

SQL on Docker

Frank Geisler

CEO GDS Business Intelligence GmbH, MVP
frank_geisler@geislars.net

GDS Business Intelligence GmbH





Session Feedback Day
(not optional!)

<http://bit.ly/DataGrillen2019Day>



Session Feedback Day (not optional!)

<http://bit.ly/DataGrillen2019Day2>



Event Feedback (not optional!)

<http://bit.ly/DataGrillen2019Ever>



Frank Geisler

frank_geisler@geislars.net

Geschäftsführer

GDS Business Intelligence GmbH

Themenschwerpunkte:

Business Intelligence, Sharepoint, Programmierung, Software Engineering

Data Platform MVP

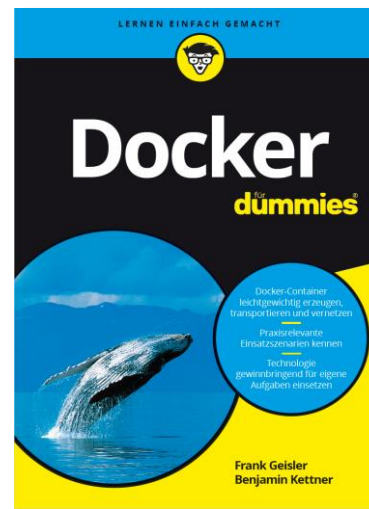
Microsoft P-TSP

Regionalgruppenleiter PASS Regionalgruppe Ruhrgebiet

Director Marketing & Communications PASS e.V.

Autor

Speaker



**Microsoft
CERTIFIED**

Solutions Developer

Azure Solutions
Architect

**Microsoft
CERTIFIED**

Trainer

**Microsoft
CERTIFIED**

Solutions Expert

Business Intelligence
Data Platform

**Microsoft
CERTIFIED**

Solutions Associate

SQL Server 2012

**Microsoft
CERTIFIED**

Technology Specialist

**Microsoft
CERTIFIED**

IT Professional

Business Intelligence
Developer 2008

**Microsoft
CERTIFIED**

Professional



 @FrankGeisler

Please Talk Data To Me

Der Data Platform Podcast mit Biml Ben, Mr. T und Angry Frank



Adaptive Query Processing ADF Azure Azure Data Studio **Azure Notebooks** Azure Stack Big Data Clusters **Biml** Black Panther Business Application Summit 2018 Data Platform **Data Platform Summit** dbatools **Docker** Flensburger Radler Alkoholfrei GDPR **Git Hub** Ignite Jupyter Notebooks Kubernetes **Las Vegas** Lissabon Microsoft Professional Program MPP **PASS Camp** PASS Deutschland e.V. PASS Essentials PASS Summit **Power BI** PowerShell Query Folding Regionalgruppen Solo SQL Management Studio 18 - Preview SQL Operations Studio **SQL Saturday** SQL Server 2019 Tabular Tomb Raider **tSQLt** TugalT Visual Studio Code **WDC**



Ben Weissman
Biml Ben



Tillmann Eitelberg
Mr. T



Frank Geisler
Angry Frank



12

Episoden



2921

Downloads



1275

Sendeminuten



13

Gäste

<https://www.pleasetalkdatatome.de>

Agenda

- Die Container Theorie
- Windows und Container
- Einen SQL Server Container bauen
- Ein angepasstes Container Image bauen
- Images exportieren
- Angepasste Images auf Docker Hub veröffentlichen
- Eine nette Oberfläche für Docker
- Einsatzszenarien
- Referenzen

Die Container Theorie



Definition Container

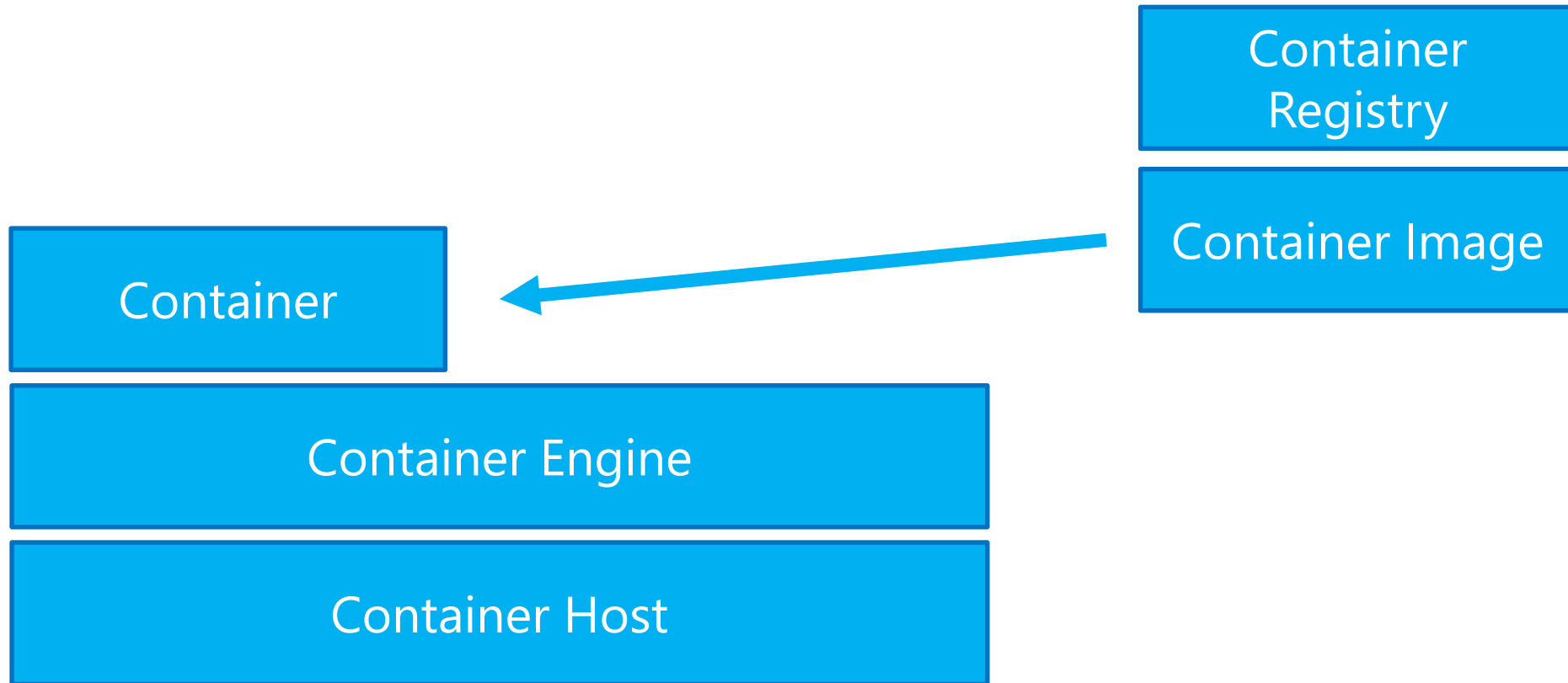
Container *kapseln* eine Software in einem vollständigen *Dateisystem* das alles enthält was der Container *benötigt* um ausgeführt werden zu können: Code, Laufzeitumgebung, System Tools, System Bibliotheken – alles was man auf einem Server installieren kann. Das *garantiert*, dass die Software immer ausgeführt werden kann, egal auf *welcher Umgebung*.

