

# Isolierte Anwendungsumgebungen mit Docker (Arbeitstitel)

Lars Tum, Henryk Hodam

+++**STORYBOARD**+++      **V | 05**

Ziel: Die Zuschauer sollen verstehen:

- Was Docker ist und wofür es eingesetzt wird
- Welchen Vorteil im geographischen/geoinformatischen Kontext bringen kann
- Wie Docker grundsätzlich funktioniert (Dockerfile/ Image / Container )

NR   SZ	SKIZZE DER VISUALISIERUNG	BESCHREIBUNG DER VISUALISIERUNG UND ANMERKUNGEN	M U S I K	SPRECHTEXT
1		Erscheinen des Dockerwals. Verschiedene Anwendungen ( Logos) werden in einen Container geladen und dann auf verschiedene Systeme „verschifft“ ( Am besten gleich den Wal statt eines Schiffes!!) i. Am Ende Betriebssysteme ( Windows, MacOS, Linux) einblenden.. oder über die dargestellten Systeme schreiben ( Dann drei nicht zwei wie im Bild!!)		<p>Anwendung sind häufig auf eine spezielle Umgebung angewiesen. Ein bestimmtes Betriebssystem, das Fehlen oder Vorhandensein weiterer Anwendung oder Bibliotheken kann entscheidend dafür sein, ob sie Ihren Anforderungen gerecht werden können.</p> <p>Mit Docker können Anwendungen in eigenen Umgebungen in einem Paket, einem so genannten Container zusammengefasst, und auf verschiedenen Systemen lauffähig bereitgestellt werden. Diese Container können einfach geteilt werden, und es ist kein Problem mehrere solcher Container gleichzeitig auf einem System zu betreiben. Und dabei steht Docker inzwischen auf allen wichtigen Betriebssystemen zur Verfügung, womit auch die Container unabhängig vom Betriebssystem lauffähig sind.</p>

2

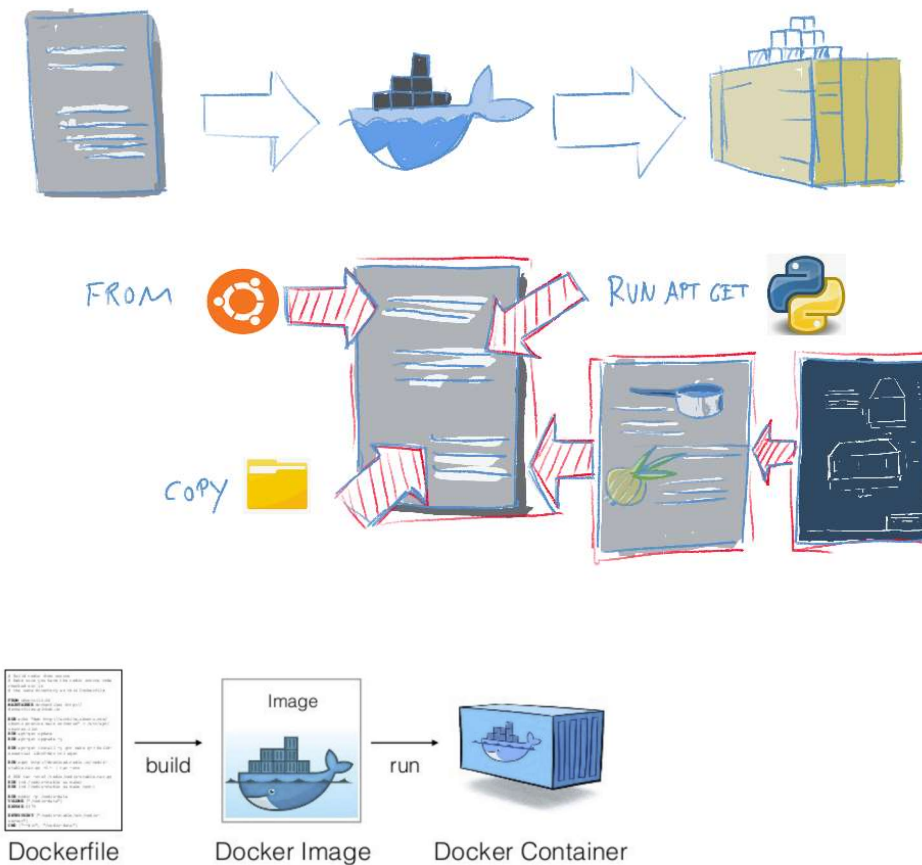


Ein Server mit WEBGIS Setup liefert Daten an einen Endnutzer. Endnutzer vervielfältigen sich -> Server mit gleicher Infrastruktur (Horizontal) entstehen. Oder auch gleiche Container auf einem System (Vertikal) Oder vertikal und die lasten werden gleichmäßiger verteilt sobald neue Container dazukommen. Lasten mit Balken visualisieren.

Nutzer sitzen vor Rechner. Im Rechner/Backend sind einige GIS Anwendungen zu sehen von denen ein paar ausgetauscht werden ( z.B. Server: Apache durch NGINX ersetzt) um „experimentieren zu verdeutlichen ) Es könnten auch verschiedenen Setups typischer Lernumgebungen durchlaufen.

