



zenon
by COPA-DATA

zenon Handbuch Docker

v.8.20



COPA-DATA

© 2020 Ing. Punzenberger COPA-DATA GmbH

Alle Rechte vorbehalten.

Die Weitergabe und Vervielfältigung dieses Dokuments ist - gleich in welcher Art und Weise - nur mit schriftlicher Genehmigung der Firma COPA-DATA gestattet. Technische Daten dienen nur der Produktbeschreibung und sind keine zugesicherten Eigenschaften im Rechtssinn. Änderungen - auch in technischer Hinsicht - vorbehalten.

Inhaltsverzeichnis

1	Willkommen bei der COPA-DATA Hilfe	4
2	Docker	4
3	Grundlagen	5
4	Terminologie	5
5	Docker for Windows.....	6
5.1	Installation und Konfiguration	6
5.2	Docker Image für zenon Runtime beziehen aus Docker Hub	8
5.3	Container mit zenon Runtime erstellen.....	8
5.4	zenon Runtime für Docker konfigurieren.....	9
5.4.1	Runtime-Docker mit Namensauflösung	12
5.4.2	zenon-Standard-Ports.....	14
5.5	Befehle für Docker (Auswahl).....	15

1 Willkommen bei der COPA-DATA Hilfe

ZENON VIDEO-TUTORIALS

Praktische Beispiele für die Projektierung mit zenon finden Sie in unserem YouTube-Kanal (https://www.copadata.com/tutorial_menu). Die Tutorials sind nach Themen gruppiert und geben einen ersten Einblick in die Arbeit mit den unterschiedlichen zenon Modulen. Alle Tutorials stehen in englischer Sprache zur Verfügung.

ALLGEMEINE HILFE

Falls Sie in diesem Hilfekapitel Informationen vermissen oder Wünsche für Ergänzungen haben, wenden Sie sich per E-Mail an documentation@copadata.com.

PROJEKTUNTERSTÜTZUNG

Unterstützung bei Fragen zu konkreten eigenen Projekten erhalten Sie vom Customer Service, den Sie per E-Mail an support@copadata.com erreichen.

LIZENZEN UND MODULE

Sollten Sie feststellen, dass Sie weitere Module oder Lizenzen benötigen, sind unsere Mitarbeiter unter sales@copadata.com gerne für Sie da.

2 Docker

Docker ist der Markenname einer freien Software zur Isolierung von Anwendungen mit Containervirtualisierung. COPA-DATA ermöglicht die Verwendung der zenon Runtime in einem Docker Container. Dazu benötigen Sie Docker für Windows.

3 Grundlagen

Docker vereinfacht die Bereitstellung von Anwendungen. Im Gegensatz zur virtuellen Maschine hat ein Docker Container kein eigenes Betriebssystem. Es wird direkt das Betriebssystem des Hosts genutzt.

Alle Abhängigkeiten einer Anwendung werden in einem sogenannten Docker **Image** abgebildet. Dieses enthält alle nötigen Pakete und lässt sich einfach als Datei transportieren sowie installieren. Container gewährleisten die Trennung und Verwaltung der auf einem Rechner genutzten Ressourcen. Sie werden lokal auf einem Rechner ausgeführt. Erstellte Container bleiben beim Beenden der Anwendung lokal erhalten und werden nicht entfernt. Falls die Entfernung gewünscht ist, kann der Container bereits beim Aufruf einen Parameter zum automatischen Löschen beim Beenden erhalten.

Docker kann mit Linux und Windows verwendet werden. Für die zenon Runtime wird Docker für Windows benötigt.

DOCKER ENGINE

Die Docker Engine stellt den Zugriff auf den Kernel des Host-Betriebssystems sicher und ist in der Lage, Container zu erstellen, zu starten und zu stoppen. Da ein Docker Image auf dem Host-Betriebssystem aufgesetzt wird, ist ein Docker Image sehr viel kleiner als eine virtuelle Maschine. Auf jedem Rechner, auf dem eine Docker Engine installiert ist, lassen sich Docker-Container ausführen.

4 Terminologie

Fachbegriffe, die in dieser Dokumentation verwendet werden:

Term	Beschreibung
Container:	Aktive Instanz eines Images. Der Container wird also gerade ausgeführt und ist beschäftigt. Sobald der Container kein Programm ausführt oder mit seinem Auftrag fertig ist, wird der Container automatisch beendet.
Container-ID:	Eindeutige ID eines Containers. Bei der Ansprache eines Containers genügt auch ein Teil der ID, solange dieser Teil eindeutig einem Container zuordenbar ist.
Docker Engine:	Stellt den Zugriff auf den Kernel des Host-Betriebssystems sicher und ist in der Lage, Container zu erstellen, zu starten und zu stoppen. Auch als Docker Daemon bezeichnet.