<https://www.docker.com/what-docker>

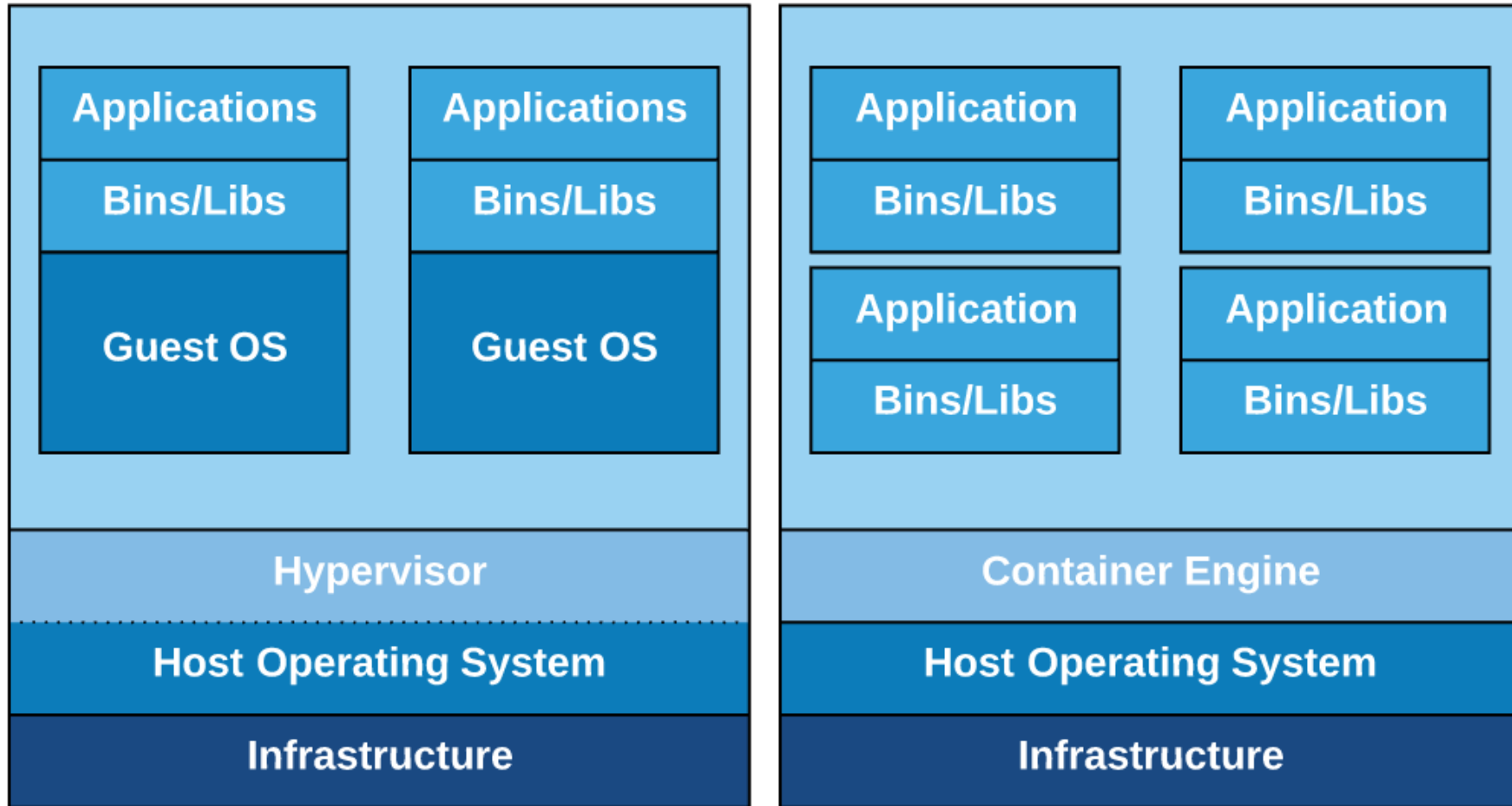
Grundlegende Elemente von Docker

- Container Image
- Container
- Container Host
- Container Engine
- Container Registry

Zusammenhang der Elemente



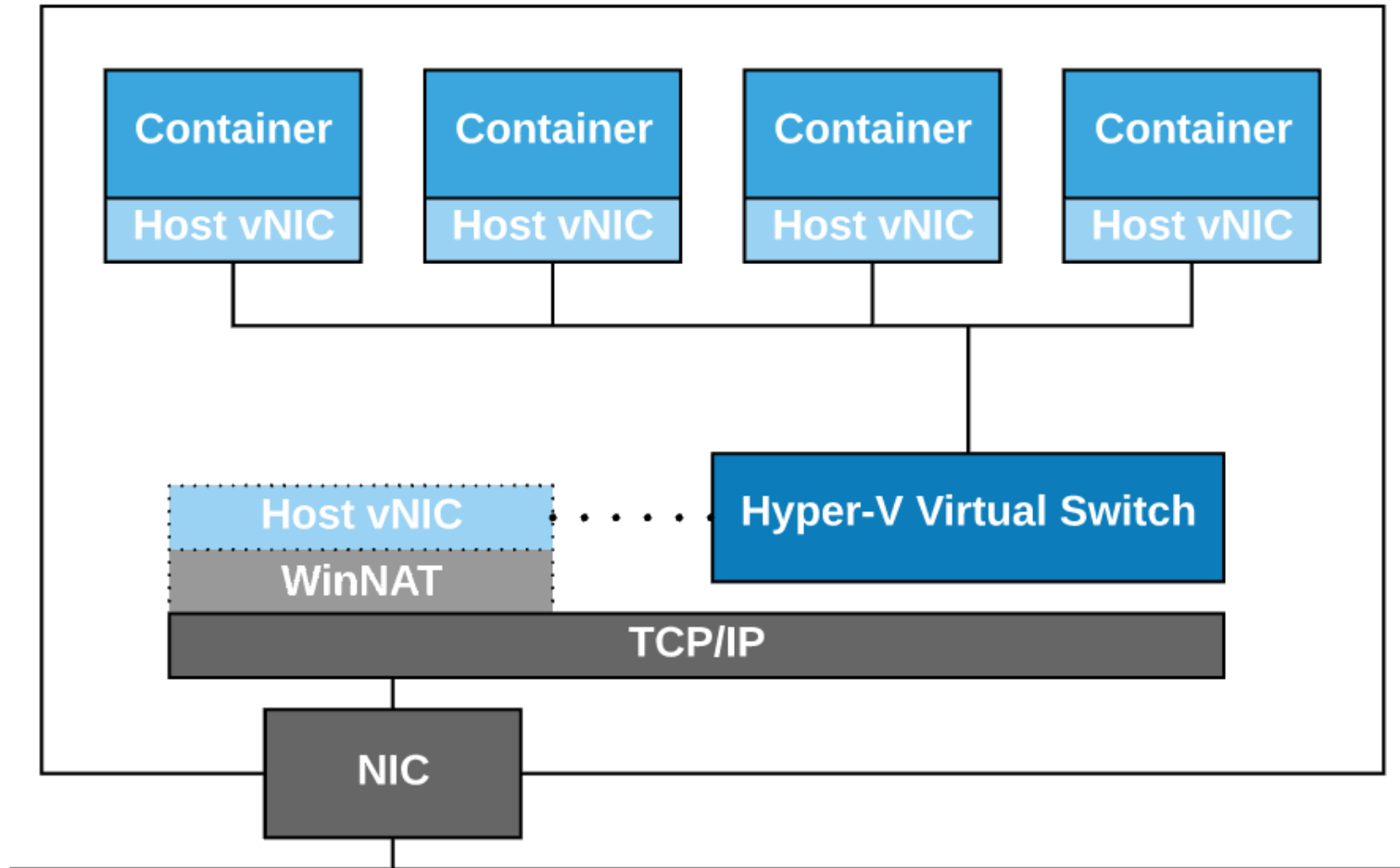
Virtuelle Maschinen vs. Container



Container sind keine VMs !

