

Spickzettel („Cheat Sheet“) Docker

Autor: Dr. Holger Schwichtenberg (www.IT-Visions.de)

V0.1 / 29.05.2018 / Seite 1 von 2

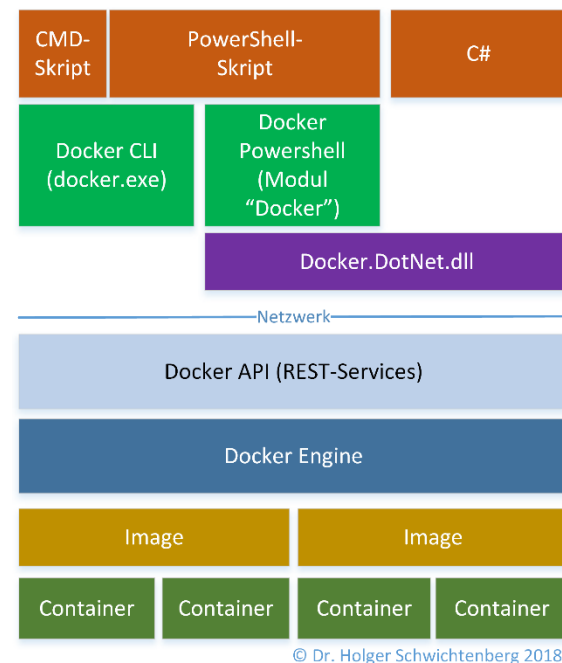
Docker-Optionen unter Windows

Container	Container	Container	Container	Container	Container	Container	Container	Container	Container
optional weitere Anwendungsimage(s)		optional weitere Anwendungsimage(s)		optional weitere Anwendungsimage(s)		optional weitere Anwendungsimage(s)		optional weitere Anwendungsimage(s)	
Anwendungsimage für Linux-Anwendung(en)		Anwendungsimage für Linux-Anwendung(en)		Anwendungsimage für Linux-Anwendung(en)		Anwendungsimage für Windows-Anwendung(en)		Anwendungsimage für Windows-Anwendung(en)	
Base Image Boot2Docker Linux		LinuxKit		Base Image Alpine Linux		Base Image Windows Server 2016 Core oder Nano v1/v2		Base Image Windows Server 2016 Core oder Nano v1/v2	
VirtualBox VM "default"		Linux Containers on Windows (LCOW)		Hyper-V VM "MobyLinuxVM"		Hyper-V-Container VMWP.cxx mit Mini-Windows-Kernel		Windows Server Container	
Produkt „Docker Toolbox“ von Docker Inc		Produkt „Docker for Windows CE/EE“ von Docker Inc				In Windows enthaltenes Feature „Windows Containers“ von Microsoft Corporation			
Windows Server ab 1709 von Microsoft Corporation				Windows Server 2016 inkl. Core (und Nano v1) von Microsoft Corporation					
Windows 10 ab 1709 von Microsoft Corporation				Windows 10 Professional, Enterprise oder Education von Microsoft Corporation					
ab Windows 7 / ab Windows Server 2008 R2 von Microsoft Corporation									

© Dr. Holger Schwichtenberg 2017 2018

© Dr. Holger Schwichtenberg 2017-2018

Docker-Programmierschnittstellen



© Dr. Holger Schwichtenberg 2018

Hinweis

Die meisten Beispiele auf diesem Spickzettel setzen voraus, dass die gezeigten Befehle in der Windows PowerShell (also nicht der CMD!) ausgeführt werden. Docker-Befehle mit Interaktion laufen nur in der PowerShell-Konsole, aber nicht in der Windows PowerShell ISE! Einige Beispiele setzen zudem die Installation des "Docker"-Moduls in der PowerShell voraus.

Windows Container auf Windows Server installieren

Add-WindowsFeature Containers

Restart-Computer

Install-Module DockerProvider

Install-Package Docker -ProviderName DockerProvider -

RequiredVersion preview

Restart-Computer

Docker for Windows 10 installieren

<https://store.docker.com/editions/community/docker-ce-desktop-windows>

Docker-Systeminformationen

Versionsinfo

docker --version

Disk Usage

docker system df

Systemeinstellungen

docker info # Alias: docker system info

Images laden von Docker Hub

Linux-Image von Docker-Hub laden

docker pull microsoft/mssql-server-linux

Linux-Image von Docker-Hub laden im LCOW-Modus

docker pull microsoft/mssql-server-linux --platform=linux

Windows-Base-Image laden für Windows Server 1709

docker pull microsoft/windowsservercore:1709

Windows-Image für IIS laden auf Windows Server 1709

docker pull microsoft/iis:windowsservercore-1709

Images auflisten und analysieren

Images auflisten

docker images

Details über ein Image

docker inspect microsoft/iis:windowsservercore-1709

Schichten eines Image

docker history microsoft/iis:windowsservercore-1709

Container starten

IIS-Container starten im Hintergrund mit Portmapping

docker run -d -p 8111:80 --name IIS1 microsoft/iis:windowsservercore-1709

IIS-Container explizit als Hyper-V-Container starten im Hintergrund mit Portmapping

