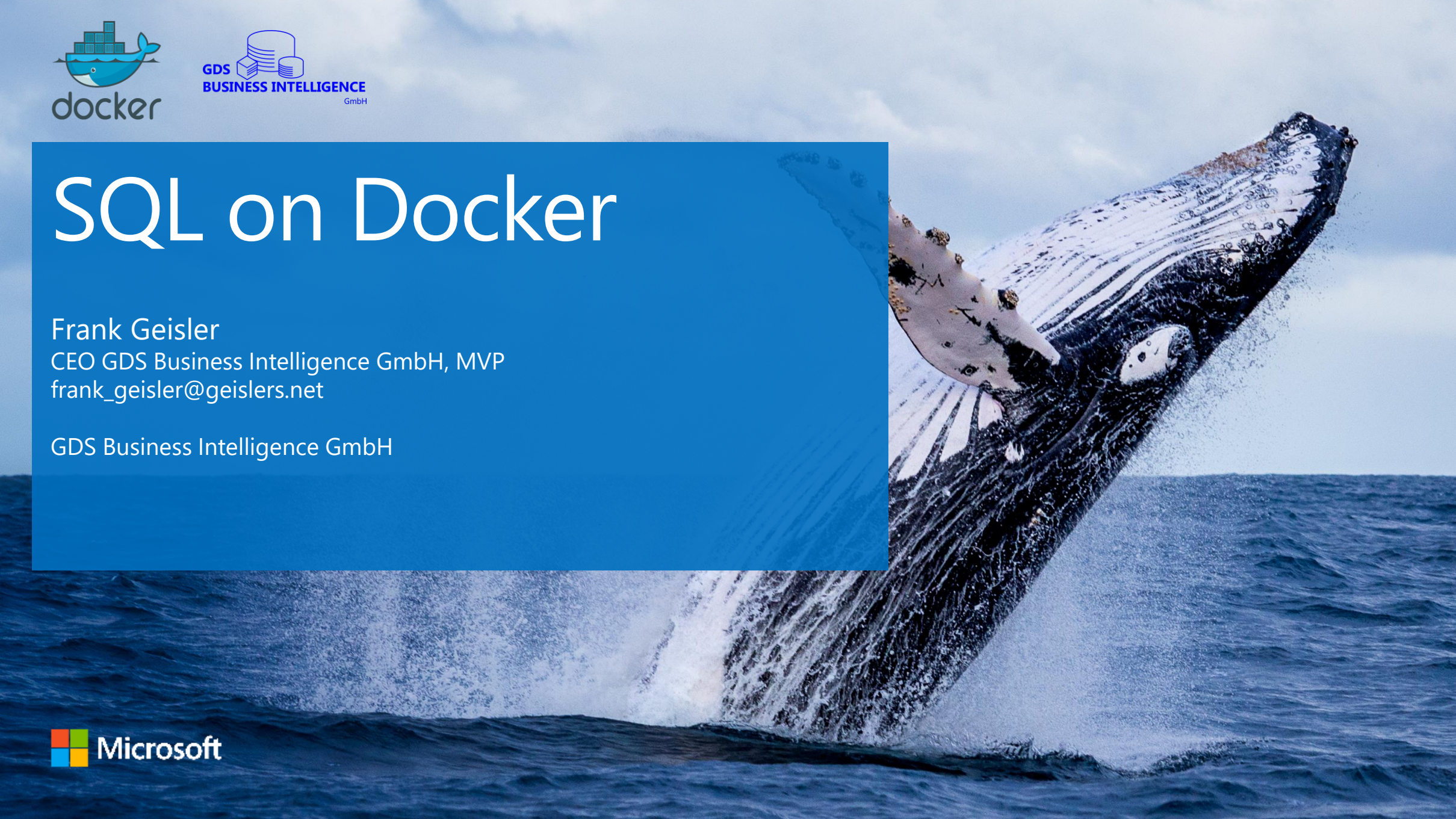


SQL on Docker

Frank Geisler
CEO GDS Business Intelligence GmbH, MVP
frank_geisler@geislers.net

GDS Business Intelligence GmbH


SOLISYON
www.solisyon.de


Zötler Bier
natürlich, sympathisch, frisch


DATAmasterminds


POWER BI SENTINEL
Documentation · Data Lineage · Change Tracking · Backup


ROSEN
empowered by technology


SentryOne


HEDDA.IO


redgate


SHOP APOTHEKE
EUROPE


dbWatch
DATABASE CONTROL

Session Feedback Day
(not optional!)

http://bit.ly/DataGrillen2019Day

Session Feedback Day
(not optional!)

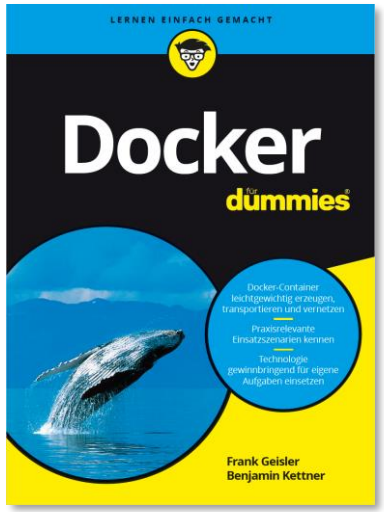
http://bit.ly/DataGrillen2019Day

Event Feedback (not optional!)

http://bit.ly/DataGrillen2019Ever

Frank Geisler

frank_geisler@geislers.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

FRIEND OF redgate 2016

MVP Microsoft Most Valuable Professional

@FrankGeisler

Please Talk Data To Me

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

Ben Weissman
Biml Ben

Tillmann Eitelberg
Mr. T

Frank Geisler
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** TugaIT Visual Studio Code **WDC**

12	2921	1275	13
Episoden	Downloads	Sendeminuten	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

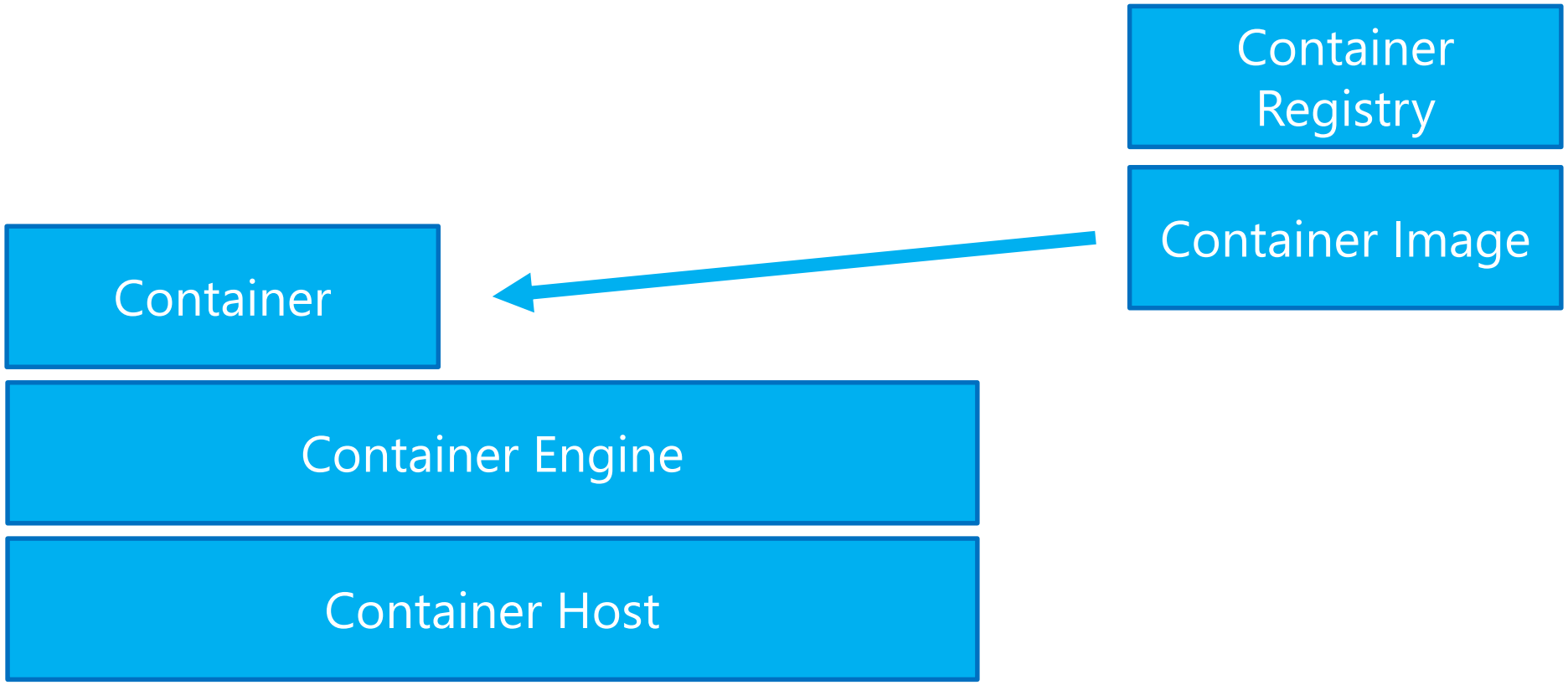
Container kapseln eine Software in einem vollständigen Dateisystem das alles enthält was der Container benötigt um ausgeführt werden zu können: Code, Laufzeitumgebung, System Tools, System Bibliotheken – alles was man auf einem Server installieren kann. Das garantiert, dass die Software immer ausgeführt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