Term	Beschreibung
Docker Hub:	Online-Dienst, der eine Registry für Docker-Images und Repositories beinhaltet.
Image:	Speicherabbild eines Containers. Ein Image besteht aus mehreren Layern, welche schreibgeschützt sind und somit nicht verändert werden können. Ein Image ist portabel, kann in Repositories gespeichert und mit anderen Nutzern geteilt werden. Aus einem Image können immer mehrere Container gestartet werden.
Registry:	<p>Dient der Verwaltung von Repositories. Zum Beispiel Docker Hub.</p> <p>Die Registry teilt sich in einen öffentlichen und einen privaten Teil.</p> <p>Im öffentlichen Teil kann jeder Nutzer seine selbst erstellten Images hochladen und damit anderen Nutzern zur Verfügung stellen. Außerdem gibt es mittlerweile offizielle Images, z. B. von Linux-Distributoren.</p> <p>Im privaten Teil können Benutzer ihre Docker-Images hochladen und dadurch einfach z. B. firmenintern verteilen, ohne dass diese damit öffentlich auffindbar sind.</p>
Repository:	Satz gleichnamiger Images mit verschiedenen Tags, zumeist Versionen.
Tag:	<p>Dienen zum Identifizieren verschiedener Versionen eines Images.</p> <p>Diese Definition gilt nur im Zusammenhang mit Docker. Sie wird nicht auf Begriffe im Zusammenhang mit Variablen oder Parameter von Batch Control verwendet.</p>

5 Docker for Windows

Docker für Windows ermöglicht Ihnen die Verwendung von zenon Runtimes in einem Docker Image.

5.1 Installation und Konfiguration

Zum Betrieb von zenon in einem Docker Container benötigen Sie neben zenon auch Hyper-V und Docker.

Für die Installation von Hyper-V und Docker kann COPA-DATA keinen Support zur Verfügung stellen.

Informationen zu Docker für Windows finden Sie unter anderem auf:

- ▶ Allgemein: <https://docs.docker.com/docker-for-windows/>
(<https://docs.docker.com/docker-for-windows/>)
- ▶ Netzwerk: <https://docs.docker.com/network/> (<http://www.>)
- ▶ Netzwerk-Features: <https://docs.docker.com/docker-for-windows/networking/>
(<https://docs.docker.com/docker-for-windows/networking/>)

VORAUSSETZUNGEN

Um Docker mit zenon nutzen zu können, benötigen Sie:

- ▶ passendes Betriebssystem
 - ▶ Windows 10 Pro oder höher
 - ▶ Windows Server 2016 oder höher
 - ▶ aktivierte Virtualisierungsumgebung
 - ▶ eine gültige Lizenz
- Hinweis:** zenon kann in einem Docker Container nicht im Demo-Modus betrieben werden.

Für die Konfiguration benötigen Sie:

- ▶ Docker Desktop
- ▶ vorhandene zenon Installation
- ▶ Powershell

DOCKER INSTALLIEREN UND KONFIGURIEREN

Um Docker für Windows zu installieren:

1. Laden Sie Docker for Windows von der Website
<https://docs.docker.com/docker-for-windows/install/> (<http://www.>)
Sie benötigen Version 2.0.3 oder neuer.
2. Installieren Sie das Paket.
Beachten Sie dabei:
 - ▶ Auf der Konfigurationsseite muss die Option **Use Windows containers instead of Linux containers Windows-Container** aktiviert sein.
 - ▶ Wenn das Setup abgeschlossen ist, werden Sie abgemeldet.
 - ▶ Überprüfen Sie nach dem erneuten Anmelden, ob Hyper-V aktiviert ist.
Ist das nicht der Fall, aktivieren Sie Hyper-V.
3. Klicken Sie mit der rechten Maustaste auf das Docker-Symbol im Infobereich der Taskleiste.

4. Wählen Sie im Kontextmenü **Switch to Windows container...**
Wird dieser Eintrag nicht angezeigt, ist der Docker Desktop bereits richtig konfiguriert.
5. Überprüfen Sie, ob Docker korrekt installiert ist.
Dazu:
 - ▶ Öffnen Sie Windows eine Powershell.
 - ▶ Geben Sie ein: **docker --version**
Die Versionsinformationen werden angezeigt.

