



Tag 3: Docker, GitOps, Deployment-Strategien

10.07.2024, Daniel Krämer

© Copyright 2024 anderScore GmbH

HECKER
CONSULTING

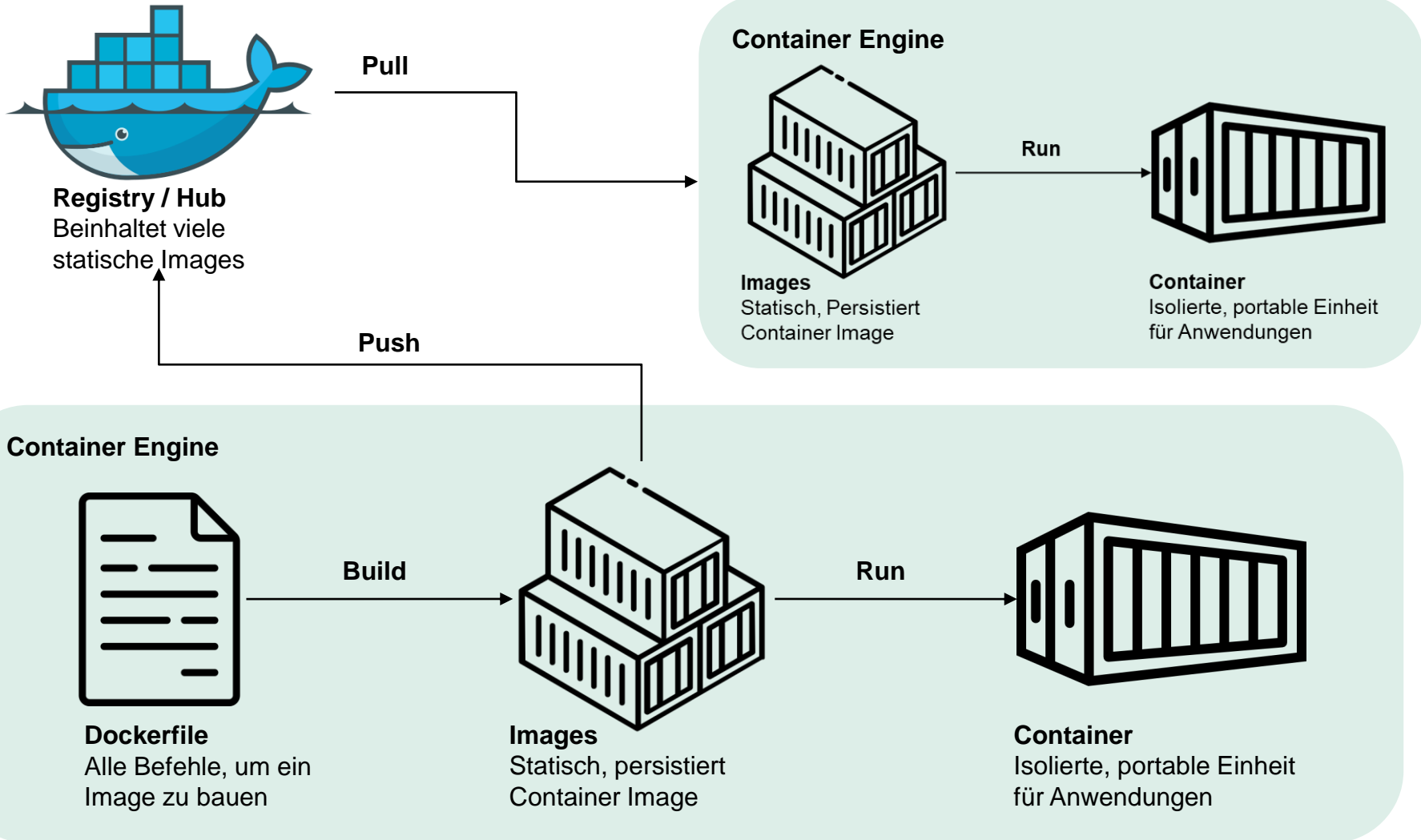
- **Tag 1 – Einführung in Git und GitLab**
 - Einführung & Kursüberblick
 - Grundlagen von Git
 - Git Rebase und Merge-Strategien
 - Git Remote
 - Grundlagen von GitLab
- **Tag 2 – Git-Workflows, CI/CD, GitLab CI**
 - Git-Workflow im Team
 - Gitflow-Workflow
 - Tags, Releases & deren Verwaltung
 - Einführung in GitLab CI/CD & gitlab-ci.yml
 - GitLab Runner
- **Tag 3 – Docker, GitOps, Deployment-Strategien**
 - Entwicklung mit Docker
 - Container/Docker-Registry
 - Erstellen von Release- und Tagged-Images
 - GitOps Grundlagen
 - Möglichkeiten des Deployments & Verwaltung von Konfiguration
 - Abschlussübung & Diskussion