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

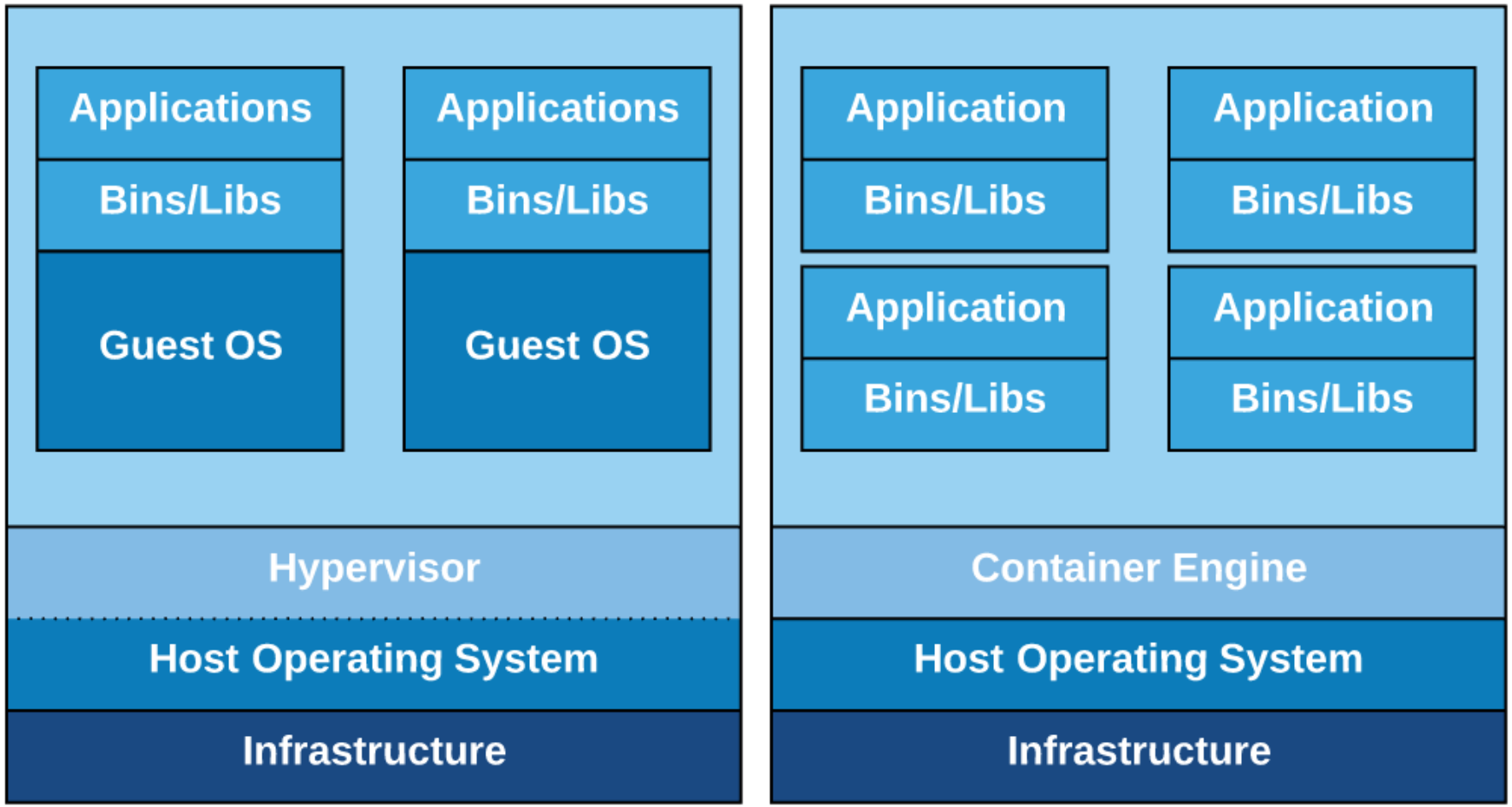
Grundlegende Elemente von Docker

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

Zusammenhang der Elemente



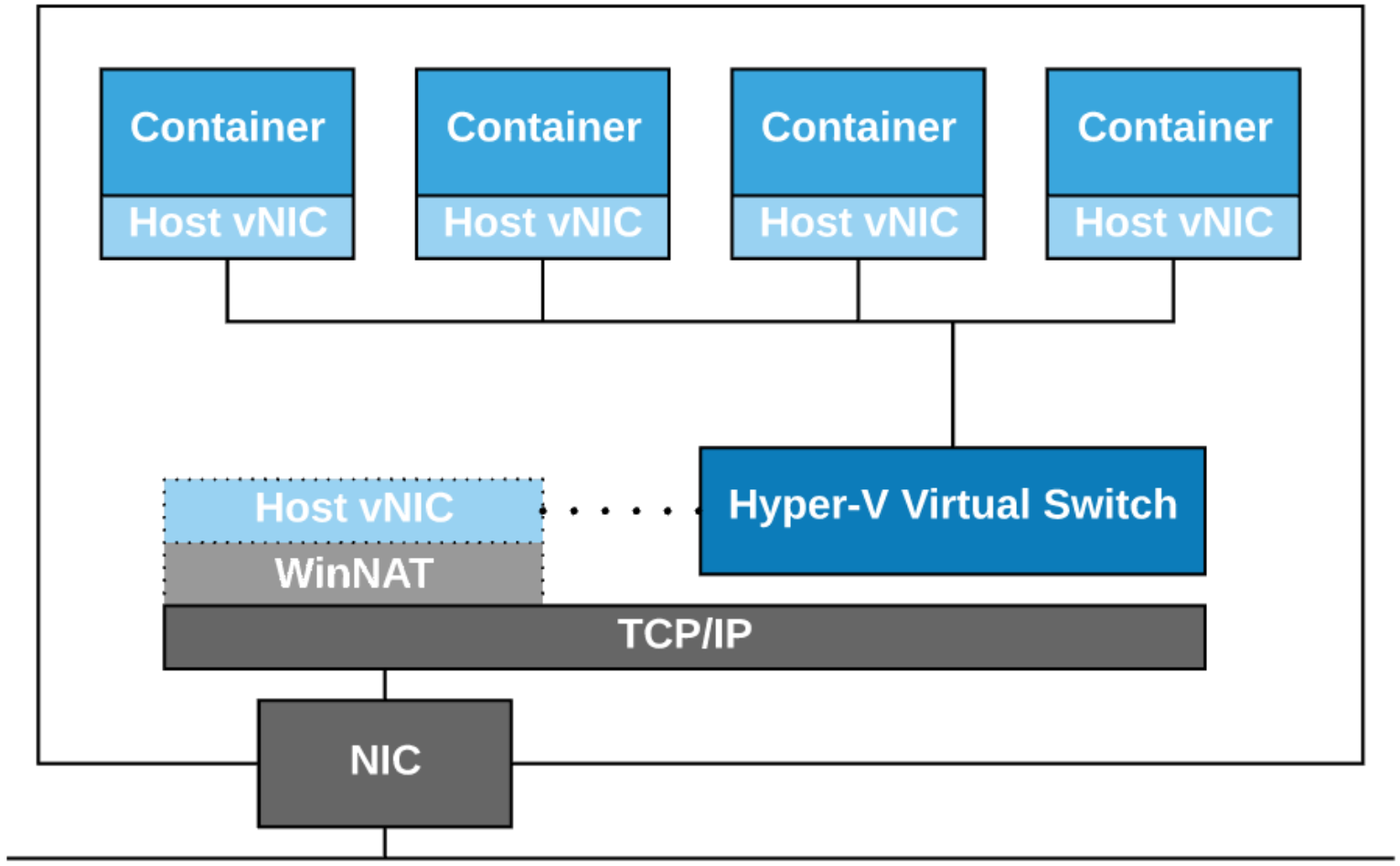
Virtuelle Maschinen vs. Container



Container sind keine VMs !