- Sie sind nur Prozesse
- Dadurch begrenzt welche Ressourcen verwendet werden können
- Werden beendet wenn der Prozess stoppt

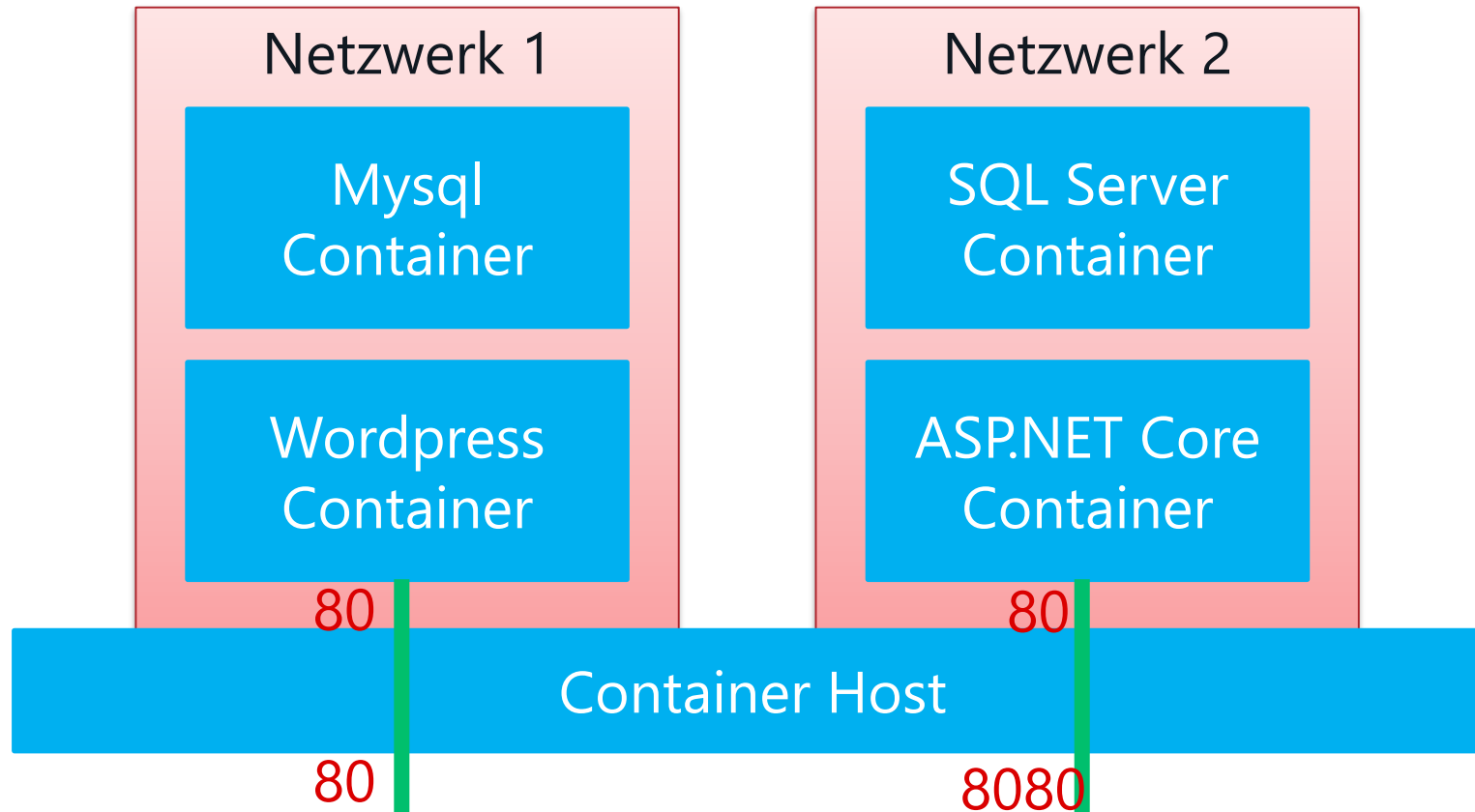
Container Netzwerk



Container Netzwerk

- Jeder Container wird standardmäßig mit einem privaten virtuellen Netzwerk verbunden das „bridge“ heißt
- Jedes virtuelle Netzwerk wird über eine NAT Firewall an die Host IP durchgeroutet
- Container im selben virtuellen Netzwerk können miteinander kommunizieren (ohne -p)
- Best practice: Für jede App ein eigenes Netzwerk erstellen

Best practices



Was ist in einem Image

- App Binaries und Abhängigkeiten
- Metadaten über die Image Daten und wie das Image ausgeführt wird
- Offizielle Definition: „An image is an ordered collection of root filesystem changes and the corresponding execution parameters for use within a container runtime.“
- Kein vollständiges OS, Kein Kernel, keine Kernel module (z.B. Treiber)
- Kann sehr klein oder sehr groß sein.

Windows und Container



Docker und Microsoft

Haben 2014 angekündigt
zusammenzuarbeiten

In Windows Server 2016 und Windows
10 Anniversary Edition werden Container
unterstützt

Docker and Microsoft partner to bring container applications across platforms

October 15, 2014 | Microsoft News Center



Editor's note – Oct. 16, 2014 –The press release below was updated to clarify that the orchestration of containers on Azure is planned to be integrated in a future Docker release.

SAN FRANCISCO and REDMOND, Wash. — Oct. 15, 2014 — Microsoft Corp. and [Docker Inc.](#), the company behind the fast-growing Docker open platform for distributed applications, on Wednesday announced a strategic partnership to provide Docker with support for new container technologies that will be delivered in a future release of Windows Server. Developers and organizations that want to create container applications using Docker will be able to use either Windows Server or Linux with the same growing Docker ecosystem of users, applications and tools.

Today's business climate requires higher levels of innovation than before, and distributed applications that support open portability are at the forefront of this demand. Docker is

<http://news.microsoft.com/2014/10/15/DockerPR/>

Container Typen

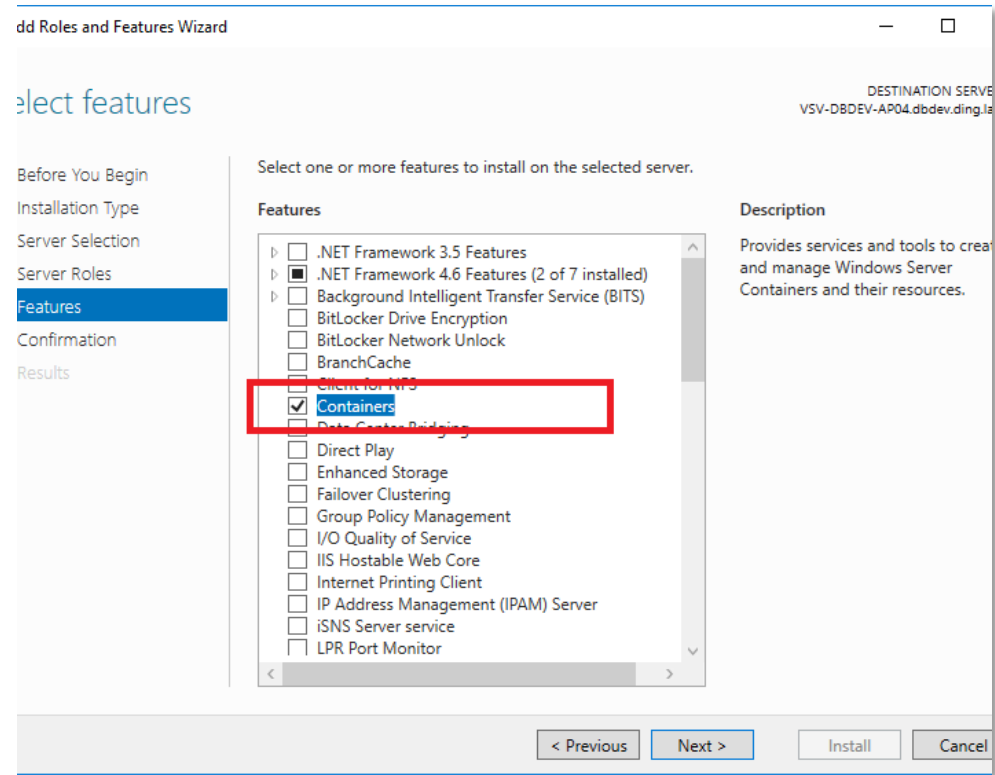
Windows Server Container

- Isolation über Namespace, Ressource und Prozess Isolierung
- Teilen das Kernel mit dem Host und anderen Containern

Hyper-V Container

- Erweitert die Isolierung gegenüber Windows Server Containern
- Läuft innerhalb einer speziellen virtuellen Maschine
- Teilt sich das Kernel nicht mit dem Host
- Wird zur Laufzeit mit **--isolation=hyperv** festgelegt
- Kann auch Linux als Host laufen lassen

Container Feature aktivieren



Docker Engine installieren - PowerShell

```
Install-PackageProvider -Name NuGet -MinimumVersion  
2.8.5.201 -Force
```

```
Install-Module -Name DockerMsftProvider -Force
```

```
Install-Package -Name docker -ProviderName  
DockerMsftProvider -Force
```

```
Restart-Computer -Force
```


Windows 10 - Docker Store

.msi herunterladen

<https://store.docker.com/>

Get Docker CE for Windows

Stable channel

This installer is fully baked and tested. This is the best channel to use if you want a reliable platform to work with.