EXTERNER HYPER-V SWITCH

Per Default wird für Docker-Container ein NAT-Switch verwendet. Allerdings werden Sie Ihre zenon Runtime von anderen Clients aus über den Docker Container Hostnamen ansprechen.

Um Ihre Docker Container für DHCP- und DNS-Server freizugeben:

1. Erstellen Sie in Hyper-V im **Virtual Switch Manager** einen externen Switch.
2. Starten Sie Docker Desktop neu.
Damit wird der virtuelle Switch erkannt.
3. Überprüfen Sie, ob der neue Switch verfügbar ist.
Dazu geben Sie in Powershell ein: **docker network ls**

Weitere Informationen zu zenon Netzwerke mit Docker finden Sie im Abschnitt **Runtime-Docker mit Namensauflösung** (auf Seite 12).

5.2 Docker Image für zenon Runtime beziehen aus Docker Hub

Das Image für die zenon Runtime finden Sie auf <https://hub.docker.com/u/copadata> (<https://hub.docker.com/u/copadata>).

Um das Image von Docker Hub zu laden:

1. Öffnen Sie Powershell.
2. Wählen Sie die gewünschte Runtime-Version aus.
3. Laden Sie das Image mit dem angezeigten **Docker Pull Command** herunter.
4. Sie erhalten Informationen zum Fingerabdruck und zum Download-Status.

5.3 Container mit zenon Runtime erstellen

Legen Sie jetzt einen Container mit einer zenon Runtime an. Der Container enthält einem Verweis auf die zenon Dateien, LOG-Dateien, Runtimedateien und Lizenzen werden dort gespeichert.

Um den Container zu erstellen:

1. Erstellen Sie auf Ihrem Host einen Ordner, in dem die Runtime-Dateien Ihre Runtime-Containers gespeichert wird. Hier werden Informationen für Einstellungen und Projekte abgelegt. Dieser Ordner wird in der Dokumentation als **data path** bezeichnet. Die Struktur finden Sie auf dem Installationsmedium als Datei **zenonDataTemplate.zip**.
2. Kopieren Sie die Runtime-Dateien in den Unterordner **data-path\projects**.
Bleibt dieser Ordner leer, werden die Dateien beim ersten Starten der Runtime im Container erzeugt.
3. Konfigurieren Sie die zenon Installation. Dazu bearbeiten Sie Dateien im **data path**.
Details dazu finden Sie im Kapitel **zenon Runtime für Docker konfigurieren** (auf Seite 9).
4. Ändern Sie die Werte in der **zenon6.ini** (**data path\system\zenon6.ini**):
 - ▶ **VPF30**: Pfad zum Runtime-Projekt.
Z. B.: **C:\zenondata\projects[workspace]**
 - ▶ **DEFANWENDUNG30**: Name des Runtime-Projekts.
5. Ändern Sie die **settings.xml** (**data-path\settings.xml**).
Hier konfigurieren Sie den Lizenzserver, den primären DNS-Namen und die zu startenden Prozesse.
6. Öffnen Sie Powershell.
7. Geben Sie das Kommando ein: **docker run -d -v [data-path]:C:\zenondata --name [Container-Name] copadata/zenon-runtime820-windows**
 - ▶ **data-path**: Pfad zum Ordner am Host, der die gespeicherten Daten enthalten soll. Der Pfad, den Sie im ersten Schritt erstellt haben.
 - ▶ **Container-Name**: Name des neu erstellten Containers, zum Beispiel: zenon

Beispiel: **docker run -d -v C:\zenondata:C:\zenondata --name zenon copadata/zenon-runtime820-windows**

Dadurch werden Dateien im Ordner **[data-path]** erstellt.

 - ▶ Stoppen Sie den Container.
Dazu geben Sie ein: **docker stop [Container-Name]**
Container-Name: Name des Containers, der zuvor mit **docker run** erstellt wurde.

5.4 zenon Runtime für Docker konfigurieren

VORAUSSETZUNGEN

Um zenon in einem Docker Container nutzen zu können, müssen auch die dafür nötigen Dienste vorhanden sein. Diese werden bei der Installation automatisch aktiviert.
Damit ist es möglich:

- ▶ Runtime-Dateien per Remote-Transport zu übertragen.
- ▶ Runtime-Dateien per Remote-Transport zurückzulesen.
- ▶ Die Runtime automatisch zu starten, neu zu laden und zu beenden.