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

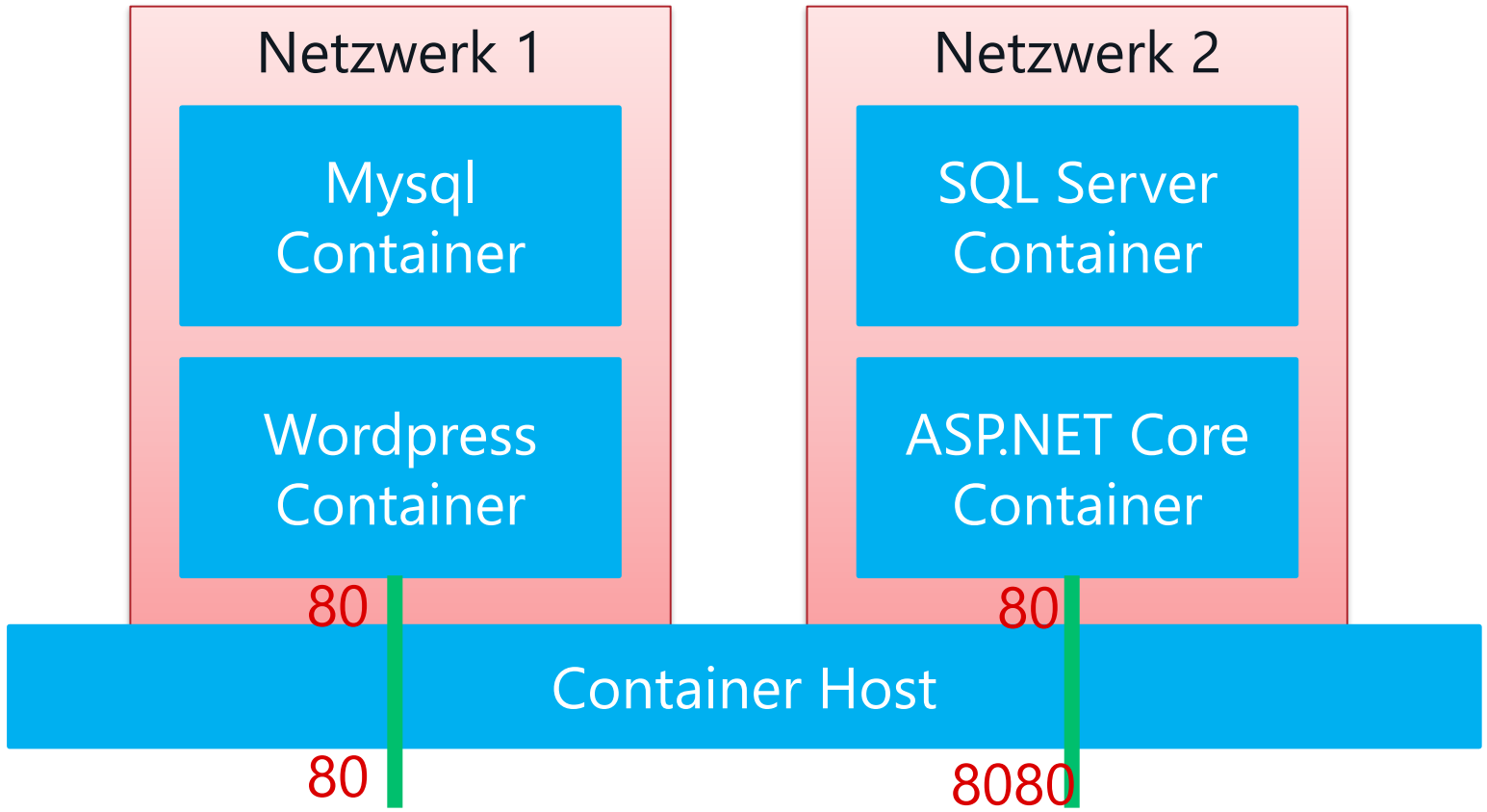
Container Netzwerk



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 eine 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

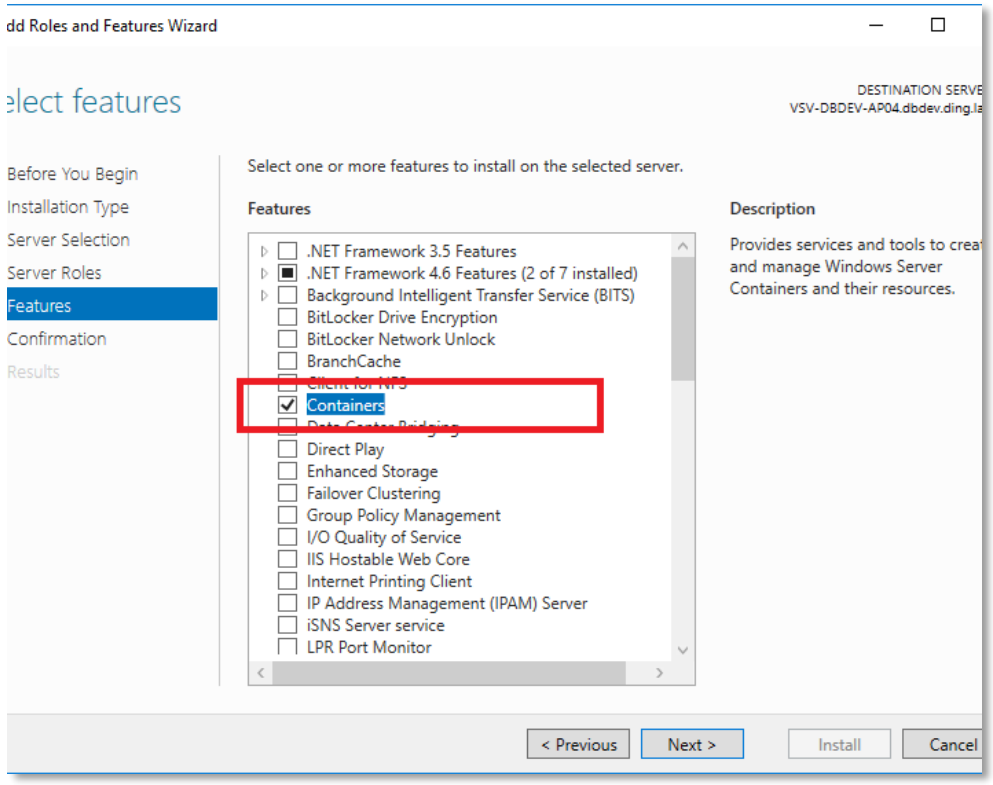
Windows Server Container

- 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 virtutellen 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

