

Container-Images erstellen in Docker (Tutorial für Anfänger)

Im ersten Tutorial zu Docker haben wir besprochen, wie du Container basierend auf bereits existierenden Images aus der Docker-Registry starten kannst. Dieses kurze Tutorial soll dir zeigen, wie du ein Image, passend zu deinen individuellen Anforderungen, erstellen kannst.

In diesem „Beispiel“-Fall wird der Container eine statische HTML-Website mit Nginx (einem Web-Server) ausführen. Der Name des Computers (oder Servers), auf dem Docker läuft, wird ganz einfach nur „Docker“ genannt. Wenn du auf einen der Dienste zugreifen möchtest, dann benutze docker anstelle von localhost oder 0.0.0.0.

(Du benötigst kein Vorwissen zu Docker. Solltest du jedoch eine kurze Anleitung zu den Basics durchlesen wollen, findest du hier den Artikel „Docker lernen für Anfänger“)

Wie funktionieren Docker-Images?

Docker-Images werden mithilfe eines Dockerfile's erstellt. Ein Dockerfile definiert alle Schritte, die nötig sind, um ein Docker-Image zu erstellen. Deine Anwendung wird dabei so konfiguriert, dass sie bereit ist als Container gestartet zu werden.

Das eigentliche Image enthält alle Dateien, vom Betriebssystem bis hin zur Konfiguration und allen Dependencies, die du benötigst, damit deine Anwendung fehlerfrei läuft.

Alle Dateien in dem Image parat zu haben hat den Vorteil, dass du deine Umgebungen überallhin migrieren kannst. Es stellt sicher, dass deine Anwendung überall funktioniert, wo du sie ausführst.

Das Dockerfile ermöglicht es, dass Images zusammengesetzt werden können, so dass Benutzer bestehende Images erweitern können. Das hat den Vorteil, dass du deine Images nicht jedes Mal von Grund auf neu bauen musst, sobald sich kleine Änderungen ergeben. Wenn du auf einem bestehenden Image aufbaust, brauchst du nur die Schritte zur Einrichtung deiner Anwendung zu definieren. Die Basis-Images können bspw. Betriebssystem-Installationen oder bereits konfigurierte Systeme sein, die lediglich einige zusätzliche Anpassungen benötigen.

1. Ein Base-Image erstellen

Alle Docker-Images beginnen mit einem Base-Image.

Ein Base-Image ist identisch mit den Images aus der Docker-Registry, die zum Starten von Containern verwendet werden. Zusammen mit dem Image-Namen können wir auch ein Image-Tag angeben. Dadurch kannst du z.B. angeben, welche Version eines Programmes installiert wird. Wird keine Angabe zur Version gemacht, wird standardmäßig die aktuellste Version benutzt.

Diese Base-Images werden als Grundlage für deine zusätzlichen Änderungen verwendet. In diesem Fall benötigen wir zum Beispiel eine konfigurierte und lauffähige Nginx-Version, bevor wir die

statischen HTML-Dateien bereitstellen können. Wir benutzen daher Nginx als Base-Image.

Dockerfile's sind einfache Textdateien, die pro Zeile einen Befehl enthalten. Damit du ein Base-Image definieren kannst, benutzt du folgenden Befehl:

```
FROM <image-name>:<tag>
```

Da wir Nginx benötigen, sollte unsere erste Zeile im Dockerfile wie folgt aussehen:

```
FROM nginx:1.11-alpine
```

Schreibe diese Zeile in dein Dockerfile und speichere die Datei.

Hinweis zur Versionierung von Images:

Es ist so schön einfach :latest-Tag zu verwenden, um die aktuelle Image-Version zu erhalten. Jedoch ist hier vorsicht geboten. Nicht immer bekommst du die Version des Image, die auf deine Anwendung abgestimmt ist. Ich empfehle dir daher immer die spezifische Versionsnummer als Tag anzugeben.

Sobald du eine Aktualisierung vorgenommen hast, solltest du dein Dockerfile auf die entsprechend Versionsnummer abändern.

Ja, das ist mehr Aufwand, aber es schützt auch vor Kopfschmerzen durch inkompatible Versionen!

From:
<https://blog.cooltux.net/> - TuxNet DokuWiki

Permanent link:
https://blog.cooltux.net/doku.php?id=it-wiki:docker:docker_createcontainerimages&rev=1634486571

Last update: 2021/10/17 16:02