- **Tag 1 – Einführung in Git und GitLab**
 - Einführung & Kursüberblick
 - Grundlagen von Git
 - Git Rebase und Merge-Strategien
 - Git Remote
 - Grundlagen von GitLab
- **Tag 2 – Git-Workflows, CI/CD, GitLab CI**
 - Git-Workflow im Team
 - Gitflow-Workflow
 - Tags, Releases & deren Verwaltung
 - Einführung in GitLab CI/CD & gitlab-ci.yml
 - GitLab Runner
- **Tag 3 – Docker, GitOps, Deployment-Strategien**
 - Entwicklung mit Docker
 - Container/Docker-Registry
 - Erstellen von Release- und Tagged-Images
 - GitOps Grundlagen
 - Möglichkeiten des Deployments & Verwaltung von Konfiguration
 - Abschlussübung & Diskussion

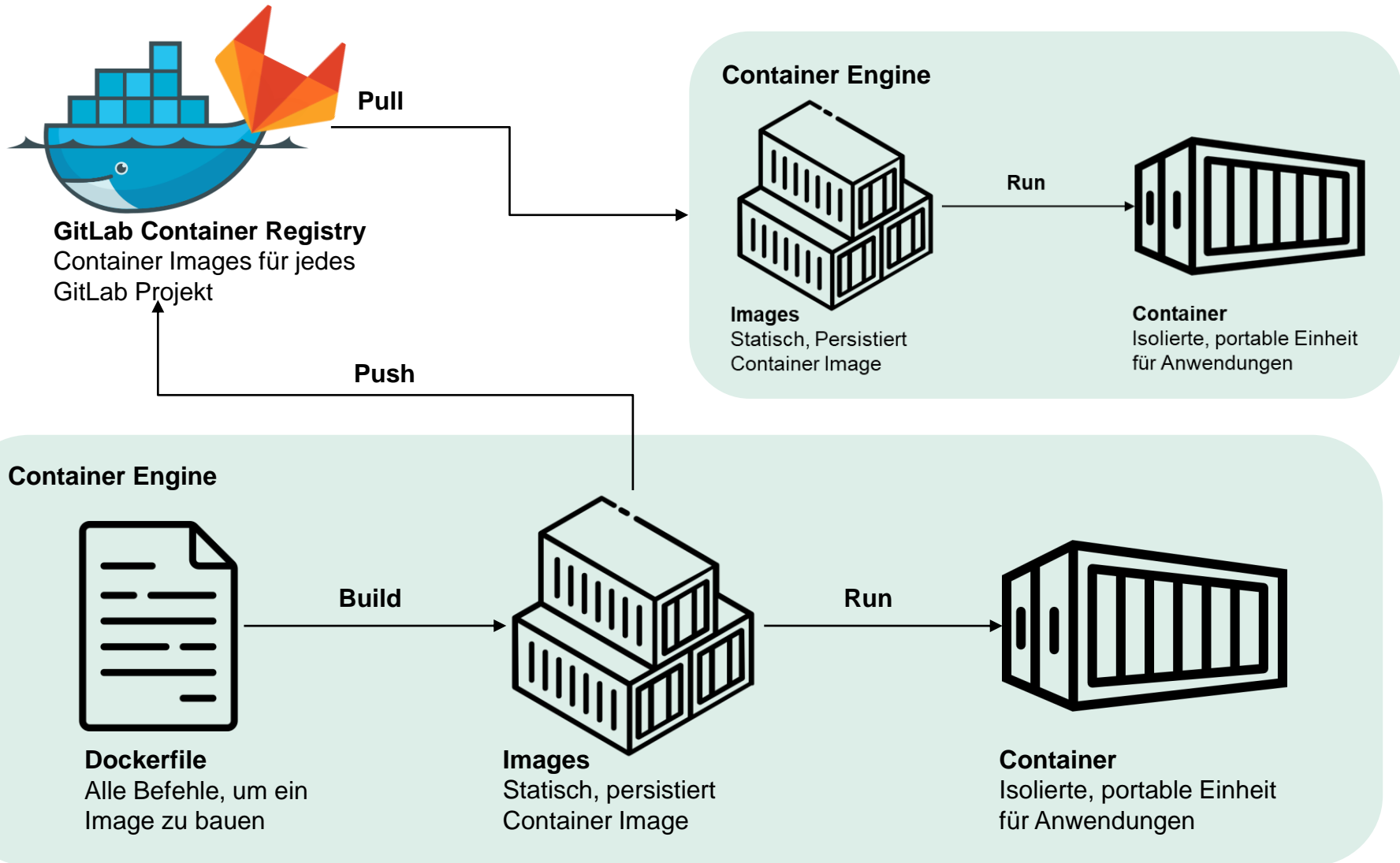
Grundlagen von

Container/Docker Registry

Container Registry



GitLab Container Registry



- DockerHub
 - Bekannteste, öffentliche Registry
 - Default Registry von Docker
- GitLab Container Registry
 - Integriert in GitLab, private Registry
 - Verwendung eigener Docker Images für CI Pipelines möglich
- Verwaltung / Aktivierung in GitLab
 - https://docs.gitlab.com/ee/administration/packages/container_registry.html
- GitLab Dependency Proxy
 - Caching von DockerHub Images
 - Vermeidet „rate limits“ und beschleunigt Pipelines

Unterstützte Image Formate

- Docker v2
- Open Container Initiative (OCI)
- Container Registry entspricht OCI Verteilungsspezifikation

Naming Convention für Container Images

- `<registry server>/<group>/<project>[/<optional path>]`
- `gitlab.example.com/mygroup/myproject`