These releases follow the Docker Engine stable releases.

[Get Docker CE for Windows \(stable\)](#)

Edge channel

This installer provides the latest Edge release of Docker for Windows and Engine, and typically offers new features in development.

Use this channel if you want to get experimental features faster, and can weather some instability and bugs. We collect all usage data on Edge releases across the board.

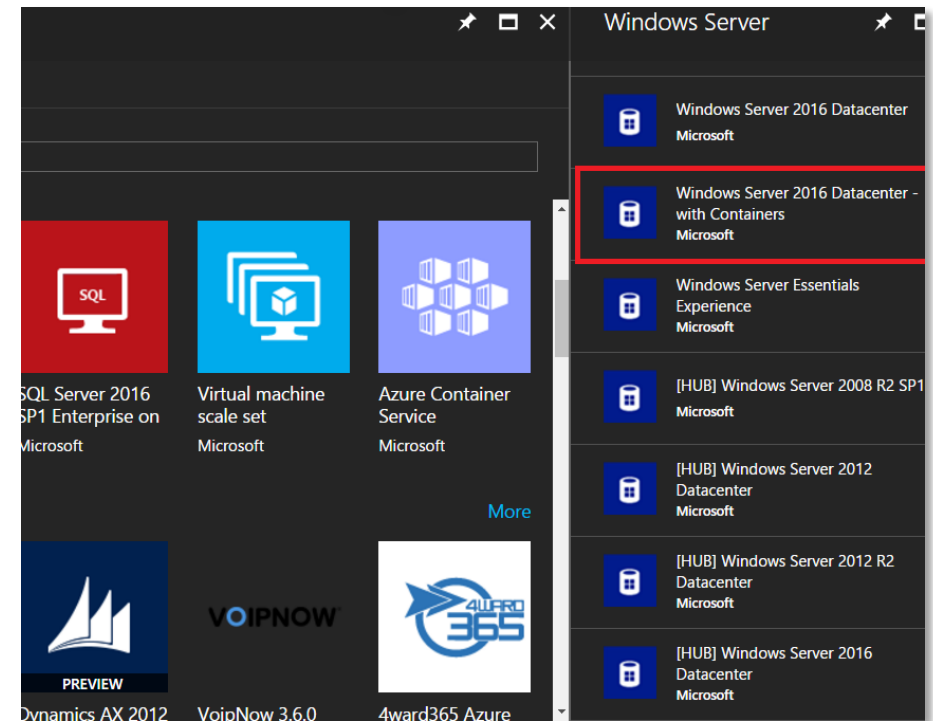
Edge builds are released once per month.

[Get Docker CE for Windows \(Edge\)](#)

Azure

Es gibt viele Möglichkeiten...

U.a.: Es gibt ein Image in dem bereits die Container Rolle aktiviert und die Docker Engine installiert ist.



Prüfen ob Docker läuft

Get-Service docker

```
Administrator: Windows PowerShell
PS C:\windows\system32> get-service docker

Status      Name      DisplayName
-----
Running     Docker    docker

PS C:\windows\system32> _
```

docker version

```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker version
Client:
 Version:      17.03.2-ee-5
 API version:  1.27
 Go version:   go1.7.5
 Git commit:   fa09039
 Built:        Wed Jul 19 23:56:45 2017
 OS/Arch:      windows/amd64

Server:
 Version:      17.03.2-ee-5
 API version:  1.27 (minimum version 1.24)
 Go version:   go1.7.5
 Git commit:   fa09039
 Built:        Wed Jul 19 23:56:45 2017
 OS/Arch:      windows/amd64
 Experimental: false
PS C:\Windows\system32> _
```

Docker mit älterem Windows / SQL Server

- www.windocks.com
- Eine Portierung des Open Source Projektes von Docker Inc.
- Software unterstützt die Erstellung von Containern auf denen frühere Versionen von SQL Server (2008+) auf Windows Server 2012 ausgeführt werden
- Kostenlose Community Edition verfügbar
www.windocks.com/leads/add?src=downloadcommunity



Einen SQL Server Container bauen



Das Docker Repository durchsuchen

docker search microsoft/mssql

```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker search microsoft/mssql
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
microsoft/aspnet	ASP.NET is an open source server-side Web ...	636		[OK]
microsoft/dotnet	Official images for .NET Core for Linux an...	608		[OK]
microsoft/mssql-server-linux	Official images for Microsoft SQL Server o...	358		
mono	Mono is an open source implementation of M...	235	[OK]	
microsoft/windowsservercore	Windows Server 2016 Server Core base OS im...	205		
microsoft/aspnetcore	Official images for running compiled ASP.N...	202		[OK]
microsoft/nanoserver	Windows Server 2016 Nano Server base OS im...	199		
microsoft/iis	Internet Information Services (IIS) instal...	150		
microsoft/mssql-server-windows-express	Official Microsoft SQL Server Express Edit...	101		
microsoft/azure-cli	Docker image for Microsoft Azure Command L...	92		[OK]
microsoft/mssql-server-windows	Official images for Microsoft SQL Server f...	80		
microsoft/aspnetcore-build	Official images for building ASP.NET Core ...	70		[OK]
microsoft/mssql-server-windows-developer	Official Microsoft SQL Server Developer Ed...	44		
microsoft/vsts-agent	Official images for the Visual Studio Team...	31		
microsoft/oms	Monitor your containers using the Operatio...	28		[OK]
microsoft/dotnet-samples	.NET Core Docker Samples	23		[OK]
microsoft/cntk	CNTK images from github.com/Microsoft/CNTK...	16		[OK]
microsoft/applicationinsights	Application Insights for Docker helps you ...	8		[OK]
rsmoorthy/mssql	MSSQL Database (version SQL2000)	7		[OK]
microsoft/dotnet-nightly	Preview bits of the .NET Core CLI	5		[OK]
microsoft/dotnet-buildtools-prereqs	Images for building the various components...	5		
swapnilinux/mssql	Microsoft SQL Server (mssql) vNext CTP 1.1...	2		[OK]
microsoft/draft	A tool for developers to create cloud-nati...	2		
softwareplant/mssql	SQL Server test database	0		[OK]
astronomerio/mssql-source	MSSQL source.	0		[OK]

```
PS C:\Windows\system32>
```

Ein Image herunterladen

docker pull microsoft/mssql-server-linux

Administrator: Windows PowerShell

```
PS C:\Windows\system32> docker pull microsoft/mssql-server-windows
Using default tag: latest
latest: Pulling from microsoft/mssql-server-windows
3889bb8d808b: Downloading [>] 16.78 MB/4.07 GB
e29afd68a947: Downloading [>] 17.32 MB/1.225 GB
bd43224dc30b: Download complete
8b0535843b49: Download complete
4374dfc05b90: Download complete
0d8d285a896c: Download complete
180220843db4: Download complete
4ce1c9b6cde5: Download complete
63478436438d: Download complete
7dd8189c4efc: Download complete
cd74dcdbdc6c: Downloading [>] 8.081 MB/1.62 GB
7367ef909a12: Waiting
cfc787c1a9c4: Waiting
1bea40cabe16: Waiting
```

Das Image überprüfen

docker images

Administrator: Windows PowerShell

```
PS C:\docker> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
microsoft/mssql-server-windows	latest	671720e6a1ea	2 weeks ago	13 GB

```
PS C:\docker> 
```