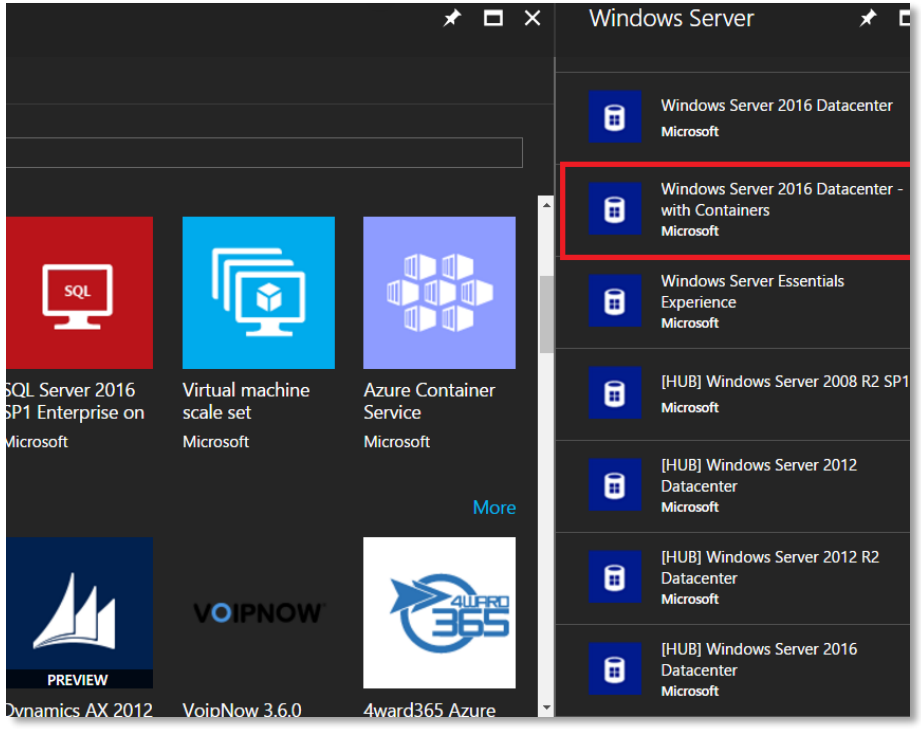
.msi herunterladen

https://store.docker.com/

Azure

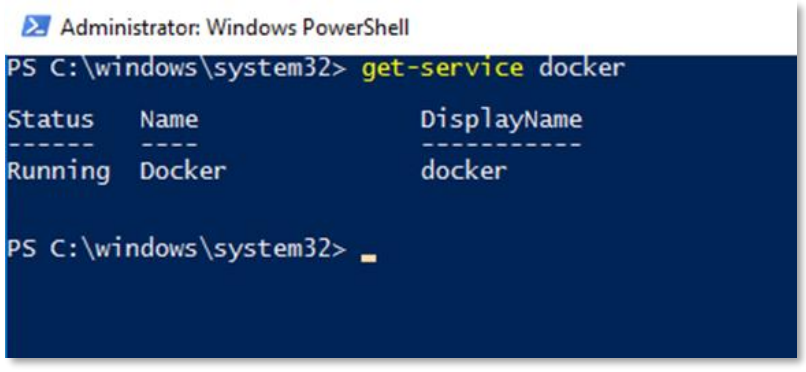
Es gibt viele Möglichkeiten…

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

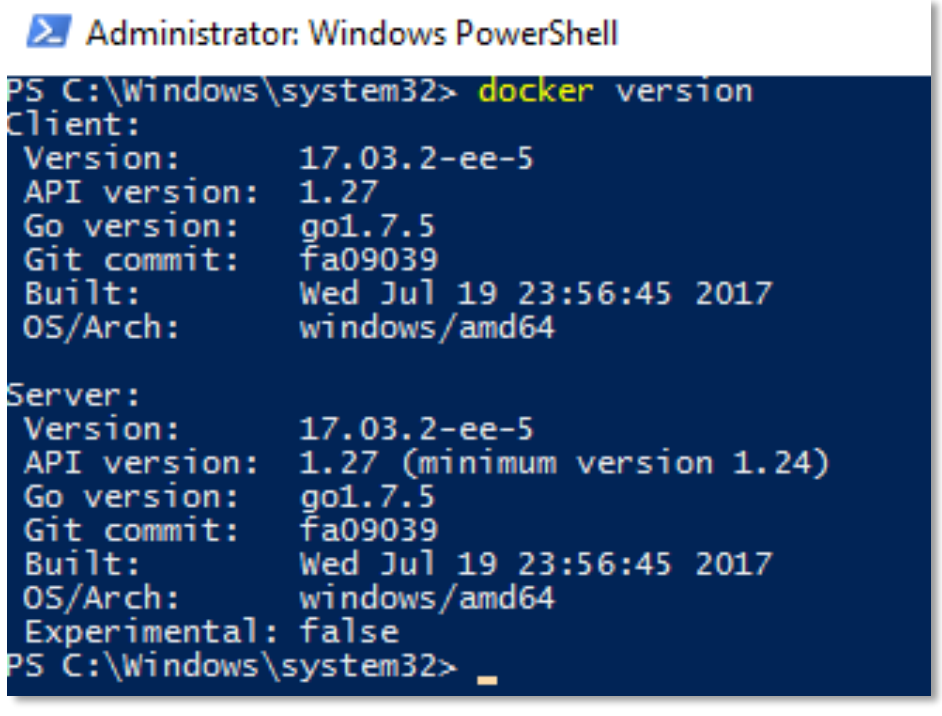


Prüfen ob Docker läuft

Get-Service docker



docker version



Docker mit älterem Windows / SQL Server

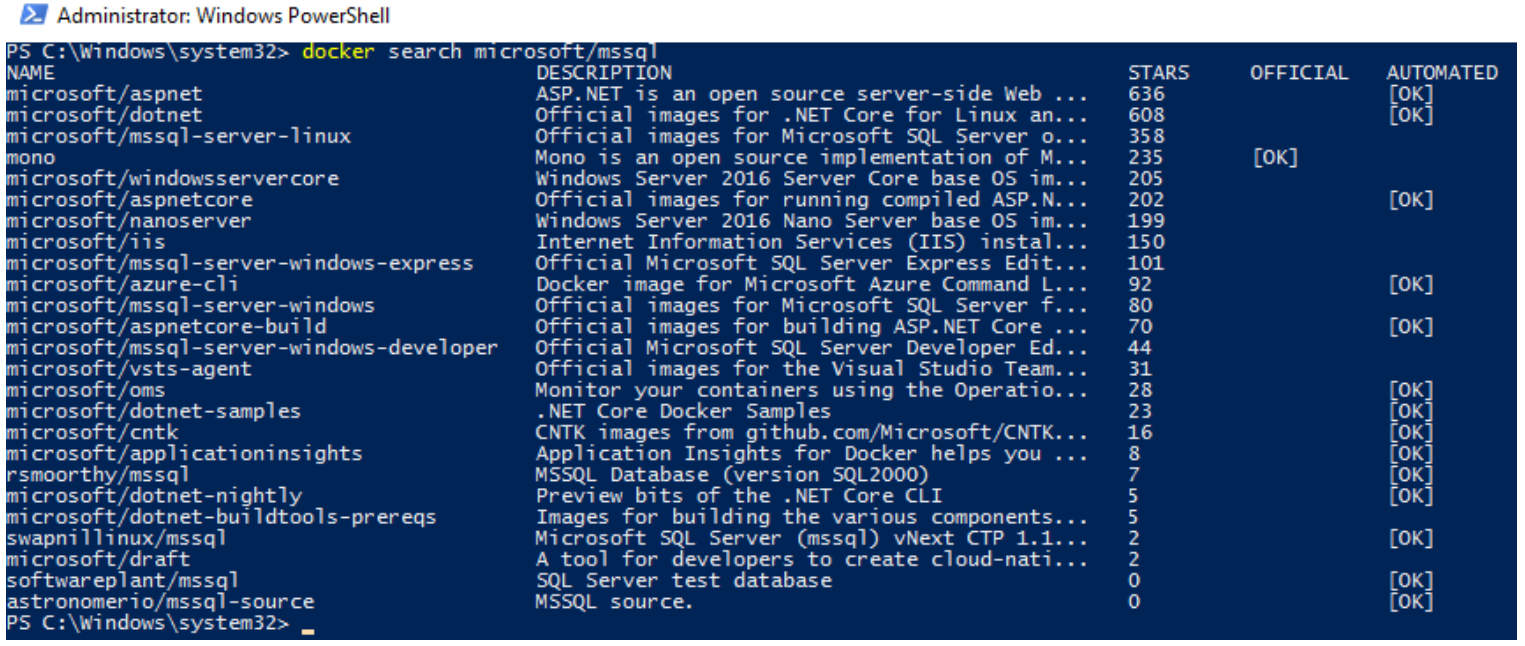
- www.windocks.com
- Eine Portierung des Open Source Projektes von Docker Inc.
- Software unterstützt die Erstellung von Containern auf denen frühere Versionen von SQL Server (2008+) auf Windows Server 2012 ausgeführt werden
- Kostenlose Community Edition verfügbar

