkout -q $env:DISTRIBUTION_VERSION ; \
    go build -o registry.exe cmd/registry/main.go



FROM mcr.microsoft.com/windows/nanoserver:sac2016

COPY --from=build /gopath/src/github.com/docker/distribution/registry.exe /registry.exe
COPY config.yml /config/config.yml

EXPOSE 5000

ENTRYPOINT ["\\registry.exe"]
CMD ["serve", "/config/config.yml"]
```

# docker-compose recreate

```
docker-compose up --force-recreate --build -d
```

```
docker image prune -f
```

# docker-compose install

Versionen: https://github.com/docker/compose/releases

Ubuntu

wget https://github.com/docker/compose/releases/download/v2.2.2/docker-compose-linux-x86_64 .

mv ocker-compose-linux-x86_64 docker-compose

chmod +x docker-compose

mv docker-compose /usr/bin/docker-compose