Einen Container aus dem Image erstellen

```
docker container run -d -p 1433:1433 --env ACCEPT_EULA=Y --env SA_PASSWORD=!test123 --name sqlcontainer microsoft/mssql-server-linux
```



The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The command entered is: `docker run -d -p 15789:1433 --env ACCEPT_EULA=Y --env sa_password=Testing11@ --name MyFirstContainer microsoft/mssql-server-windows`. The output shows the container ID: `3dc5591d6561ff4d32823fd7223f3f5a7a5d6fcf51e51b311982076cc8746233`. The prompt then returns to `PS C:\Windows\system32>`.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker run -d -p 15789:1433 --env ACCEPT_EULA=Y --env sa_password=Testing11@ --name MyFirstContainer microsoft/mssql-server-windows
3dc5591d6561ff4d32823fd7223f3f5a7a5d6fcf51e51b311982076cc8746233
PS C:\Windows\system32>
```

Was passiert eigentlich bei docker run im Hintergrund?

1. Docker schaut ob das Image lokal im cache liegt (in entspr. Version)
2. Wird Image nicht gefunden wird in remote image repository nachgeschaut (Standard: Docker Hub)
3. Lädt die neuste Version runter (in unserem Fall, da nichts angegeben)
4. Erstellt einen neuen Container auf Basis des Images und bereitet den Start vor
5. Gibt dem Container eine virtuelle IP-Adresse auf dem privaten Docker Netzwerk
6. Bindet den internen Port 1433 and den externen Port 1433
7. Startet den Container und führt die CMD Befehle im Dockerfile aus

Überprüfen dass der Container läuft

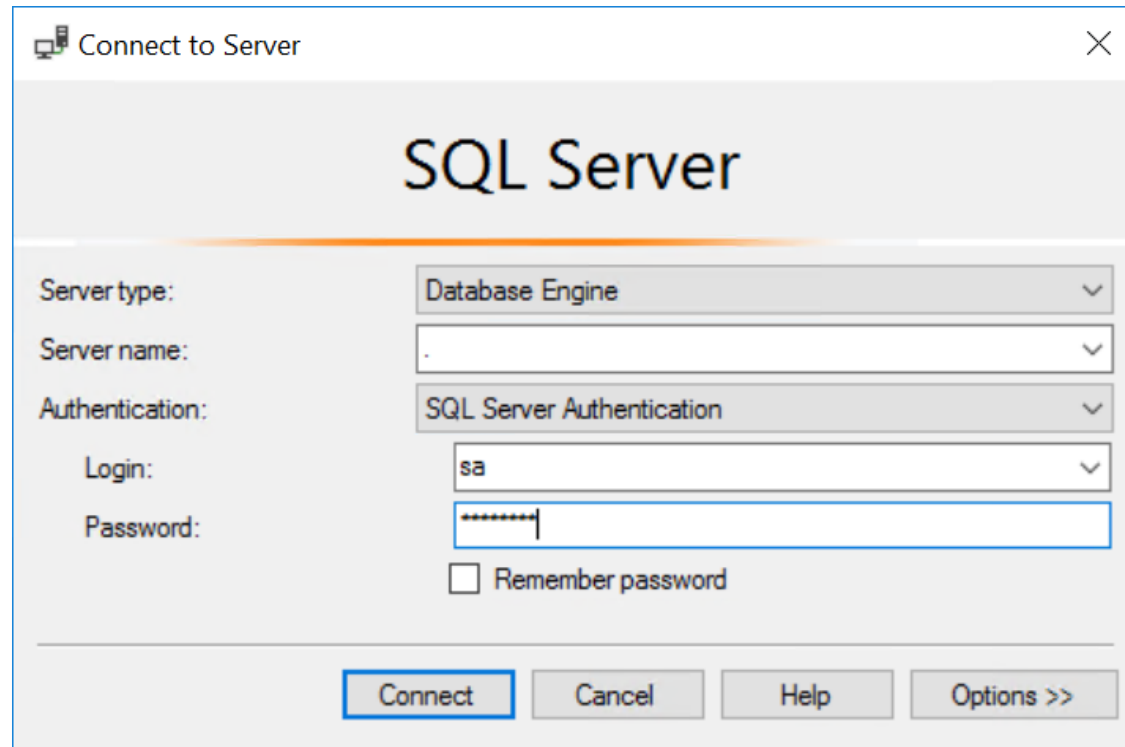
`docker ps [-a]`

```
Administrator: Windows PowerShell
PS C:\Windows\system32> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
RTS	NAMES			
3dc5591d6561	microsoft/mssql-server-windows	"cmd /S /C 'powers..."	29 seconds ago	Up 21 seconds
0.0.0:15789->1433/tcp	MyFirstContainer			

```
PS C:\Windows\system32>
```

Mit dem SQL Server verbinden



The screenshot shows the 'Connect to Server' dialog box. The title bar reads 'Connect to Server' with a close button. The main heading is 'SQL Server'. Below this, there are several fields and a checkbox:

- Server type:** A dropdown menu showing 'Database Engine'.
- Server name:** A dropdown menu showing '.'.
- Authentication:** A dropdown menu showing 'SQL Server Authentication'.
- Login:** A dropdown menu showing 'sa'.
- Password:** A text box containing a masked password (represented by asterisks).
- ☐ Remember password

At the bottom, there are four buttons: 'Connect' (highlighted with a blue border), 'Cancel', 'Help', and 'Options >>'.

Container überprüfen

docker inspect sqlcontainer

```
Administrator: Windows PowerShell

{"SecondaryIPAddresses": null,
 "SecondaryIPv6Addresses": null,
 "EndpointID": "",
 "Gateway": "",
 "GlobalIPv6Address": "",
 "GlobalIPv6PrefixLen": 0,
 "IPAddress": "",
 "IPPrefixLen": 0,
 "IPv6Gateway": "",
 "MacAddress": "",
 "Networks": {
  "nat": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "NetworkID": "9a8ee6b4f05a18a14f4feb52b1d43ef0cb45b509a467f99bb79e616189eadc7e",
    "EndpointID": "5e016622a227a6412ebd40559ac47dbc72e5c85efb3d657d1ce77623803125b7",
    "Gateway": "",
    "IPAddress": "172.27.167.239",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "MacAddress": "00:15:5d:cb:16:4a"
  }
 }
}
```

PS C:\windows\system32>

Mit dem Container verbinden

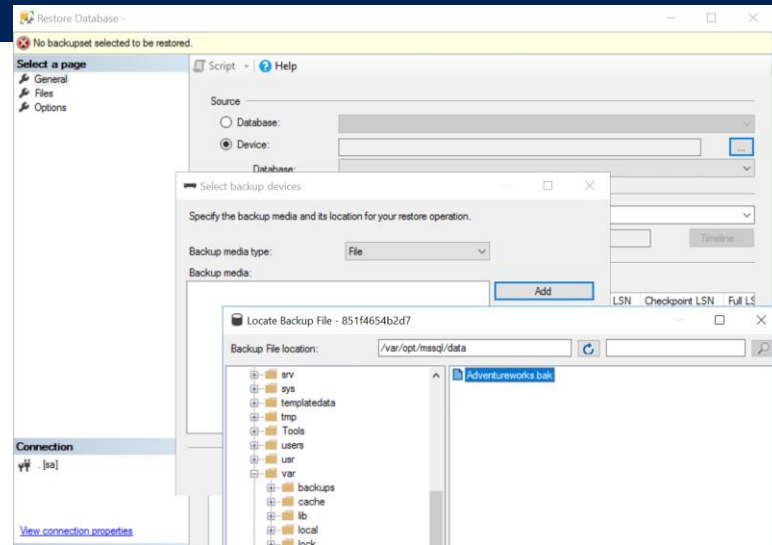
`docker exec -it sqlcontainer bash`

```
PS C:\Users\testadmin> docker exec -it sqlcontainer bash
root@851f4654b2d7:/# ls
bin    core  etc   install.sh  lib64  mnt  proc  run  srv  tmp  var
boot  dev   home  lib        media  opt  root  sbin sys  usr
root@851f4654b2d7:/# cd /var/opt/mssql/data
root@851f4654b2d7:/var/opt/mssql/data# ls
master.mdf  mastlog.ldf  model.mdf  modellog.ldf  msdbdata.mdf  msdblog.ldf  tempdb.mdf  templog.ldf
root@851f4654b2d7:/var/opt/mssql/data#
```

Dateien in den Container kopieren

```
docker cp c:\temp\Adventureworks.bak  
sqlcontainer:/var/opt/mssql/data/Adventureworks.ba  
k
```

```
PS C:\Users\testadmin> docker cp c:\temp\Adventureworks.bak sqlcontainer:/var/opt/mssql/data/Adventureworks.bak  
PS C:\Users\testadmin> █
```





DEMO

START



SCALABLE



DISPLAY PROPERTIES



EASY STYLE

Research

- Marketing
- Customer

IDEAS

ACTION // PLAN

Tel. 1011 3

* send Email

how to do?