  www.windocks.com/leads/add?src=downloadcommunity

Einen SQL Server Container bauen

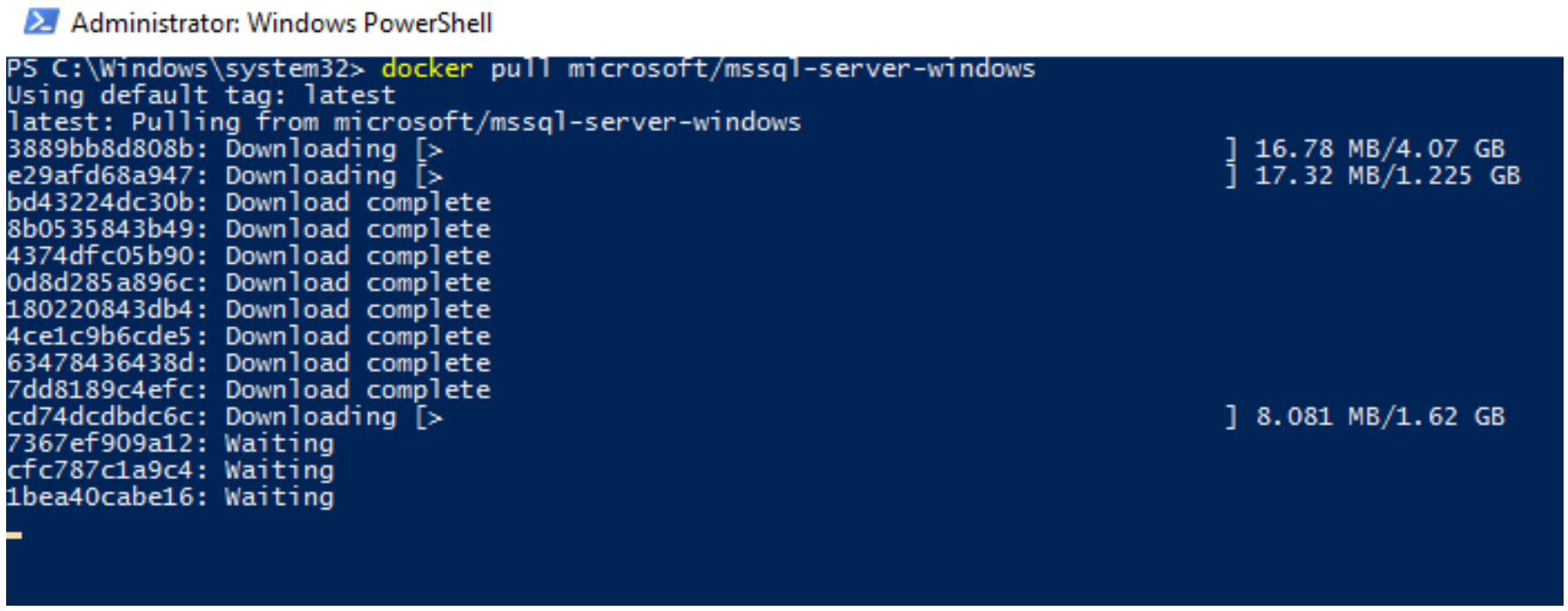
Das Docker Repository durchsuchen

docker search microsoft/mssql



Ein Image herunterladen

docker pull microsoft/mssql-server-linux

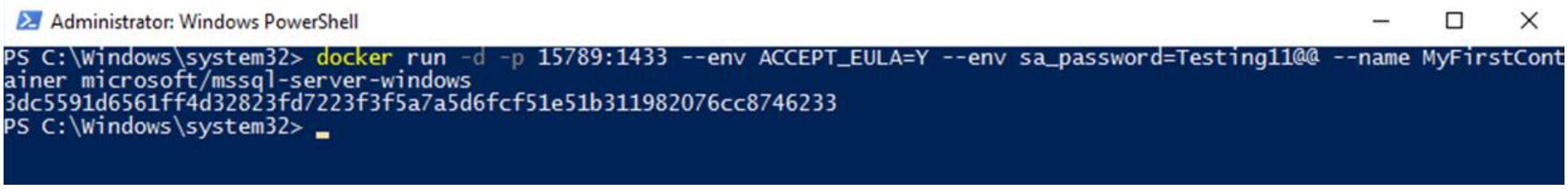


Das Image überprüfen

docker images

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

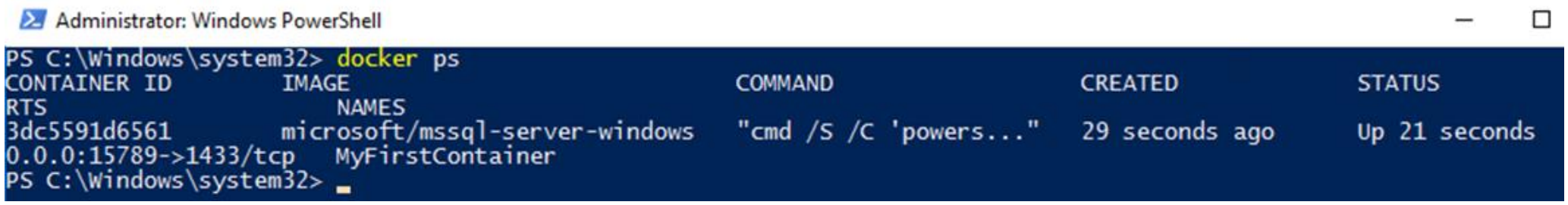


Was passiert eigentlich bei docker run im Hintergrund?

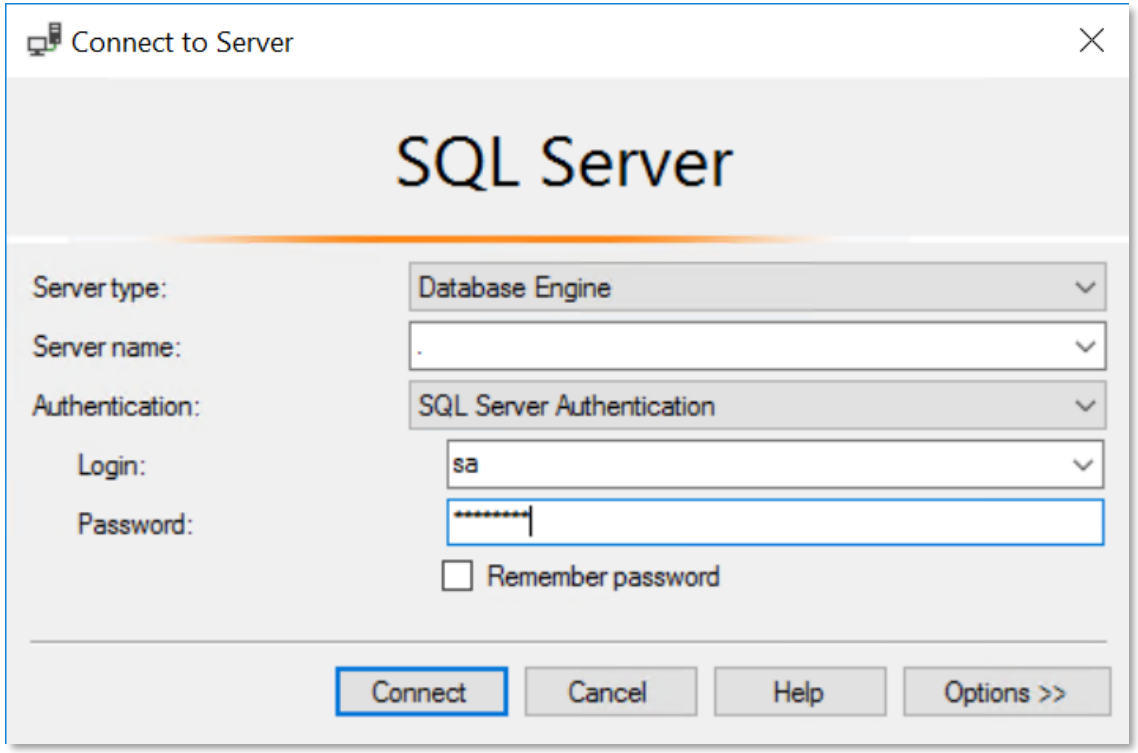
1. Docker schaut ob das Image lokal im cache liegt (in entspr. Version)
2. Wird Image nicht gefunden wird in remote image repository nachgeschaut (Standard: Docker Hub)
3. Lädt die neuste Verison runter (in unserem Fall, da nichts angegeben)
4. Erstellt einen neuen Container auf Basis des Images und 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]