ORDNER UND DATEIEN

Für den Start müssen im **Datenpfad** alle benötigten Unterordner und Dateien vorhanden sein:

- ▶ **System\K5LICENSE.INI**: Lizenz für zenon Logic.
- ▶ **System\License.ini**: Lizenz für zenon.
- ▶ **System\zenon6.ini**: Einstellungen für zenon.
- ▶ **settings.xml**: Einstellungen für Lizenzserver, Startverhalten des Containers und Primary DNS-Suffix.
- ▶ **Projects**: Sammelt Projektdateien.
- ▶ **LOG**: Sammelt LOG-Dateien.

Beim Start werden die konfigurierten Lizenzserver überprüft.

Ist eine gültige Netzwerklizenz vorhanden, ist die Runtime lizenziert. Ist keine gültige Lizenz vorhanden, kann die Runtime nicht gestartet werden.

LOGGING

Wird in der Runtime eine Funktionalität verwendet, die mit Docker nicht unterstützt wird, dann erfolgt ein Eintrag in die LOG-Datei.

Speicherorte:

- ▶ zenon LOG-Dateien: ...*[data path]*\LOG
Esehbar mit Diagnosis Viewer.
- ▶ Docker LOG-Dateien: *Windows Logs\Application ...*
Esehbar mit Windows Ereignisanzeige.

KONFIGURATION

Um zenon für die Verwendung in einem Docker Image zu konfigurieren, müssen die Dateien **settingx.xml** sowie **license.ini** und **zenon6.ini** mit einem Texteditor angepasst werden.

SETTINGS.XML

In der Datei settings.xml konfigurieren Sie:

- ▶ CodeMeter Lizenzserver: Dieser Eintrag wird von der **license.ini** benötigt.
- ▶ Primäres DNS-Suffix: Für Netzwerk-Weiterleitungen.

Um settings.xml zu konfigurieren:

1. Navigieren Sie zum Pfad ...\ [data path]\.
2. Öffnen Sie die Datei **settings.xml**.
3. Navigieren Sie zum Abschnitt **<CodeMeterLicenseServer>**.
 - ▶ Tragen Sie zwischen den Tags **<Address>** und **</Address>** den Namen des Rechners ein, auf dem die Lizenz hinterlegt ist.
4. Wiederholen Sie diesen Schritt bei Bedarf für einen weiteren Server-Eintrag.
5. Wechseln Sie zum Abschnitt **<Network>**.
 - ▶ Tragen Sie zwischen den Tags **<PrimaryDnsSuffix>** und **</PrimaryDnsSuffix>** das primäre DNS-Suffix für Ihr Netzwerk ein.
Das ist der Teil Ihres DNS-Eintrags ohne den Host-Namen.
Bleibt dieser Eintrag leer, wird die Defaulteinstellung des Containers benutzt.
6. Konfigurieren Sie bei Bedarf den Autostart der Runtime im Container.
Dazu tragen Sie zwischen den Tags **<Autostart>** und **</Autostart>** die Angaben für Dienst und Programm ein:
 - ▶ **<Program>**: Leitet den Abschnitt für den Start des Dienstes ein.
 - ▶ **<Active> </Active>**: Aktiviert den Autostart für den Dienst.
Aktiv: *true*
Inaktiv: *false*
Zum Beispiel: **<Active>true</Active>**.
 - ▶ **<Path> </Path>**: Pfad zum Dienst.
Zum Beispiel **<Path>C:\zenonSetup\64\zenSysSrv.exe</Path>**.
 - ▶ **</Program>**: Schließt den Abschnitt.
 - ▶ **<Program>**: Leitet den Abschnitt für den Start des Programms ein.
 - ▶ **<Active> </Active>**: Aktiviert den Autostart für das Programm analog zum Dienst.
Zum Beispiel: **<Active>true</Active>**.
 - ▶ **<Path> </Path>**: Pfad zum Programm.
Zum Beispiel **<Path>C:\zenonSetup\64\zenrt32.exe</Path>**.
 - ▶ **</Program>**: Schließt den Abschnitt.
7. Speichern Sie die Datei.

LICENSE.INI: LIZENZ EINTRAGEN

Um zenon in einem Docker Container nutzen zu können, müssen Sie eine gültige Lizenz zur Verfügung stellen und den Ort, wo die Lizenz gespeichert ist, definieren.

