FAQ Docker-Einstieg

Antworten auf die häufigsten Fragen

Von Peter Siering

Einordnung

? Was ist Docker, wofür soll es überhaupt gut sein?

! Im IT-Alltag steht Docker dafür, einzelne Programme in Containern stärker voneinander zu trennen, als Betriebssysteme das für Prozesse bewerkstelligen. Primär hilft das, Programme mit unterschiedlichen Ansprüchen in individuell auf sie abgestimmten Umgebungen zu betreiben, sodass es nicht zu Versionskonflikten bei Bibliotheken kommt.

Das ist für Web-Entwickler praktisch, die Projekte aus vielen Einzelteilen, etwa Webserver, Datenbank und Skriptsprache zusammensetzen. Die Teile liegen in eigenen Containern, sodass Tests schnell in reproduzierbaren Umgebungen möglich sind (DevOps). Auch Admins und Server-Betreiber können profitieren: Server-Software lässt sich heute als Container bequem erproben und auch produktiv nutzen, etwa auch auf einem x86-NAS (siehe Seite 142).

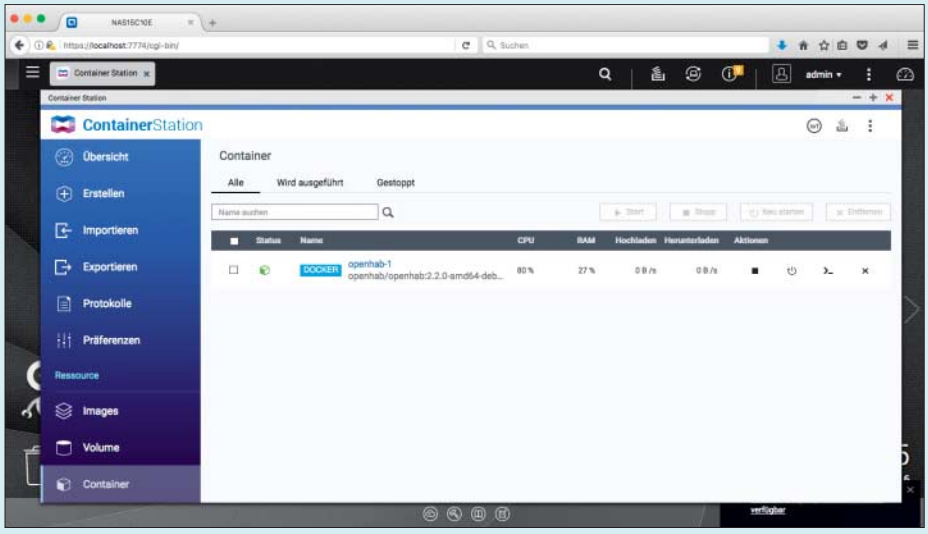
Hinter Docker steht eine gleichnamige Firma, die die Werkzeuge entwickelt, um Images zu bauen, zu verteilen und mit diesen Images Container zu starten. Die Technik, die bei der isolierten Ausführung hilft, steckt in gängigen Linux-Kerneln und in Windows. Die Software unter Apache Lizenz ist kostenlos nutzbar. Docker will mit Support Geld verdienen.

Grafische Oberfläche für Docker

? Geht Docker nur auf der Kommandozeile?

! Es gibt schon einige grafische Werkzeuge, um Docker-Container zu verwalten. Allerdings entwickelt sich der Markt relativ schnell, sodass mancher viel versprechende Ansatz bereits wieder eingestellt worden ist. Einen bislang haltbaren, pragmatischen Ansatz stellt Portainer dar, das selbst ein Docker-Container ist und eine Weboberfläche bereitstellt.

Portainer kann übrigens mehr, als nur einen simplen Host mit nur wenigen Containern verwalten: Es kann sich auch um einen Docker-Schwarm kümmern. Das sind auf mehreren Servern laufende Docker-Daemons, die ein gemeinsames logisches Netzwerk nutzen und die eine Skalierung von Diensten innerhalb des Schwarms erlauben. Wer im großen Stil mit Containern hantiert, setzt zur „Orchestrierung“ auf Kubernetes und Co.



