

Erste Anwendungen mit Docker und Windows Server 2019

Übersetzung von: <https://blog.sixeyed.com/getting-started-with-docker-on-windows-server-2019/>

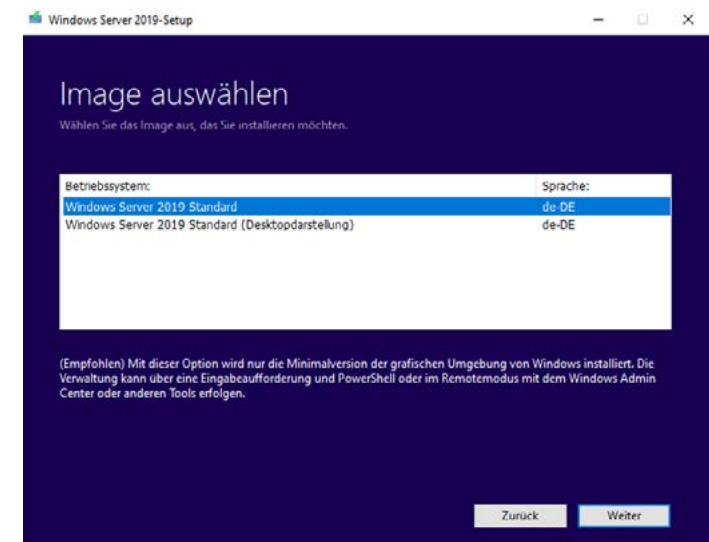
Windows Server 2019 ist der aktuelle langfristige Windows Server und seit 2018 verfügbar! Neu sind sehr nützliche Tools, um Docker Windows Container zu benutzen. Stefan Scherer hat diese Neuheiten auf seinem Blog veröffentlicht und kann in seinem Artikel „What’s new for Docker on Windows Server 2019“ nachgelesen werden. Ebenfalls empfehlenswert ist das Buch „Docker on Windows“ von Elton Stonemann, das sich ausschließlich mit Windows Server 2019 beschäftigt.

Um mit Docker zu starten benötigst Du einen Windows Server, auf dem der Container-Prozess direkt auf dem Host-OS laufen kann. Analog können die selben Docker-Images, -Files und Befehle unter Windows 10 benutzt werden, hierbei gibt es jedoch einen zusätzlichen Virtualisierungs-Overhead (also eine erhöhte Arbeitsbelastung des Computers der durch die Virtualisierung entsteht), weshalb es sich empfiehlt einen Windows Server zu verwenden. Falls ein solcher Server nicht vorhanden/gewollt ist, sollte **Docker Desktop** für die Verwendung von Docker unter Windows 10 verwendet werden. In diesem Artikel erklären wir Dir, wie Du vorgehen musst, um mit der neusten Version des Windows Servers zu starten und Docker Container zu nutzen.

Installiere Windows Server 2019

Lade die ISO-Datei herunter, um Windows Server 2019 zu installieren. Diese ist entweder in Deinem Visual Studio Abonnement enthalten oder alternativ empfehlen wir Dir eine 180-Tage-Testversion zu nutzen. Virtuelle Maschinen auf denen Windows Server 2019 bereits installiert ist sind ebenfalls bei Azure erhältlich.

Der Installationsprozess ist der gleiche wie bei allen vorigen Versionen – Starte eine VM von der ISO und der Installationsprozess beginnt. Wir bevorzugen an dieser Stelle die Core-Installation ohne grafische Oberfläche, alternativ steht aber auch die **Desktop Experience** Version zur Verfügung.



Der SQL Server 2019 wurde von uns auf eine Hyper-V VM unter Windows 10 installiert, wobei die VM auf einer externen SSD abgelegt wurde. Das Setup ist innerhalb von ein paar Minuten fertig und läuft sehr schnell, und das sogar mit nur 4 GB RAM.

Hinweis: Die ISO kann auch dazu verwendet werden, eine vorhandene Installation von Windows Server auf Windows Server 2019 zu updaten.