Informationsquellen für:

- ▶ Lizenz: Lizenzschein oder License Manager
- ▶ Lizenzablageort: Codemeter Web Admin

Um zenon für Docker zu lizenzieren:

1. Navigieren Sie zum Pfad ...*[data path]*\system.
2. Öffnen Sie die Datei **License.ini**.
3. Tragen Sie die Lizenzdaten entsprechend Ihres Lizenzscheins ein:
 - ▶ **SERIAL0** = : Seriennummer Ihrer Lizenz.
Dieser Eintrag ist obligatorisch.
 - ▶ **SERIAL0_DESC** =: Optionale Beschreibung für die Verwendung der Lizenz.
 - ▶ **SERIAL0_LOCATION** =: Adresse des Lizenzservers.
Dieser Eintrag ist obligatorisch.
4. Speichern Sie die Datei.
5. Stoppen Sie den Docker-Container und starten Sie ihn neu.
Geben Sie dazu ein: **docker stop [container name]** und **docker start [container name]**

Damit steht Ihnen ein lizenziertes zenon im Docker Container zur Verfügung.

ZENON6.INI

Um das Runtime-Projekt zu spezifizieren:

1. Navigieren Sie zum Pfad ...*[data path]*\system.
2. Öffnen Sie die **zenon6.ini**.
3. Bearbeiten Sie die folgenden Einträge:

[PATH]

VPF30= Tragen Sie hier den Pfad zum Arbeitsbereich ein.

Beispiel: **C:\zenondata\projects\Workspace1**

[DEFAULT]

DEFANWENDUNG30= Tragen Sie hier den Projektnamen ein.

5.4.1 Runtime-Docker mit Namensauflösung

Damit der zenon Runtime Server läuft und erreichbar ist, wird ein Docker-Container benötigt, auf den von außen über den Container-Hostnamen zugegriffen werden kann. Soll der Zugriff domainübergreifend erfolgen, muss der vollständig qualifizierte Hostnamen (FQDN) verwendet werden.

Hinweis: Hostnamen dürfen laut Windowskonvention maximal 15 Zeichen lang sein.

VORAUSSETZUNGEN

Sie benötigen:

- ▶ Ein bereits geladenes zenon Runtime Docker Image.
- ▶ Eine bereits für Docker konfigurierte (auf Seite 9) zenon Runtime. Vor allem **settings.xml** und **license.ini** müssen konfiguriert sein.

KONFIGURIEREN EINES VIRTUELLEN EXTERNEN SWITCHES IN HYPER-V

Sie benötigen in Ihrem Hyper-V Server einen externen Netzwerk-Switch. Überprüfen Sie dies im Hyper-V Server in der Einstellung **Virtual Switch Manager**. Ist kein externer Switch verfügbar ist, erstellen Sie einen. Dieser externe Switch wird vom Docker-Container verwendet.

PRIMARY DNS SUFFIX IM DOCKER-CONTAINER SETZEN

Wenn Sie einen Servernamen mit DNS-Suffix verwenden möchten, müssen Sie dem Docker-Container das **Primary DNS Suffix** mitteilen. Zum Beispiel: **zenRT001.domain.internal**. Ist das **Primary DNS Suffix** nicht gesetzt, erkennt die zenon Runtime nicht, dass sie als Server fungiert. Das gilt auch, wenn der Docker-Container von außen mit Ping erreicht werden kann. Das Suffix konfigurieren Sie in der **Settings.xml** (auf Seite 9) im Tag **<Settings> <Network> <PrimaryDnsSuffix>**.

CONTAINER MIT EXTERNEM SWITCH ERSTELLEN

Einen Container mit externen Switch erstellen Sie mit dem Befehl:

```
docker run -d -v [data path]c:\zenondata --network [name of external switch] --hostname [unique host name] --name [container name] copadata/zenon-runtime820-windows
```

BEISPIEL

Es soll ein Container erstellt werden, der folgende Bedingungen erfüllt:

- ▶ verwendet den virtuellen Hyper-V-Switch **zrtExtSwitch**
- ▶ setzt den Hostname auf **zenRT001**
- ▶ kann den Container mit dem Namen **zenon** verwalten

Befehl: **PS> docker run -d -v C:\zenondata:C:\zenondata --network zrtExtSwitch --hostname zenRT001 --name zenon copadata/zenon-runtime820-windows**

ZENON SERVER EINRICHTEN

Um den zenon Server einzurichten:

- ▶ Verwenden Sie den zuvor konfigurierten Hostnamen **zenRT001.domain.internal** als Servernamen in Ihrem zenon Netzwerkprojekt.
- ▶ Den gleichen Hostnamen können Sie für den zenon Remote Transport verwenden.
- ▶ Nach der Übertragung der Runtime-Dateien und dem Start der zenon Runtime auf dem Docker-Container können zenon Clients den Server unter **zenRT001.domain.internal** erreichen.