Moderne x86-NAS-Systeme bieten Web-Oberflächen zur Docker-Nutzung – für Container-basierte Web-Oberflächen auf regulären Linux-Distributionen muss man oft zumindest einmal auf die Kommandozeile.

Container vs. Virtualisierung

? Was ist der Unterschied zwischen Virtualisierung und Containern?

! Letztlich sind Container eine leichtgewichtige Form der Virtualisierung: Sie trennen Arbeitsumgebungen voneinander, begrenzen auf Wunsch die bereitgestellten Ressourcen und erlauben es, ohne großen Software-Kuddelmuddel auch fette Hardware auszunutzen. Sie teilen sich einen Kernel und enthalten ansonsten nur die nötigsten Bibliotheken. Trotzdem hat jeder Container beispielsweise eine eigene Netzwerkadresse, meist aus einem Docker-eigenen Netzwerk, auf Wunsch aber auch aus dem lokalen Netz.

Docker frisst Plattenplatz

? Nach umfangreichen Experimenten mit verschiedenen Docker-Containern sind zig GByte freier Plattenplatz verschwunden. Wie findet man heraus, wo der abgeblieben ist?

! Viel Platz nehmen die Images und Volumes ein – also die Vorlagen für Container und die beim Starten angelegten Volumes, in denen Docker die Daten der Container aufbewahrt. Auf einem Linux-System landen diese Daten der Container in der Regel in /var/lib/docker. Dort manuell durchzufegen ist aber eine doofe Idee. Der Docker-Daemon weiß am besten, was weg kann und was noch gebraucht wird, etwa weil ein anderer Container dasselbe Image nutzt.

Mit `docker volume prune` und `docker image prune` lässt sich ein Aufräumprozess starten. Nicht mehr in Nutzung befindliche Volumes und Images löscht Docker. Vorsicht: Es ist üblich, die Nutzdaten eines Containers auf separaten Volumes zu lagern, und der prune-Befehl löscht die, wenn kein Container mehr darauf ver-

weist (er muss nicht laufen, aber eben existieren). Deswegen gehört /var/lib/docker ins Backup.

Dateien austauschen

? Wie kann man am einfachsten eine überarbeitete Datei in einen laufenden Container hineinbringen?

! Wenn ein Container läuft, sollte das nur mit Dockers Hilfe geschehen: Der Befehl `docker cp` akzeptiert als Quell- oder Zielpfad einen dem Pfad vorangestellten Container-Namen. Mit `docker cp web:/var/www/index.html /tmp` würden Sie die HTML-Datei aus dem Container `web` ins temporäre Verzeichnis des Hosts kopieren.

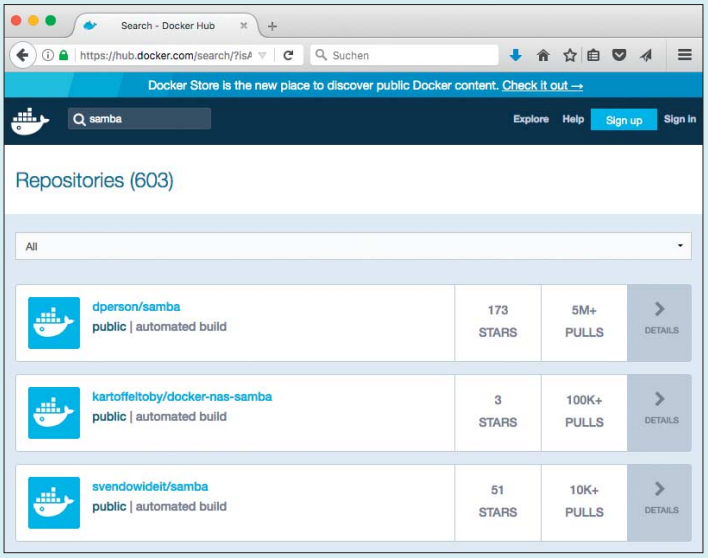
Fertige Container-Images

? Wo findet man fertige Container-Images, wenn man nicht gleich selbst Container bauen will?