Mit dem SQL Server verbinden



Container überprüfen

docker inspect sqlcontainer

Mit dem Container verbinden

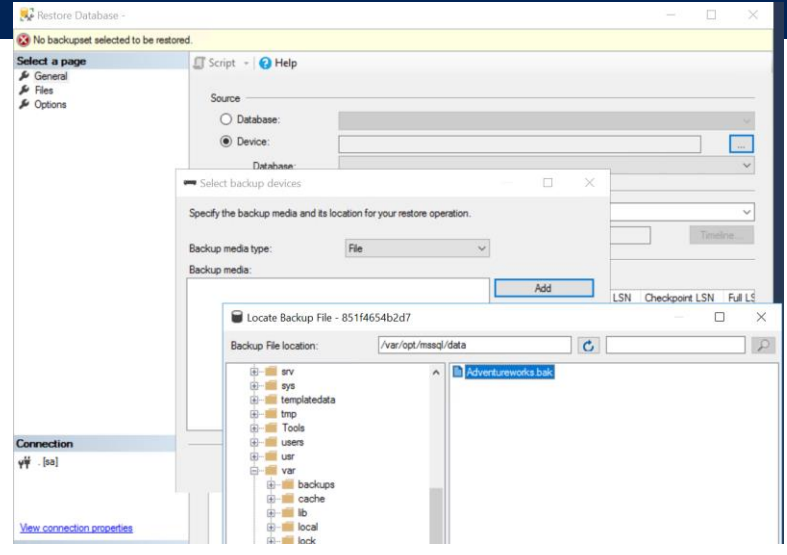
docker exec –it sqlcontainer bash

```
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.bak

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




DEMO
START
EASY STYLE
DISPLAY PROPERTIES
SCALABLE
Research
-Marketing
-Customer
IDEAS
ACTION
PLAN
Tel 1011 3
*Send E-mail
how to do ?
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 Overcomitting der Ressourcen

Ein angepasstes Container Image bauen

Dockerfile

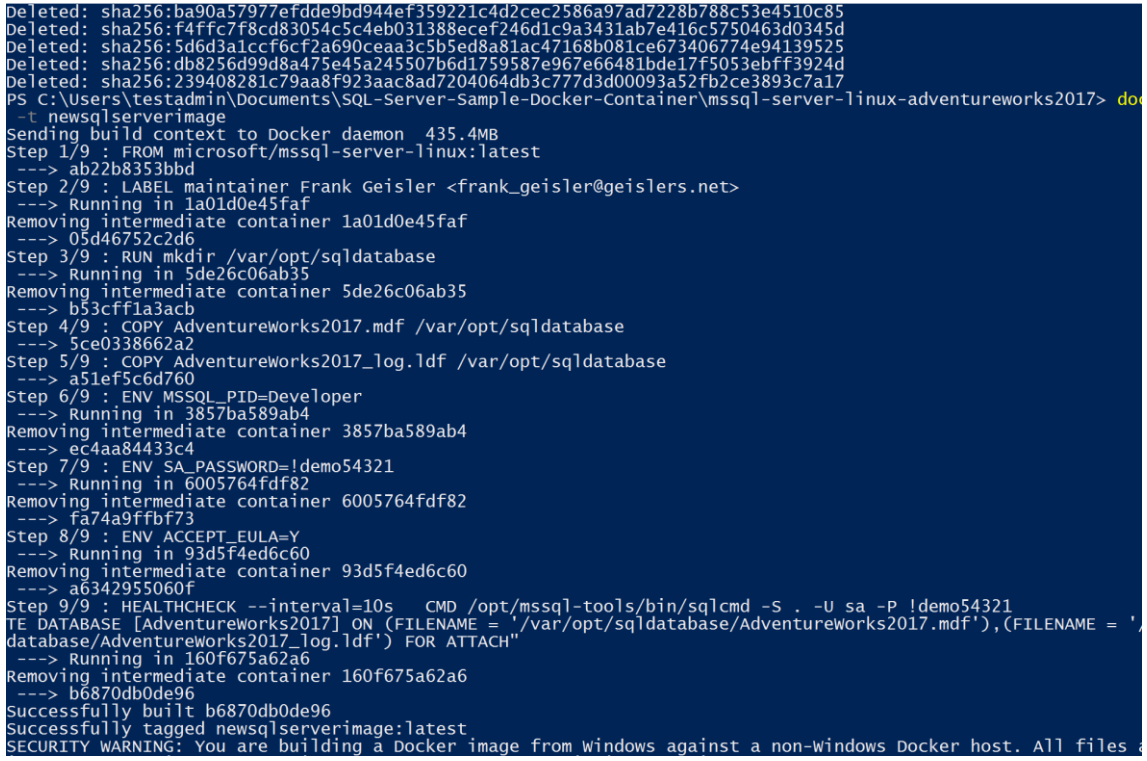
Eine Datei auf dem Docker Host die Befehle enthält die ein angepasstes Image erzeugen.

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 .



Das neue Image überprüfen

docker images

```
PS C:\Users\testadmin\Documents\SQL-Server-Sample-Docker-Container\mssql-server-linux-adventureworks2017> doc
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
newsqlserverimage             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 newsqlservercontainer newsqlserverimage

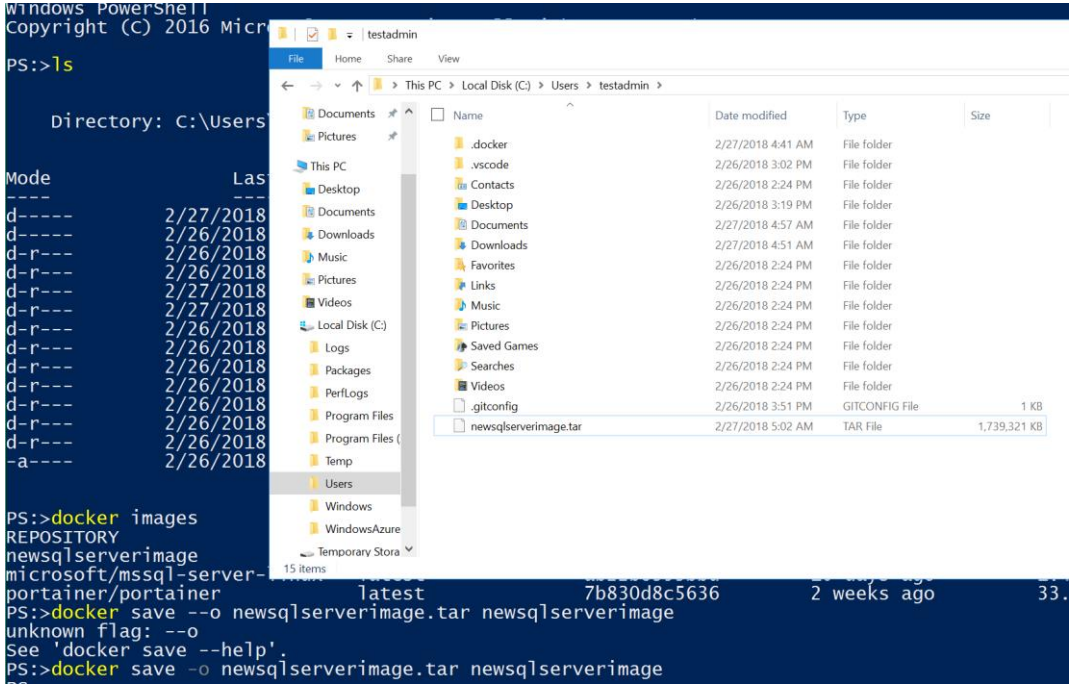
```
PS C:\Users\testadmin\Documents\SQL-Server-Sample-Docker-Container\mssql-server-linux-adventureworks2017> docker conta
er run -d -p 1433:1433 --name neuersqlcontainer newsqlserverimage
89b7b3d96ce33f107f2689ab977d9f46e94dbbba0ee74f72bd4d769f95abfcfa
```


DEMO
START
EASY STYLE
DISPLAY PROPERTIES
SCALABLE
Research
-Marketing
-Customer
IDEAS
ACTION
PLAN
Tel 1011 3
*Send E-mail
how to do ?
Business
Plan