- Zusätzliche Pfade am Ende eines Images erlaubt
 - Aber: nur bis zu zwei Ebenen tief!
- Beispiele:
 - `gitlab.ads.anderscore.com/trainings/gitlab:some-tag`
 - `gitlab.ads.anderscore.com/trainings/gitlab/image:latest`
 - `gitlab.ads.anderscore.com/trainings/gitlab/my/image:rc1`

Verschieben oder Umbenennen einer Container Registry

- Registry-Pfad matched zugehörigem Projekt
 - → Projekt verschieben oder umbenennen
- Von GitLab.com gehosteten Instanzen unterstützt
- Self-managed:
 - Alle Container Images vorher löschen
 - <https://gitlab.com/gitlab-org/gitlab/-/issues/18383#possible-workaround>
 - <https://gitlab.com/groups/gitlab-org/-/epics/9459>

Live Demo

- Container Registry ansehen
- Tags eines bestimmten Container Images ansehen
- Container Images von der Container Registry nutzen
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern



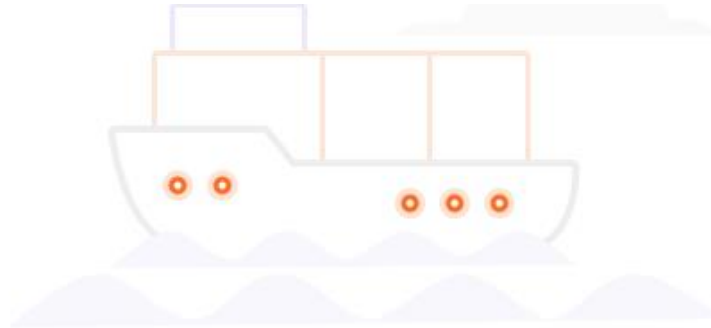
- Container Registry ansehen
- Tags eines bestimmten Container Image ansehen
- Container Images von der Container Registry nutzen
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern



Container Registry ansehen

1. Gewünschtes Projekt oder Gruppe in GitLab auswählen
 2. „Deploy“ → „Container Registry“
- Container Images
 - Suchen, Sortieren, Filtern und Löschen
 - View mit Filter teilen (URL kopieren)
 - Private Project
 - Nur Members des Projektes und der Gruppe haben Zugriff
 - Wenn das Projekt öffentlich ist, dann auch die Registry!