Auch im Bereich Geographischer Informationssysteme ist Docker nicht mehr wegzudenken, denn es vereinfacht die reibungslose Entwicklung, Bereitstellung und Skalierung webbasierter GIS-Infrastrukturen. Und auch das Experimentieren und Lernen mit neuen innovativen Werkzeugen zur Geodatenverarbeitung fällt in isolierten Umgebungen leichter, in denen verschiedene GIS- Anwendung und Versionen von Tools aufeinander abgestimmt und unabhängig vom Betriebssystem bereitgestellt werden können.

3



Dockerfile, Docker Image und Container werden als Illustrationen dargestellt (Auf originale Logos achten!!)

Zoom auf abstrahiertes Dockerfile. Wird kurz als Rezept und als Bauplan (Blaupause) dargestellt.

Einzelne im File definierte Elemente ( Arbeitsumgebung: Ubuntu (besser Alpine??), Python, Daten-> siehe Text) werden dargestellt und angedeutet mit dem Befehl (FROM,RUN APT, COPY) als Icons in das Dockerfile eingetragen.

Für ein einfaches Verständnis dieser Technologie verfolgen wir einmal die Entstehung eines Docker Containers über drei Schritte: Vom Dockerfile, zum Image und schließlich zum Container.

1. Dockerfiles

Ein Dockerfile ist eine Textdatei, welche Instruktionen dazu enthält, wie das Docker Image „gebaut“ werden soll. Stell dir ein Dockerfile wie ein Rezept beim Kochen oder einen Bauplan vor.

Welche Benutzerumgebung soll unserem Dockercontainer zu Grunde liegen?. Soll er im Netz als Server auffindbar sein? Ist eine Laufzeitumgebung für eine Programmiersprache notwendig? Sollen dem Container vielleicht wichtige Daten von Anfang an zur Verfügung stehen? Hier im Dockerfile wird all das festgelegt!



## 2. Image

Darstellung z.B. einer Blaupause oder eines Kochrezepts (mit der Aufschrift „Dockerfile“) und Animation eines sich aufbauenden Docker Images. (Build- Befehl in einer abstrahierten Konsole visualisieren)

Visualisierung, dass die Komponenten lokal und aus dem Netz zusammengetragen werden. Ein Schloss am Image um die Unveränderbarkeit zu zeigen.

Illustration eines Klemmbaustein Baukastens (Logo verändert, Name nicht lesbar!) Darin Die Bausteine und eine Anleitung) Das fertige Modell daneben wird durchgestrichen.

## 2. Images

Sind alle notwendigen Bestandteile unseres Containers im Dockerfile beschrieben, erfolgt die Anweisung an Docker diese zu einem Image zusammenzufassen. Jede der im Dockerfile beschriebenen Komponenten wird zusammengetragen und im Image abgelegt. Es enthält damit alle notwendigen Bestandteile und Informationen um einen lauffähigen Container zu erstellen. Bereits erstellte Images sind unveränderlich, können unter Anwendern geteilt und von öffentlichen Repositories wie Docker Hub heruntergeladen werden.

Ein Docker Image kann man sich auch wie einen Bausatz vorstellen, den man beliebig oft bauen kann weil seine Bausteine unendlich sind. In Ihm ist alles vorhanden, um ein Modell zusammenzusetzen. Aber: Der Bausatz ist noch nicht das Modell selbst.





3. Container

Darstellung der Bausteine die aus der Packung gekippt werden. Setzen sich zu einem Model zusammen. Das Modell wird vervielfältigt.

„Docker run“ wird in eine abstrahierte Konsole eingetippt, Das Image Paket wird ausgekippt , und das „System“ fällt aus dem Paket. Und baut sich auf dem System auf ( Arbeitsumgebung, Dateisystem laufende Prozesse....) -> vielleicht fällt als „schwarzweiß“ aus dem Paket in das System wo sich nach und nach Arbeitsumgebung, Dateisystem, Prozesse einfärben, also bereitstehen)

Das Ganze auf verschiedenen Rechnern/Systemen mit unterschiedlichen Betriebssystemen

Die Container vervielfältigen sich in den Rechnern

3. Container

Erst wenn dieser Baukasten der Anleitung folgend, unter Verwendung der beigelegten Bausteine zusammengesetzt wurde ist eine Instanz des Modells, vorhanden. Und da die Bausteine in diesem speziellen Baukasten unendlich sind kann auch eine weitere Instanz gebaut werden ...und noch eine ... und noch eine.

Und so wird mit dem richtigen Befehl auf einem System auf dem Docker installiert ist auch aus dem Image ein Container - eine Instanz des Image als Isolierte Arbeitsumgebung mit eigenem Dateisystem, laufenden Prozessen und Anwendungen. Ganz gleich auch welchem Betriebssystem. Und, bei Bedarf, auch noch eine zweite.. oder dritte.

Animierte Ausschnitte der vorherigen Sequenzen.

Docker ermöglicht es also mithilfe isolierter Container ganze Anwendungsumgebungen auf verschiedenen Systemen zu bereitzustellen.

Die Entwicklung und Implementierung von anspruchsvollen geografischen Informations- und Analysesystemen werden so erleichtert. Zudem eröffnen isolierte Docker-Container vielfältige Möglichkeiten für das Lernen und das Experimentieren mit Geodaten.

Im Dockerfile sind alle Anweisungen für Docker enthalten, um ein Image anzufertigen. Dieses ist unveränderlich und kann leicht verbreitet werden. Ein Docker Container ist eine Instanz dieses Images und kann auf jedem System auf dem Docker lauffähig ist gestartet werden.