Images exportieren

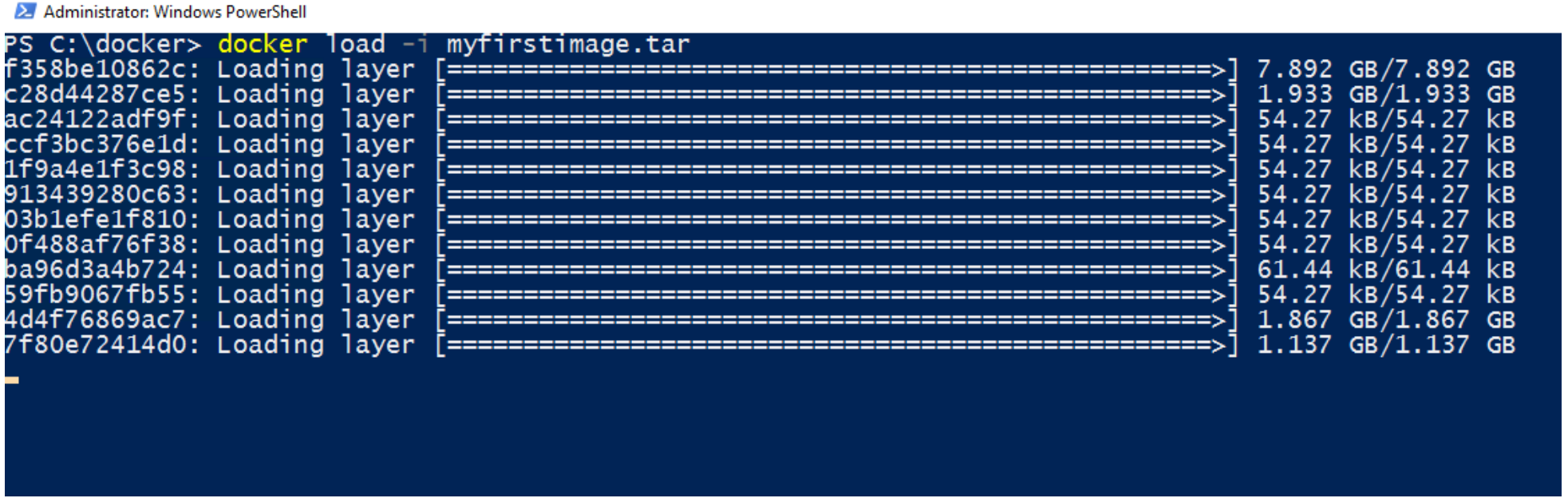
Lokal teilen

docker save -o newsqlserverimage.tar newsqlserverimage



Images importieren

docker load -i newsqlserverimage.tar

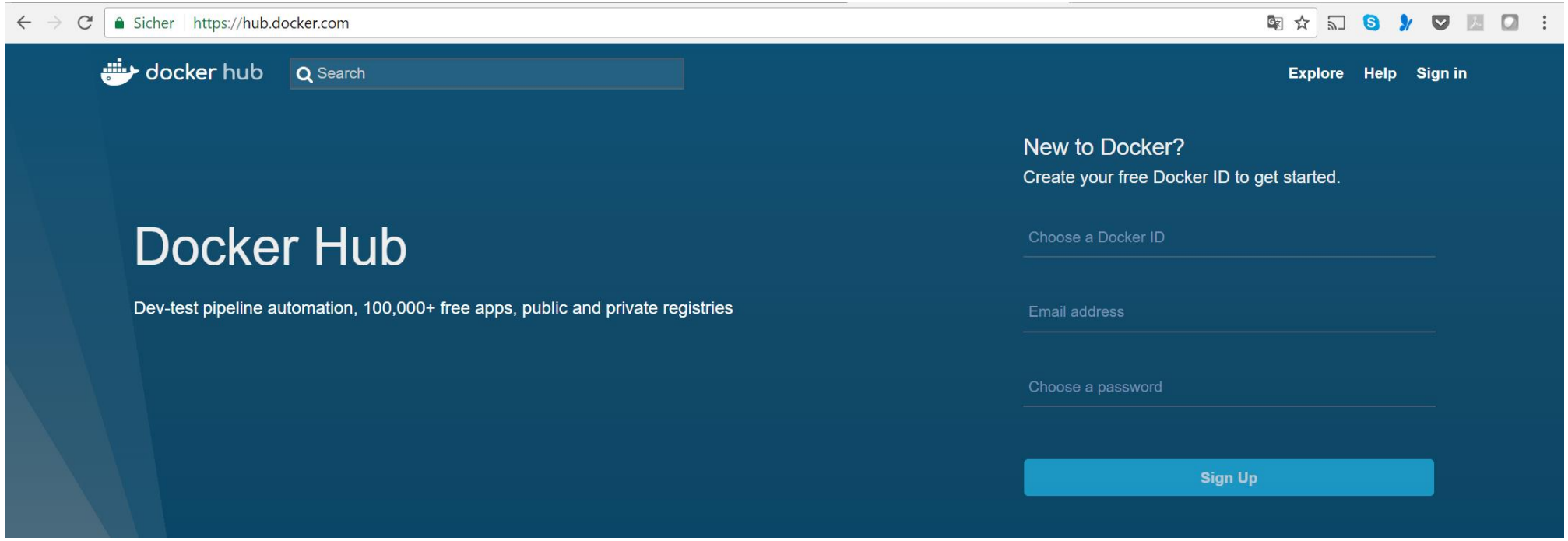



DEMO
START
EASY STYLE
DISPLAY PROPERTIES
SCALABLE
Research
-Marketing
-Customer
IDEAS
ACTION
PLAN
Tel 1011 3
*Send E-mail
how to do ?
Business
Plan

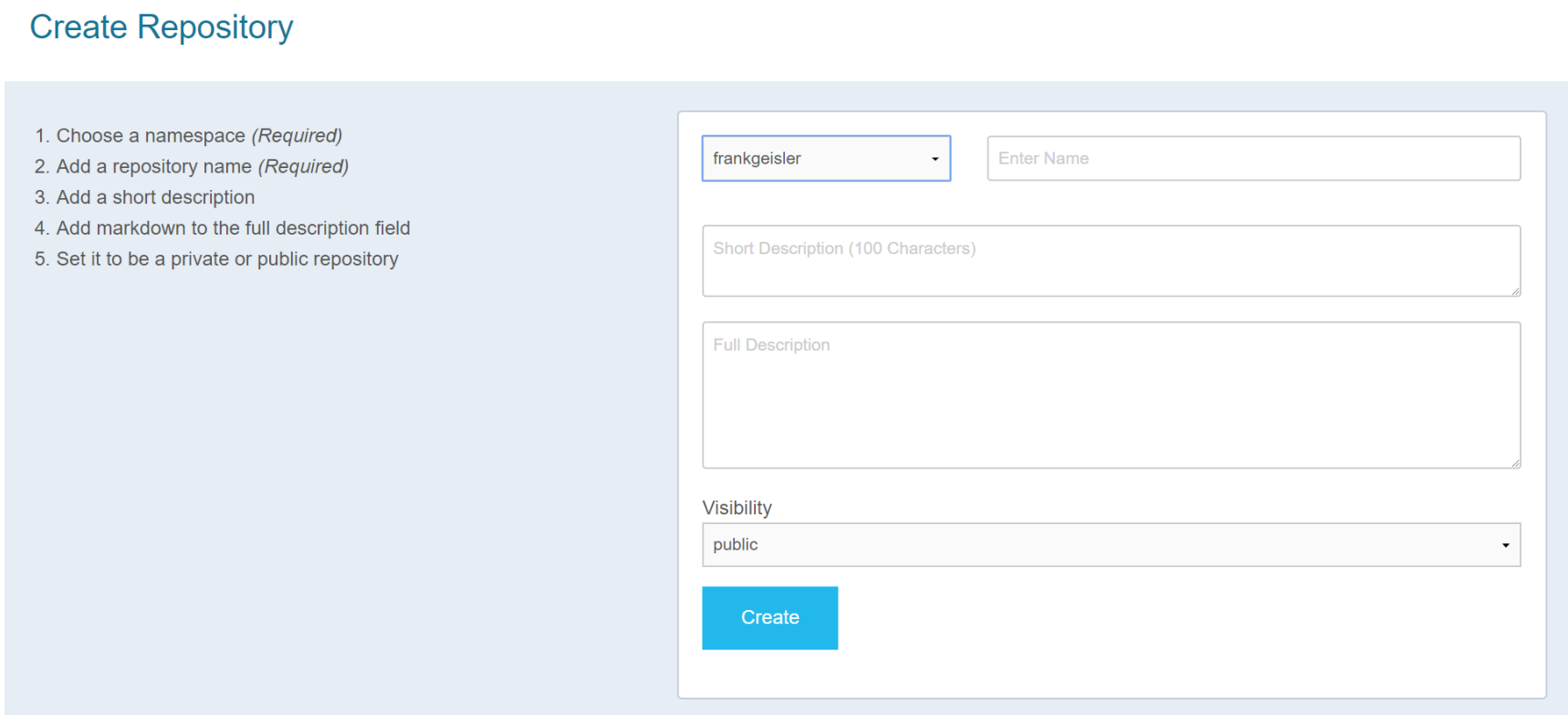
Angepasste Images auf Docker Hub veröffentlichen

Der Docker Hub

https://hub.docker.com



Ein Repository erstellen



Ein Image taggen

docker tag newsqlserverimage frankgeisler/newsqlserverimage:v1

```
PS:>docker tag newsqlserverimage frankgeisler/newsqlserverimage:v1
PS:>
```

In Docker Hub einloggen

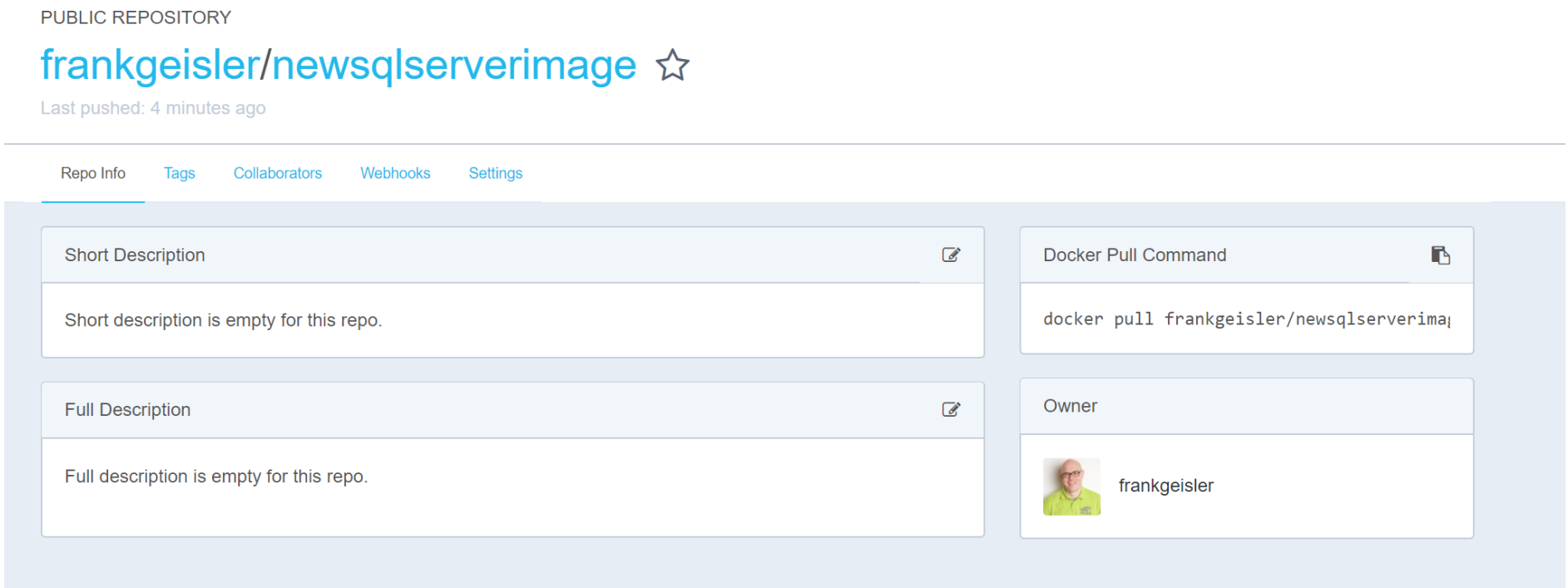
docker login

```
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:/
ub.docker.com to create one.
Username: frankgeisler
Password:
Login Succeeded
PS:>
```