! Die Quelle schlechthin ist der Docker-Hub, den die Firma Docker betreibt. Dort finden sich für nahezu jede gängige Software Images. Aber: Das Angebot ist riesig. Jeder kann dort Images hochladen – Docker spricht selbst von „wildem Westen“. Ergänzt wird der Hub durch einen Store, in dem es kuratierte Images gibt, dazu zählen auch nach Community-Standards gepflegte offizielle Images. Die Inhalte von Hub und Store stellt Docker als sogenannte Registry bereit.

Die Docker-Software ruft die Images aus der Registry ab (Pull genannt) und kann sie durchsuchen. Zur Orientierung eignen sich die Sterne, mit denen andere Benutzer ein Image bewertet haben, die Anzahl der Downloads und der Hinweis, dass es sich um ein offizielles Image handelt. Die Suche per Webbrowser auf dem Docker-Hub liefert alle Treffer, die auf der Kommandozeile (`docker search <name>`) standardmäßig nur 20, maximal 100.

Letztlich geht hier Probieren über Studieren. Viele Images auf dem Docker-Hub sind schlecht gepflegt und ohne weitere Zuwendung nicht nutzbar. In der Regel gibt es Bauanleitung und weitere Dateien als GitHub-Projekt. Dort finden sich dann auch Einträge im Bugtracker, sodass man sich schon vor dem Herunterladen und Ausprobieren ein erstes Bild machen kann.



Die Suchergebnisse auf der Kommandozeile trügen: Wie viele Images es zu einem Suchwort wirklich im Docker-Hub gibt, verrät nur die Web-Site.

Image versus Container

? Welcher Begriff ist denn jetzt eigentlich der richtige: Container oder Image?

! Das kommt auf den Kontext an: Container ist zunächst ein Gattungsbegriff für eine Technik. Ein Docker-Image ist letztlich ein Dateibündel aus dem auszuführenden Programm und von ihm benötigten Komponenten. Das Docker-Image ist die Kopiervorlage für einen Container, also das Starten eines isolierten Prozesses – sozusagen einer Container-Instanz. Images baut man oder lädt sie aus einer Registry herunter. Container konfiguriert und startet man.

Datenhaltung für Container

? Container gelten als Wegwerfware. Wo lassen sie dann ihre Daten?

! Beim Starten von Containern ist es üblich, für eine Trennung der Nutz- und Konfigurationsdaten von den übrigen Dateien zu sorgen. In der Regel liegen die Nutz- und Konfigurationsdaten auf einem oder auch mehreren separaten Volumes, die auch über Updates hinweg erhalten bleiben. In einer Standardinstallation sind Volumes letztlich Verweise ins Dateisystem des Hosts. Beim Aktualisieren wird der Container gestoppt, ein neues Image geladen und ein neuer Container gestartet, der wieder Zugriff auf die erhaltenen gebliebenen Volumes erhält. Ein Watchtower-Container kann das automatisch erledigen.

Sicherheit von Container-Images

? Wie gelangen Sicherheitsupdates in einen Container?

! Dafür gibt es verschiedene Strategien. Spätestens, wenn ein neues Image da ist, sollte ein Container ausgetauscht werden. Oft krankt das aber an der Art, wie neue Images entstehen: Nur das Aktualisieren einer Container-Bauanleitung löst den Neubau aus, nicht aber Änderungen am Basis-Image – wo eher Sicherheitsupdates anfallen. So jedenfalls hat Docker den Hub angelegt. Deshalb gibt es Container, die Sicherheitsupdates über die Mechanismen der verwendeten Distribution einspielen.

Dieses Vorgehen widerspricht aber eigentlich der Idee, in einem Container ausschließlich Software für die jeweilige Aufgabe und keine weitere zu verpacken. Letztlich ist das Problem der Updates nicht zufriedenstellend gelöst, weil es an den Menschen hängen bliebt, die die Images bauen. Hinweise auf eventuelle Sicherheitslücken in den Images, die es einst für angemeldete Nutzer auf dem Docker-Hub gab, versteckt Docker jetzt im Store. *(ps@ct.de)*

Dokumentation erwähnter Container:
ct.de/yj4x