Verbindung zum Server

Um eine Verbindung zu einem Windows Server aufzubauen, ist eine Kommando-Zeile vollkommen ausreichend. Beim ersten Zugriff musst Du ein Passwort für den primären **Administrator**-Account festlegen. Danach konfigurieren wir gerne PowerShell als primäre Kommando-Zeile, sodass Du bei jedem RDP-Zugriff eine PowerShell Session mit folgender Eingabe starten kannst:

```
Set-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon' -name Shell -Value 'PowerShell.exe -noExit'
```

Konfiguration der Windows Features

Um Container zu starten, musst Du das **Containers** Feature aktivieren. Für Virtuelle Maschinen die sich nicht in einer Produktionsumgebung befinden kann es helfen, den **Windows Defender** auszuschalten um so ein paar CPU-Zyklen einzuparen und die Performanz des Systems zu erhöhen. Nach diesen Schritten kannst Du mit folgender Befehlseingabe neustarten:

```
Install-WindowsFeature -Name Containers  
Uninstall-WindowsFeature Windows-Defender  
Restart-Computer -Force
```

Konfiguration der Windows Features

Überprüfe, ob die Software in der aktuellsten Version läuft, danach kannst Du automatische Updates deaktivieren, um nur noch Updates durchzuführen, wenn Du es wirklich möchtest. Für eine Windows Server Core Installation (also eine Installation ohne GUI) geschieht dies mit Hilfe des Befehls **sconfig**. Folgende Optionen müssen gewählt werden:

- × Option **5**, um Windows Updates manuell zu installieren
- × Option **7**, um Remote Desktop Zugriff auf den Server zu erlauben
- × Option **6**, um alle Updates zu downloaden und zu installieren

Nach diesen Schritten ist Dein Computer bereit für die Installation von Docker!

Installation von Docker auf dem Windows Server

Die Lizenzierung von Windows Server beinhaltet ebenfalls die Lizenzen für Docker Enterprise. Die Enterprise Version mit Produktionsunterstützung für Container von Microsoft und Docker können also einfach heruntergeladen werden.

Die aktuelle Docker Enterprise Version ist 19.03, die mit PowerShell heruntergeladen werden kann. Nutze hierfür folgenden Befehl:

```
Install-Module -Name DockerMsftProvider -Repository PSGallery -Force
```

```
Install-Package -Name docker -ProviderName DockerMsftProvider -Force  
-RequiredVersion 19.03
```

Dieser Befehl erstellt Docker als Windows Service, was Du für den Start benötigst:

```
Start-Service docker
```

Windows Base Images beziehen

Alle Docker Container, die Du auf Windows Server 2019 öffnest, basieren auf dem Windows Server Core oder Nano Server. Du benötigst beides – achte hier darauf, dass die Base Images in der Microsoft Container Registry (MCR) registriert sind:

```
docker image pull mcr.microsoft.com/windows/servercore:ltsc2019
```

```
docker image pull mcr.microsoft.com/windows/nanoserver:1809
```

Diese images sind nahezu vergleichbar mit dem Windows Server 2016. Allerdings ist der Download-Speicher des Windows Server Cores mit 1,5 GB über 10 GB kleiner als zuvor und der Speicher des Nano Servers ist von 1 GB auf nur 90 MB gesunken!

[Optional] Die .NET Core Images beziehen

Das .NET Core-Team hat mit dem Launch von Windows Server 2019 Versionen von SDK und Runtime Images veröffentlicht. Du kannst diese downloaden und deine .NET Core Anwendungen direkt verwenden. Zusätzlich stehen die .NET Framework SDK und ASP.NET Images zur Verfügung.

```
docker image pull mcr.microsoft.com/dotnet/core/aspnet:3.0
```

```
docker image pull mcr.microsoft.com/dotnet/core/sdk:3.0.100
```

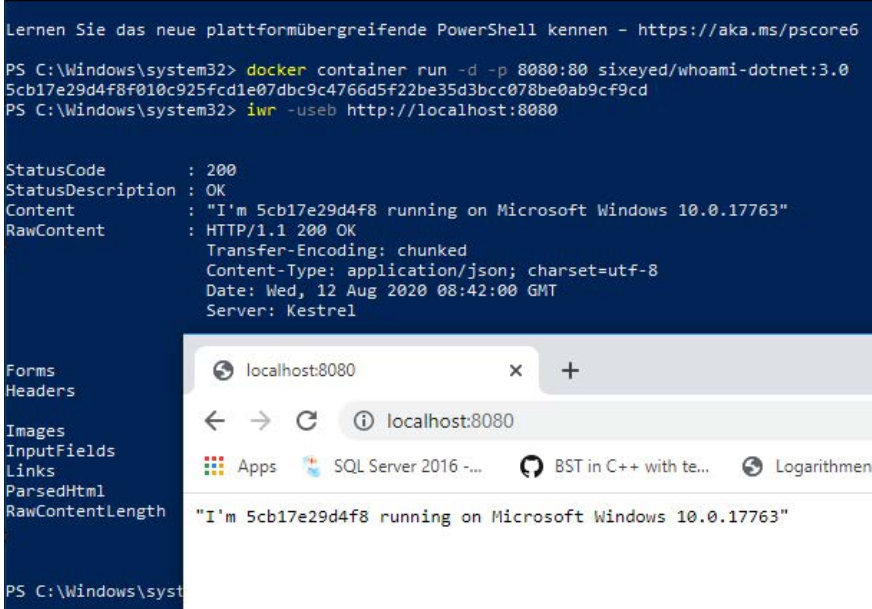
Die vorigen Docker Images sind immer noch auf Docker Hub gelistet und werden direkt von Microsofts eigener Image Registry (MCR) hochgeladen.