Docker Image hochladen

```
PS:>docker push frankgeisler/newsqlserverimage:v1
The push refers to repository [docker.io/frankgeisler/newsqlserverimage]
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



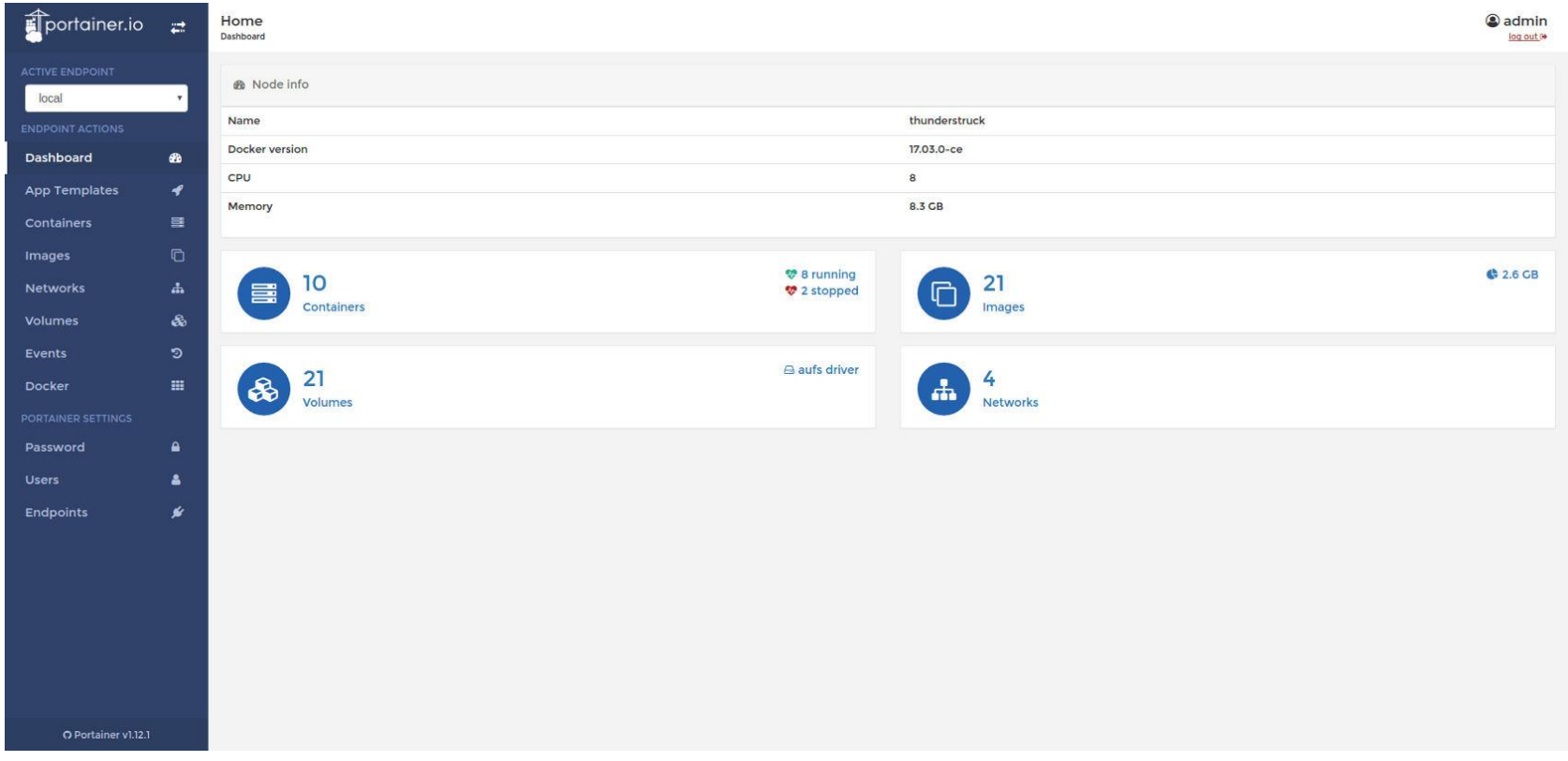

DEMO
START
EASY STYLE
DISPLAY PROPERTIES
SCALABLE
Research
-Marketing
-Customer
IDEAS
ACTION
PLAN
Tel 1011 3
*Send E-mail
how to do ?
Business
Plan

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


portainer.io
ACTIVE ENDPOINT
local
ENDPOINT ACTIONS
Dashboard
App Templates
Containers
Images
Networks
Volumes
Events
Docker
PORTAINER SETTINGS
Password
Users
Endpoints
Portainer v1.12.1
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


DEMO
START
EASY STYLE
DISPLAY PROPERTIES
SCALABLE
Research
-Marketing
-Customer
IDEAS
ACTION
PLAN
Tel 1011 3
*Send E-mail
how to do ?
Business
Plan

Einsatzszenarien

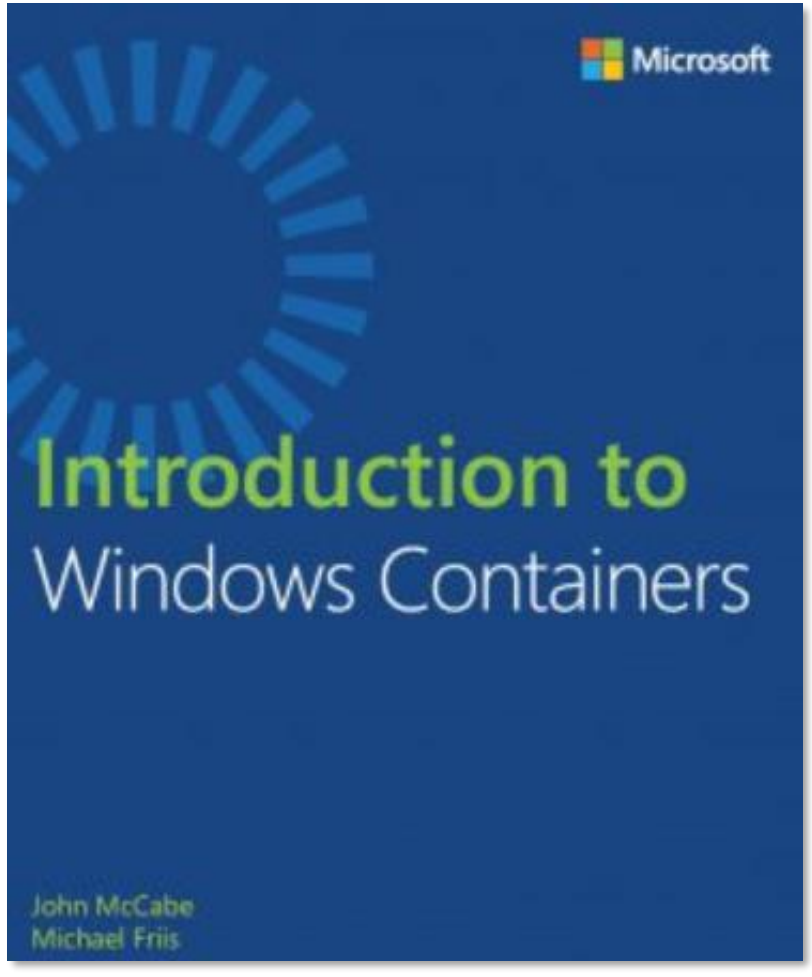
Einsatzszenarien

- 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/