Chem!

Business Plan

Vorteile

Einfache und schnelle Installation

Neue Container können innerhalb von Sekunden bereitgestellt werden

Im Vergleich zu Virtuellen Maschine recht kleiner Footprint

Man kann Images anpassen

Zugriff auf das Docker Repository (Hunderte Images sind verfügbar)

Portierbar – Images können auf dem Docker Hub gespeichert werden

Vom Cloud Anbieter unabhängig

Nachteile für SQL Server

Momentan wird nur die relationale Engine unterstützt

Wird offiziell nur in Windows Server 2016 / Windows 10 Anniversary Edition unterstützt

Es gibt nur offizielle SQL Server Images für 2016 und 2017

SQL Images können recht groß sein (~13GB)

Kein Overcommitting der Ressourcen

Ein angepasstes Container Image bauen



Dockerfile

Eine Datei auf dem Docker Host die Befehle enthält die ein angepasstes Image erzeugen.

Dockerfile Code

```
FROM microsoft/mssql-server-linux:latest
LABEL maintainer Frank Geisler <frank_geisler@geislers.net>

# Create a directory where we can copy the Database files.
RUN mkdir /var/opt/sqldatabase

COPY AdventureWorks2017.mdf /var/opt/sqldatabase
COPY AdventureWorks2017_log.ldf /var/opt/sqldatabase

ENV MSSQL_PID=Developer
ENV SA_PASSWORD=!demo54321
ENV ACCEPT_EULA=Y

HEALTHCHECK --interval=10s \
    CMD /opt/mssql-tools/bin/sqlcmd -S . -U sa -P !demo54321 \
        -Q "CREATE DATABASE [AdventureWorks2017] ON (FILENAME =
'/var/opt/sqldatabase/AdventureWorks2017.mdf'),(FILENAME =
'/var/opt/sqldatabase/AdventureWorks2017_log.ldf') FOR ATTACH"
```

Ein Image bauen

docker build -t newsqlserverimage .

```
Deleted: sha256:ba90a57977efdde9bd944ef359221c4d2cec2586a97ad7228b788c53e4510c85
Deleted: sha256:f4ffc7f8cd83054c5c4eb031388ecef246d1c9a3431ab7e416c5750463d0345d
Deleted: sha256:5d6d3a1ccf6cf2a690ceaa3c5b5ed8a81ac47168b081ce673406774e94139525
Deleted: sha256:db8256d99d8a475e45a245507b6d1759587e967e66481bde17f5053ebff3924d
Deleted: sha256:239408281c79aa8f923aac8ad7204064db3c777d3d00093a52fb2ce3893c7a17
PS C:\Users\testadmin\Documents\SQL-Server-Sample-Docker-Container\mssql-server-linux-adventureworks2017> docker build -t newsqlserverimage .
Sending build context to Docker daemon 435.4MB
Step 1/9 : FROM microsoft/mssql-server-linux:latest
----> ab22b8353bbd
Step 2/9 : LABEL maintainer Frank Geisler <frank_geisler@geislars.net>
----> Running in 1a01d0e45faf
Removing intermediate container 1a01d0e45faf
----> 05d46752c2d6
Step 3/9 : RUN mkdir /var/opt/sql/database
----> Running in 5de26c06ab35
Removing intermediate container 5de26c06ab35
----> b53cfff1a3acb
Step 4/9 : COPY Adventureworks2017.mdf /var/opt/sql/database
----> 5ce0338662a2
Step 5/9 : COPY Adventureworks2017_log.ldf /var/opt/sql/database
----> a51ef5c6d760
Step 6/9 : ENV MSSQL_PID=Developer
----> Running in 3857ba589ab4
Removing intermediate container 3857ba589ab4
----> ec4aa84433c4
Step 7/9 : ENV SA_PASSWORD=!demo54321
----> Running in 6005764fdf82
Removing intermediate container 6005764fdf82
----> fa74a9ffbf73
Step 8/9 : ENV ACCEPT_EULA=Y
----> Running in 93d5f4ed6c60
Removing intermediate container 93d5f4ed6c60
----> a6342955060f
Step 9/9 : HEALTHCHECK --interval=10s CMD /opt/mssql-tools/bin/sqlcmd -S . -U sa -P !demo54321
TE DATABASE [Adventureworks2017] ON (FILENAME = '/var/opt/sql/database/Adventureworks2017.mdf'),(FILENAME = '
database/Adventureworks2017_log.ldf') FOR ATTACH"
----> Running in 160f675a62a6
Removing intermediate container 160f675a62a6
----> b6870db0de96
Successfully built b6870db0de96
Successfully tagged newsqlserverimage:latest
SECURITY WARNING: You are building a Docker image from windows against a non-Windows Docker host. All files
```

Das neue Image überprüfen

docker images

```
PS C:\Users\testadmin\Documents\SQL-Server-Sample-Docker-Container\mssql-server-linux-adventureworks2017> docker images
REPOSITORY              TAG                IMAGE ID           CREATED            SIZE
newsqserverimage        latest            b6870db0de96      11 seconds ago    1.77GB
microsoft/mssql-server-linux latest            ab22b8353bbd      10 days ago       1.42GB
PS C:\Users\testadmin\Documents\SQL-Server-Sample-Docker-Container\mssql-server-linux-adventureworks2017>
```

Einen neuen Container aus dem Image erstellen

```
docker run -d -p 1433:1433 --name  
newsqservercontainer newsqserverimage
```

```
PS C:\Users\testadmin\Documents\SQL-Server-Sample-Docker-Container\mssql-server-linux-adventureworks2017> docker conta  
er run -d -p 1433:1433 --name neuersqlcontainer newsqserverimage  
89b7b3d96ce33f107f2689ab977d9f46e94dbbba0ee74f72bd4d769f95abfcfa
```