docker run -d -p 8111:80 --isolation=hyperv --name IIS2

microsoft/iis:windowsservercore-1709

MSSQL-Container starten im Hintergrund mit Portmapping

docker run -e "ACCEPT_EULA=Y" -e

"MSSQL_SA_PASSWORD=geheim+123" -e "MSSQL_PID=Developer" -p 1434:1433 --name SQL8 --restart always -d microsoft/mssql-server-linux

CMD in Windows-Container interaktiv starten

docker run -it --name cmd1 microsoft/windowsservercore-1709 cmd

Bash in Linux-Container interaktiv starten

docker run -it --name ub1 ubuntu:17.10 bash

Container auflisten

Alle laufenden Container auflisten

docker ps

Alle Container auflisten

docker ps -a

Alle Container mit IIS im Namen auflisten

docker ps -a -q -f name=IIS*

Container analysieren

Details zu einem Container

docker inspect IIS1

Prozesse, die in dem Container laufen

docker container top IIS1

Containerart auslesen: liefert hyperv oder process

docker inspect --format="{{.HostConfig.Isolation}}" IIS1

IP-Adresse auslesen

docker inspect --format '{{.NetworkSettings.Networks.nat.IPAddress }}' IIS1

Laufend Ressourcennutzung aller Container anzeigen

docker container stats

Befehle in laufendem Container ausführen

Einzelne Befehle im Container ausführen

docker exec IIS1 powershell del C:\inetpub\wwwroot*. *

Interaktiv in den Container gehen

docker exec -it IIS1 cmd

docker exec -it IIS1 powershell

Dateien in Container kopieren

In Hyper-V-Container kann man nur kopieren, wenn er gestoppt ist!

\$art = docker inspect --format="{{.HostConfig.Isolation}}" IIS1

if (\$art -eq "hyperv") { docker stop IIS1 }

\$Quelle = "h:\ml\dist-prod"

foreach(\$datei in (dir \$Quelle))

{

docker cp \$datei.Fullname IIS1:C:\inetpub\wwwroot

}

if (\$art -eq "hyperv") { docker start IIS1 }

explorer http://\$([System.Environment]::MachineName):8111

Spickzettel („Cheat Sheet“) Docker

Autor: Dr. Holger Schwichtenberg (www.IT-Visions.de)

V0.1 / 29.05.2018 / Seite 2 von 2

Eigenes Docker-Image erstellen und veröffentlichen

```
# In Windows muss man Container zuerst anhalten
docker stop IIS1
# Image erstellen. Tagname darf keine Großbuchstaben enthalten!
docker commit IIS1 itvisions/miraclelistwebclient:v0.5
# Zusatztag vergeben
docker tag itvisions/miraclelistwebclient:v0.5
itvisions/miraclelistwebclient:latest
# Veröffentlichen auf Docker Hub
docker login
docker push itvisions/miraclelistwebclient:latest
```

Problem diagnose

```
# Ereignisse laufend ausgeben
docker events # Alias: Docker system events // in eigenem Prozess!
# Details zu einem Ereignis (ID aus docker events)
docker logs docker logs e24d0...
```

Docker aufräumen

```
# Alle Container stoppen, die mit IIS beginnen
docker stop $(docker ps -a -q -f name=IIS*)
# Alle Container löschen, die mit IIS beginnen
docker rm $(docker ps -a -q -f name=IIS*)
# Aufräumen (gestoppte Container, nicht verwendete Images)
docker system prune
```

Docker-Modul in PowerShell installieren

```
$ErrorActionPreference = "stop"
"Docker PowerShell installieren..."
$MyModuleDir = $env:PSModulePath.Split(";") | Where { $_ -like
"*$([system.Environment]::Username)\*" }
[Net.ServicePointManager]::SecurityProtocol = "Tls12, Tls11, Tls"
Invoke-WebRequest https://github.com/Microsoft/Docker-
PowerShell/releases/download/v0.1.0/Docker.0.1.0.zip -OutFile
$MyModuleDir/dockerpowershell.zip
Expand-Archive -Path $MyModuleDir/dockerpowershell.zip -
DestinationPath $MyModuleDir\Docker
Import-Module Docker
```

