

Register in Docker

In Docker-Registries (dt: Register) kann man Images hochladen um sie öffentlich mit anderen Personen zu teilen oder privat mit verschiedenen Geräten. Ein Beispiel für ein bereits bekanntes Image aus einem Register wäre `mysql`. Dabei fügt Docker jedoch implizit noch einiges an Zusatzinformation hinzu. Eine längere Schreibweise wäre: `docker.io/library/mysql`. `docker.io` ist das Standardregister welches verwendet wird wenn kein anderes angegeben ist. Häufige Alternativen wären `registry.gitlab.com` oder `ghcr.io` von Github. `library` ist der Username welcher das Image veröffentlicht hat. Auch dieser ist der Standardname wenn kein anderer angegeben wurde. Alle Container von diesem Usernamen wurden von der Docker Foundation veröffentlicht und geprüft. Auch tragen sie immer das Abzeichen `Docker official image`.

Selbst Images veröffentlichen

Docker Hub bietet einem die Möglichkeit dass man auf deren Plattform Images hochladen kann. Github und Gitlab verfolgen hier einen etwas anderen Ansatz. Neben der Bereitstellung von Git-Repositories bieten sie auch Container-Registries sowie eine Build-Infrastruktur an. Docker Hub bietet ausschließlich an bereits fertige Images zu veröffentlichen.

Erstelle dir für unsere einen User auf hub.docker.com. Dannach hast du die Möglichkeit lokale Images dort hochzuladen. Auf deinem Rechner musst du dich dann noch mit dem Kommando `docker login` in der Konsole einloggen. Username und Passwort sind die gleichen wie auf Docker Hub.

Zu dem bereits bekannten Kommando `docker pull` zum Herunterladen von Images gibt es auch ein Gegenstück. Mit `docker push` ist es möglich Images in ein Registry hochzuladen.

Um ein Image zu bauen muss man `docker build` ausführen. Dabei bekommt man ein Image mit einem Hashwert als Namen. Möchte man es auch benennen ist das mit der Option `-t localname` möglich. `t` steht für tag. Hat man ein benanntes lokales Image kann man dieses mit `docker push localname registry/username/imagename:tag` hochladen.

Veröffentlichtes Image in Compose verwenden

Um dein eigenes Image nicht immer dort bauen zu müssen wo du es brauchst kannst du in compose auf ein Register verweisen. Für unsere Datenbank haben wir das mit `image: mysql` bereits verwendet. Für deine Spring-Anwendung kannst du `build: ...` ersetzen mit `image: <dein-dockerhub-name>/<dein-imagename>`. Ist alles fertig aufgesetzt kannst du mit dieser Datei alleine deine Anwendung auf jedem Host mit Docker verwenden. Vorausgesetzt dein Image ist öffentlich zugänglich. Ansonsten müsstest du dich auch auf dem entsprechenden Rechner zuerst wieder mit `docker login` einloggen.

Quellen und weitere Infos

[Anleitung auf Stackoverflow](#)

[Offizielle Docker Images](#)

[Gitlabs Container Registry erklärt](#)

[GitHubs Container Registry erklärt](#)