Research
- Marketing
- Customer

DEMO

START



SCALABLE



DISPLAY PROPERTIES



EASY STYLE

Tel. 1011 3
* send Email

IDEAS
ACTION // PLAN

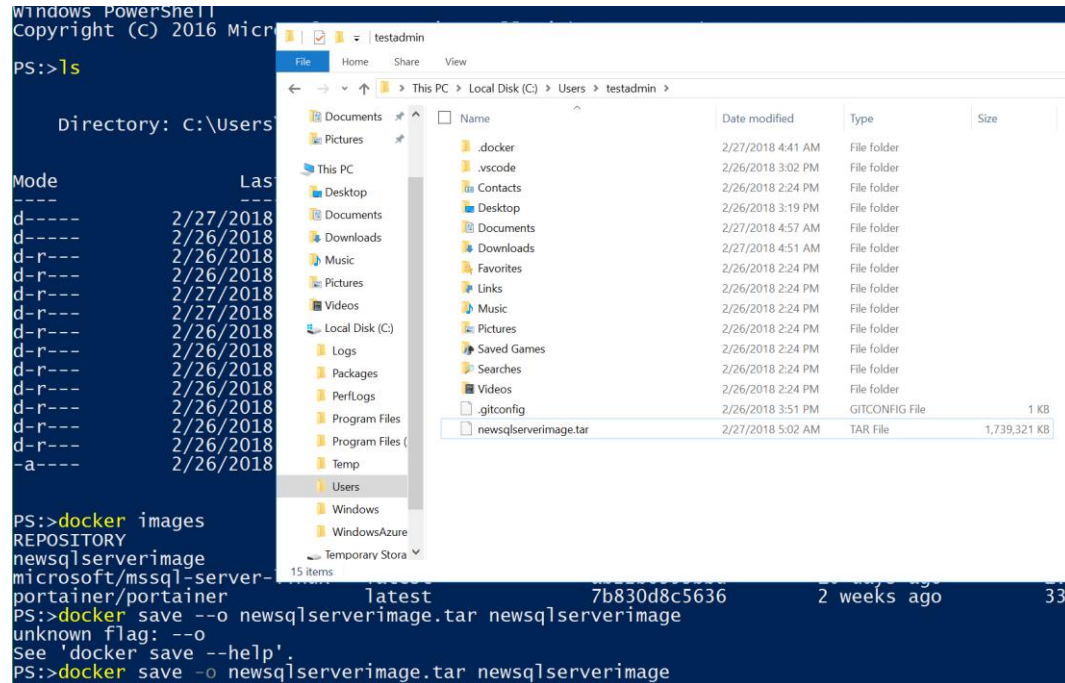
how to do ?
Chem!
Business Plan

Images exportieren



Lokal teilen

`docker save -o newsqserverimage.tar newsqserverimage`



The screenshot shows a Windows PowerShell terminal window and a File Explorer window. The PowerShell window displays the output of the `docker images` command, listing the `newsqserverimage` repository. Below this, the `docker save -o newsqserverimage.tar newsqserverimage` command is executed, resulting in an "unknown flag: --o" error. The File Explorer window shows the contents of the `C:\Users\testadmin` directory, where the `newsqserverimage.tar` file has been created, with a size of 1,739,321 KB.

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS:~>ls

Directory: C:\Users\testadmin

Mode                LastWriteTime         Length Name
----                -
d-----          2/27/2018 4:41 AM             0 .docker
d-----          2/26/2018 3:02 PM             0 .vscode
d-----          2/26/2018 2:24 PM             0 Contacts
d-----          2/26/2018 3:19 PM             0 Desktop
d-----          2/27/2018 4:57 AM             0 Documents
d-----          2/27/2018 4:51 AM             0 Downloads
d-----          2/26/2018 2:24 PM             0 Favorites
d-----          2/26/2018 2:24 PM             0 Links
d-----          2/26/2018 2:24 PM             0 Music
d-----          2/26/2018 2:24 PM             0 Pictures
d-----          2/26/2018 2:24 PM             0 Saved Games
d-----          2/26/2018 2:24 PM             0 Searches
d-----          2/26/2018 2:24 PM             0 Videos
-r-----          2/26/2018 3:51 PM              1 KB .gitconfig
-r-----          2/27/2018 5:02 AM    1,739,321 KB newsqserverimage.tar

PS:~>docker images
REPOSITORY              TAG         IMAGE ID      CREATED      SIZE
newsqserverimage        latest      7b830d8c5636  2 weeks ago  33.1 MB
microsoft/mssql-server-latest      7b830d8c5636  2 weeks ago  33.1 MB
portainer/portainer      latest      7b830d8c5636  2 weeks ago  33.1 MB

PS:~>docker save -o newsqserverimage.tar newsqserverimage
unknown flag: --o
See 'docker save --help'.
PS:~>docker save -o newsqserverimage.tar newsqserverimage
```

Images importieren

`docker load -i newsqserverimage.tar`

Administrator: Windows PowerShell

```
PS C:\docker> docker load -i myfirstimage.tar
f358be10862c: Loading layer [=====>] 7.892 GB/7.892 GB
c28d44287ce5: Loading layer [=====>] 1.933 GB/1.933 GB
ac24122adf9f: Loading layer [=====>] 54.27 kB/54.27 kB
ccf3bc376e1d: Loading layer [=====>] 54.27 kB/54.27 kB
1f9a4e1f3c98: Loading layer [=====>] 54.27 kB/54.27 kB
913439280c63: Loading layer [=====>] 54.27 kB/54.27 kB
03b1efe1f810: Loading layer [=====>] 54.27 kB/54.27 kB
0f488af76f38: Loading layer [=====>] 54.27 kB/54.27 kB
ba96d3a4b724: Loading layer [=====>] 61.44 kB/61.44 kB
59fb9067fb55: Loading layer [=====>] 54.27 kB/54.27 kB
4d4f76869ac7: Loading layer [=====>] 1.867 GB/1.867 GB
7f80e72414d0: Loading layer [=====>] 1.137 GB/1.137 GB
```




Research
- Marketing
- Customer

DEMO

START



SCALABLE



DISPLAY PROPERTIES



EASY STYLE

Tel. 1011 3
* send Email

IDEAS
ACTION PLAN

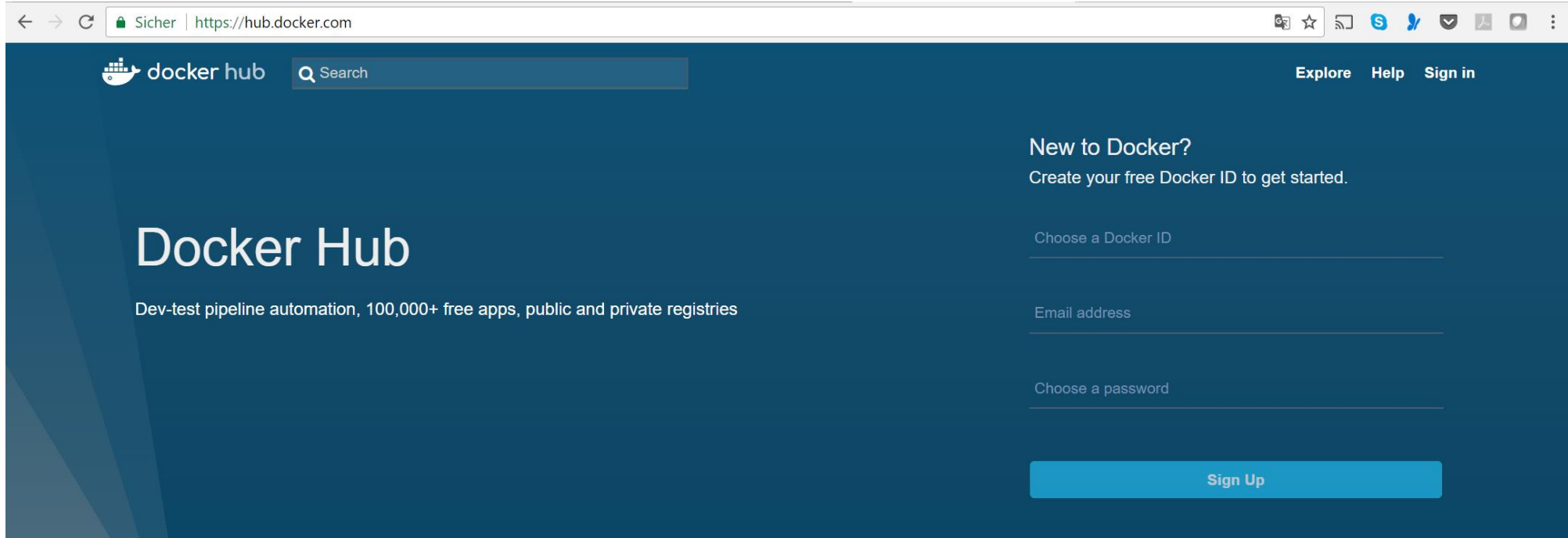
how to do?
Chem!
Business Plan

Angepasste Images auf Docker Hub veröffentlichen



Der Docker Hub

<https://hub.docker.com>



The screenshot shows the Docker Hub website in a web browser. The browser's address bar displays "Sicher | https://hub.docker.com". The website has a dark blue background. In the top left, there is a "docker hub" logo and a search bar with the placeholder text "Search". In the top right, there are links for "Explore", "Help", and "Sign in". The main content area on the left features the text "Docker Hub" in large white letters, followed by the tagline "Dev-test pipeline automation, 100,000+ free apps, public and private registries". On the right side, there is a section titled "New to Docker?" with the subtext "Create your free Docker ID to get started." Below this, there are three input fields: "Choose a Docker ID", "Email address", and "Choose a password". At the bottom of this section is a blue button labeled "Sign Up".

← → ↻ Sicher | https://hub.docker.com

docker hub Search

Explore Help Sign in

Docker Hub

Dev-test pipeline automation, 100,000+ free apps, public and private registries

New to Docker?

Create your free Docker ID to get started.

Choose a Docker ID

Email address

Choose a password

Sign Up

Ein Repository erstellen

Create Repository

1. Choose a namespace (*Required*)
2. Add a repository name (*Required*)
3. Add a short description
4. Add markdown to the full description field
5. Set it to be a private or public repository

frankgeisler

Enter Name

Short Description (100 Characters)

Full Description

Visibility

public

Create

Ein Image taggen

docker tag newsqserverimage
frankgeisler/newsqserverimage:v1