Docker-Modul in PowerShell verwenden

```
# Alle Befehle des Moduls auflisten
Get-Command -module docker
# Image laden
Pull-ContainerImage microsoft/iis:windowsservercore-1709 -Verbose
# Container mit Portmapping starten
$config = [Docker.DotNet.Models.Config]::new()
($config.ExposedPorts =
[System.Collections.Generic.Dictionary[string,object]]::new()).Add("80/tc
p", $null)
$hostConfig = [Docker.DotNet.Models.HostConfig]::new()
$pb = new-object Docker.DotNet.Models.PortBinding
$pb.HostPort = "8112"
```

```
$hostConfig.PortBindings =
[System.Collections.Generic.Dictionary[string,
System.Collections.Generic.IList[Docker.DotNet.Models.PortBinding]]]::n
ew()
$hostConfig.PortBindings.Add("80/tcp", [System.Collections.Generic.List[
Docker.DotNet.Models.PortBinding]]::new([Docker.DotNet.Models.PortB
inding[]]@($pb)))
Run-ContainerImage -ID microsoft/iis:windowsservercore-1709 -Name
IIS2 -Isolation Hyperv -Configuration $config -Detach
# Alle Images auflisten
Get-ContainerImage
# Alle Images mit IIS im Namen
Get-ContainerImage | where repotags -like "*IIS*"
# Alle Container auflisten
Get-Container
# Alle laufenden container anhalten
Get-Container | where { $_.names -like "*IIS*" -and $_.state -eq
"running" } | Stop-Container
# Container starten
Start-Container IIS1
# Prozess in Container starten
Start-ContainerProcess -ID IIS2 -Command PowerShell -input
# Alle gestoppten Container löschen
Get-Container | where { $_.state -ne "running" } | Remove-Container
# Alle Container löschen
Get-Container | Remove-Container -force
```

Docker.DotNet in .NET/.NET Core-Projekt einfügen

Install-Package Docker.DotNet

Docker.DotNet verwenden (in C#)

```
DockerClient client = new DockerClientConfiguration(new
Uri("npip://./pipe/docker_engine"))
.CreateClient();
Console.WriteLine("Starte neuen Container...");
var hc = new HostConfig();
var pb = new Docker.DotNet.Models.PortBinding
{
    HostPort = "8113"
};
var x = new
System.Collections.Generic.List<Docker.DotNet.Models.PortBinding>();
x.Add(pb);
hc.PortBindings = new Dictionary<string, IList<PortBinding>>();
hc.PortBindings.Add("80/tcp", x);
var parameters = new Config
{
    Image = "microsoft/iis:windowsservercore-1709"
};
var ccp = new CreateContainerParameters(parameters)
{
    Name = "IIS3",
    HostConfig = hc
```

```
};
CreateContainerResponse response = await
client.Containers.CreateContainerAsync(ccp);
var config = new ContainerStartParameters();
bool ok = await client.Containers.StartContainerAsync(response.ID,
config);
if (ok) { Console.WriteLine("Container angelegt!"); }

Console.WriteLine("Liste aller Container...");
IList<ContainerListResponse> containers = await
client.Containers.ListContainersAsync(
    new ContainersListParameters()
    {
        Limit = 10,
    });
foreach (var c in containers)
{
    Console.WriteLine(c.State + " Container " + String.Join(", ", c.Names)
+ " created " + c.Created + " based on " + c.Image);
}
```

Links

Docker-Dokumentation
<https://docs.docker.com/>
<https://docs.microsoft.com/de-de/virtualization/windowscontainers>
Docker Hub Container Registry
<https://hub.docker.com>
.NET Client for Docker Remote API
<https://github.com/Microsoft/Docker.DotNet>
Docker PowerShell
<https://github.com/Microsoft/Docker-PowerShell>
Microsofts Container-Dienste in der Azure-Cloud
<https://www.microsoft.com/de-de/cloud-platform/containers>

Über den Autor

Dr. Holger Schwichtenberg gehört zu den bekanntesten Experten für Webtechniken und .NET in Deutschland. Er hat zahlreiche Fachbücher veröffentlicht und spricht regelmäßig auf Fachkonferenzen. Sie können ihn und seine Kollegen für Entwicklungsarbeiten, Schulungen, Beratungen und Coaching buchen.
E-Mail: buer0@IT-Visions.de
Website: www.IT-Visions.de
Weblog: www.dotnet-doktor.de