FEHLERBEHEBUNG

Beim Auftreten von Fehlern:

- ▶ Stellen Sie sicher, dass die zenon Runtime sich selbst als Server erkennt. Überprüfen Sie dazu die LOG-Dateien.
- ▶ Überprüfen Sie die Netzwerkverbindung zwischen Client und Server.

REMOTE-TRANSPORT: NEUSTART DES CONTAINERS VERHINDERT VERBINDUNG

Beim Neustart eines Docker Containers erhält dieser eine neue IP-Adresse. Dabei wird aber der DNS-Cache für den Hostnamen des Containers nicht aktualisiert. Verbindungen mit dem Container per Hostname sind damit nicht mehr möglich.

Lösungen:

1. Im Kontext der gleichen Domäne: Führen Sie die Befehlszeile **ipconfig /flushdns** aus.
2. Generell: Warten Sie den Timeout für die Aktualisierung des DNS Cache ab.

5.4.2 zenon-Standard-Ports

Von zenon per Default benutzte Ports:

Anwendung	Standard-Port
Network Service	1100
Transport Service	1101
WEB Service Classic	1102
DB Service	1103
SQL Browser Service, (für Verteiltes Engineering im Editor)	1434

Anwendung	Standard-Port
zenAdminSrv.exe	50777
zenLicTransfer (License Transfer Service)	50784
Logging Service	50780
SNMP Trap Service	50782
WEB Service Tunneling	8080

5.5 Befehle für Docker (Auswahl)

Für die Konfiguration von zenon mit Docker benötigen Sie eine Reihe von Befehlen. Diese geben Sie über Windows Powershell ein. Die Tabelle enthält häufig benötigte Befehle.

Informationen dazu finden Sie auch auf:

<https://docs.docker.com/engine/reference/commandline/docker/> (http://www.)

Befehl	Beschreibung
<code>docker --version</code>	Zeigt Docker Versionsinformationen an.
<code>docker network ls</code>	Listet alle Netzwerke auf, die der engine daemon kennt. Das schließt Netzwerke ein, die sich über mehrere Hosts in einem Cluster erstrecken.
<code>docker images</code>	Listet alle geladenen Images auf.
<code>docker run -d -v [data path]:c:\zenondata --name [container name] [image name]</code>	Erstellen eines Containers aus einem geladenen Image.
<code>docker run -d -v [data path]:c:\zenondata --network [name of switch] --hostname [host name] --name [container name] [image name]</code>	Erstellen eines Containers aus einem geladenen Image inklusive Weiterleitung von Ports für das zenon Netzwerk.
<code>docker ps -a</code>	Listet alle Container auf, einschließlich der gestoppten.
<code>docker start [container name]</code>	Startet einen gestoppten Container.
<code>docker stop [container name]</code>	Stoppt einen gestarteten Container. Tipp: Um alle Container zu stoppen, benutzen Sie den Befehl <code>\$(docker ps -a -q) as</code>

Befehl	Beschreibung
	[container name].
<code>docker inspect [container name]</code>	Zeigt Containerdetails einschließlich Hostname und Netzwerkkonfiguration an.
<code>docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' [container name]</code>	Druckt die statische IP-Adresse des Containers.
<code>docker rm [container name]</code>	Entfernt einen Container. Tipp: Um alle Container zu entfernen, benutzen Sie den Befehl <code>\$(docker ps -a -q) as [container name]</code> .
<code>docker exec -it [container name] powershell</code>	Ermöglicht es, Powershell-Befehle im Docker-Container auszuführen.
<code>docker system prune -a</code>	Dieser Befehl entfernt: <ul style="list-style-type: none"> ▶ alle gestoppten Container ▶ alle Netzwerke, die nicht von mindestens einem Container benutzt werden ▶ alle Images, die nicht mit mindestens einem Container verknüpft sind ▶ alle Build Caches