```
PS:>docker tag newsqserverimage frankgeisler/newsqserverimage:v1  
PS:>_
```


In Docker Hub einloggen

docker login

```
PS:>docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: frankgeisler
Password:
Login Succeeded
PS:>_
```

Docker Image hochladen

```
PS:>docker push frankgeisler/newsqserverimage:v1
The push refers to repository [docker.io/frankgeisler/newsqserverimage]
aacc577327d3: Pushed
1b938a328b45: Pushed
541171b43e86: Pushed
2c2b7e5b6217: Mounted from microsoft/mssql-server-linux
01249431b734: Mounted from microsoft/mssql-server-linux
45feb6b3c7be: Mounted from microsoft/mssql-server-linux
912a24c355e6: Mounted from microsoft/mssql-server-linux
bb83128af95f: Mounted from microsoft/mssql-server-linux
49907af65b0a: Mounted from microsoft/mssql-server-linux
4589f96366e6: Mounted from microsoft/mssql-server-linux
b97229212d30: Mounted from microsoft/mssql-server-linux
cd181336f142: Mounted from microsoft/mssql-server-linux
0f5ff0cf6a1c: Mounted from microsoft/mssql-server-linux
v1: digest: sha256:8850682e806e1356637878402eb3012670936a7b7bd262775fad96d987a3ebe7 size: 3041
PS:>_
```





Das veröffentlichte Image in Docker Hub

PUBLIC REPOSITORY

frankgeisler/newsqlserverimage ☆

Last pushed: 4 minutes ago

[Repo Info](#) [Tags](#) [Collaborators](#) [Webhooks](#) [Settings](#)

<p>Short Description </p> <p>Short description is empty for this repo.</p>	<p>Docker Pull Command </p> <pre>docker pull frankgeisler/newsqlserverimage</pre>
<p>Full Description </p> <p>Full description is empty for this repo.</p>	<p>Owner</p> <div> frankgeisler</div>



DEMO

START



SCALABLE



DISPLAY PROPERTIES



EASY STYLE

Research

- Marketing
- Customer

IDEAS

ACTION PLAN

Tel. 1011 3

* send Email

how to do?

Chem!

Business Plan

Eine nette Oberfläche für Docker



Portainer

- <http://portainer.io>
- Open Source Projekt
- Läuft selbst in Docker
- Zwei Befehle zum Starten:
 - `docker volume create portainer_data`
 - `docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer`

Portainer

The screenshot displays the Portainer v1.12.1 dashboard. On the left is a dark blue sidebar with the 'portainer.io' logo and a list of navigation items: Dashboard, App Templates, Containers, Images, Networks, Volumes, Events, Docker, and Portainer Settings (Password, Users, Endpoints). The main content area is titled 'Home Dashboard' and shows the 'local' active endpoint. A 'Node info' table lists system details for the 'thunderstruck' node. Below this, four summary cards show: 10 Containers (8 running, 2 stopped), 21 Images (2.6 GB), 21 Volumes (aufs driver), and 4 Networks. The user 'admin' is logged in, and the version 'v1.12.1' is shown at the bottom left.

portainer.io

ACTIVE ENDPOINT

local

ENDPOINT ACTIONS

Dashboard

App Templates

Containers

Images

Networks

Volumes

Events

Docker

PORTAINER SETTINGS

Password

Users

Endpoints

Home Dashboard

admin [log out](#)

Node info

Name	thunderstruck
Docker version	17.03.0-ce
CPU	8
Memory	8.3 GB

10 Containers 8 running, 2 stopped

21 Images 2.6 GB

21 Volumes aufs driver

4 Networks

Portainer v1.12.1



Research
- Marketing
- Customer

DEMO

START



SCALABLE



DISPLAY PROPERTIES



EASY STYLE

Tel. 1011 3
* send Email

IDEAS
ACTION PLAN

how to do?
Chem!
Business Plan

Einsatzszenarien



Einsatzszenarien

- Dev / Test
- DevOps
- Kein Know How über die Serveranwendung vorhanden

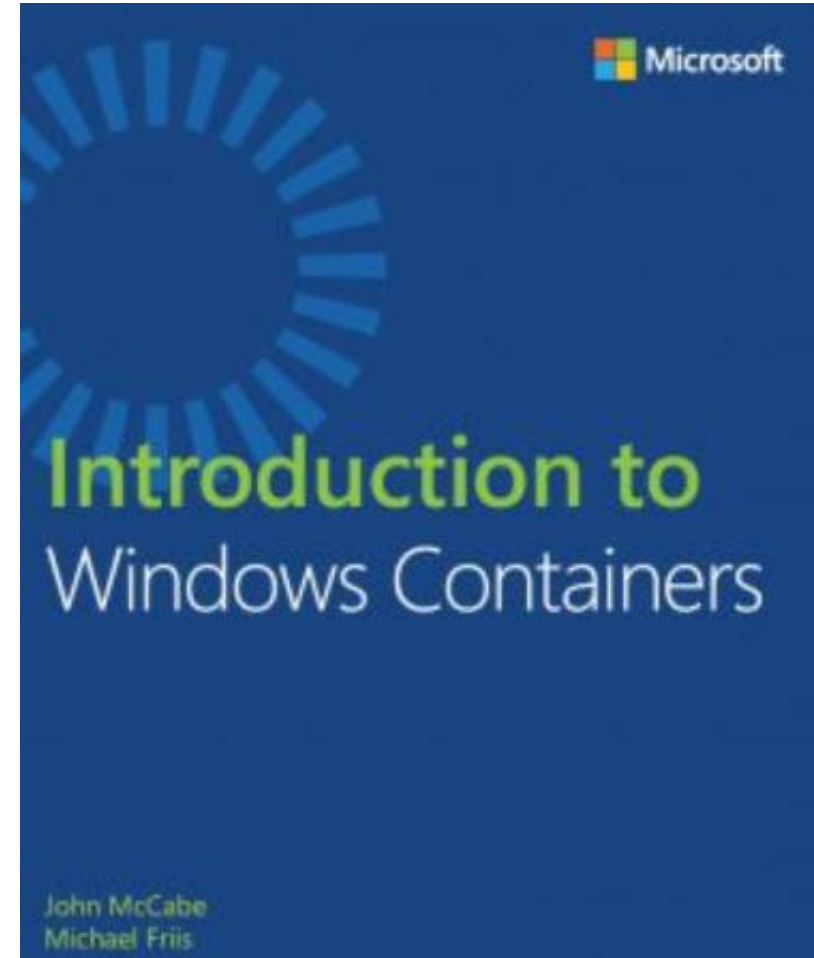
Referenzen



Kostenloses E-Book

Kostenloses E-Book:

https://blogs.msdn.microsoft.com/microsoft_press/2017/08/30/free-ebook-introduction-to-windows-containers/



Weblinks

Docker Kurs auf unserer Website www.gdsbi.de

SQL Server Sample Docker Container:

<https://github.com/Frank-Geisler/SQL-Server-Sample-Docker-Container>

SQL Server Samples auf Docker Hub:

<https://hub.docker.com/u/frankgeisler/>