Probiere es aus!

Es wurde eine aktualisierte Version von .NET Core **whoami** image veröffentlicht, mit der Du ASP.NET Core 3.0 auf einem Windows Server Core 2019 Container testen kannst:

```
docker container run -d -p 8080:80 sixeyed/whoami-dotnet:3.0
```

Eine dieser Verbesserungen für Docker auf Windows Server 2019 ist, dass sog. **loopback Adressen** jetzt funktionieren. Der Container kann mit der **localhost** Adresse auf dem Server unter dem selben veröffentlichten Port einer externen Maschine besucht werden.



The screenshot shows a PowerShell terminal window with the following commands and output:

```
Lernen Sie das neue plattformübergreifende PowerShell kennen - https://aka.ms/pscore6
PS C:\Windows\system32> docker container run -d -p 8080:80 sixeyed/whoami-dotnet:3.0
5cb17e29d4f8f010c925fcd1e07dbc9c4766d5f22be35d3bcc078be0ab9cf9cd
PS C:\Windows\system32> iwr -useb http://localhost:8080
StatusCode      : 200
StatusDescription : OK
Content          : "I'm 5cb17e29d4f8 running on Microsoft Windows 10.0.17763"
RawContent      : HTTP/1.1 200 OK
                  Transfer-Encoding: chunked
                  Content-Type: application/json; charset=utf-8
                  Date: Wed, 12 Aug 2020 08:42:00 GMT
                  Server: Kestrel
```

Below the terminal, a web browser window is open to localhost:8080, displaying the same JSON response: "I'm 5cb17e29d4f8 running on Microsoft Windows 10.0.17763".

Swarm Mode

Es gibt viele Verbesserungen von Windows Server 2016 zu Windows Server 2019, eine davon ist der Support von [ingress networking](#) für den Docker Swarm Mode. Der Vorteil des Swarm Modes ist das gleichzeitige Laufen verschiedener Container auf einem Server, die auf einem Port gelistet sind. Dabei kann Docker einen Ausgleich (Load Balance) zwischen den bestehenden und neuen Containern gewährleisten.

Mehr Details zum Managen von **Load Balancing** und die Anpassung in das Docker Swarm Mode Cluster kannst du in dem folgenden Artikel nachlesen: [Managing Load Balancing and Scale in Docker Swarm Mode Clusters](#).

Wechsel mit Deinem Server zu einem single-node (Einzelknoten) swarm mit folgendem Befehl:

```
docker swarm init --advertise-addr 127.0.0.1
```

Starte nun die **whoami** app als swarm service mit mehreren Kopien und dem folgenden veröffentlichten Port:

```
docker service create --publish 8070:80 `--replicas 5 ` sixeyed/whoami-dotnet:nanoserver-1809
```

Docker kann Anfragen innerhalb von fünf Containern ausgleichen, die das VM extern erreichen.

Das war noch nicht alles!

Windows Server 2019 ist eine Entwicklung der Container Funktionalität von Docker. Windows Server 2016 ist immer noch perfekt für eine Produktion, allerdings ist die Version 2019 geeigneter für die reibungslose Gleichstellung mit Linux Containern – mit der Version 2016 ist es um einiges schwieriger, ein perfektes Ergebnis zu erreichen.

Außerdem interessant ist der Artikel [Getting Started with Kubernetes on Windows](#). Seit 2018 können die Funktionen auf einem geteilten Linux-Windows Kubernetes Cluster angewendet werden.