There are no container images stored for this project

With the Container Registry, every project can have its own space to store its Docker images. [More Information](#)

CLI Commands

If you are not already logged in, you need to authenticate to the Container Registry by using your GitLab username and password. If you have [Two-Factor Authentication](#) enabled, use a [Personal Access Token](#) instead of a password.

```
docker login gitlab.ads.anderscore.com:5006
```



You can add an image to this registry with the following commands:

```
docker build -t gitlab.ads.anderscore.com:5006/trainings/g
```



```
docker push gitlab.ads.anderscore.com:5006/trainings/gitla
```



Wir brauchen ein Docker Image!

1. Docker muss installiert sein
2. Mit Docker auf GitLab einloggen
 - Siehe vorangegangene Hinweise von GitLab!
 - Hier: `docker login gitlab.ads.anderscore.com:5006`
3. Lokales Dockerfile im Projekt erstellen
4. Docker Image bauen
 - `docker build -t gitlab.ads.anderscore.com:5006/trainings/gitlab .`
5. Docker Image in die GitLab Container Registry pushen
 - `docker push gitlab.ads.anderscore.com:5006/trainings/gitlab`



Dockerfile

```
FROM centos:7
LABEL maintainer="Patrick Ungewiß"
ARG TIMEZONE="Germany/Cologne"

# set a directory for the app
WORKDIR /usr/src/app

# copy all the files to the container
COPY . .

#update yum
RUN yum update -y && \
yum clean all

# installing sshd, httpd, sudo, openssl
RUN yum install -y \
openssh-server \
openssh-clients \
httpd \
```









```
httpd-tools \
sudo \
openssl && \
yum clean all

# installing more tools
RUN yum install -y \
git \
sed \
telnet \
vim \
unzip \
crontabs \
zip && \
yum clean all

# define the port number the container
should expose
EXPOSE 5000
```

DEMO

Project

-  GitLab
-  Pinned ▼
- Issues 17
- Merge requests 0
-
-  Manage >
-  Plan >
-  Code >
-  Build >
-  Secure >
-  Deploy ▼
- Releases
- Feature flags
- Package Registry
- Container Registry**

Container Registry

 1 Image repository  Cleanup is not scheduled.

trainings/gitlab 

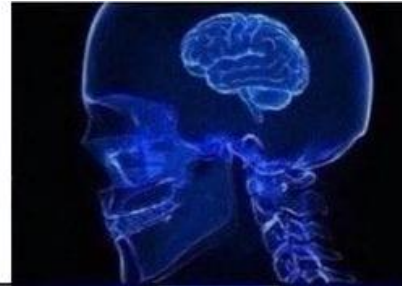
1 tag

DEMO

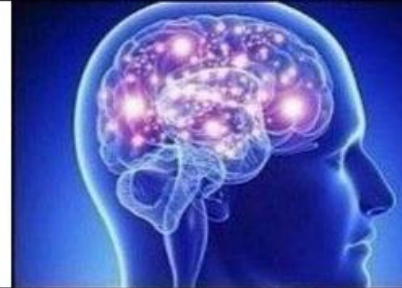
- Das Pushen hat lange gedauert...
 - → Größe des Images beachten!
- Bad Practice
 - Keine Version → z.B. docker
 - Latest tag → z.B. docker:latest
 - Stable tag → z.B. docker:stable
 - Major Version → z.B. docker:26
- Good Practice
 - Spezifische Version → z.B. docker:26.1.3-dind
 - --version nutzen → zuletzt funktionierende Version anzeigen



:latest



:1-stable



:1.41.1



@sha256:82becede498899e



imgflip.com

DEMO

- Container Registry ansehen
- Tags eines bestimmten Container Images ansehen
- Container Images von der Container Registry nutzen
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern

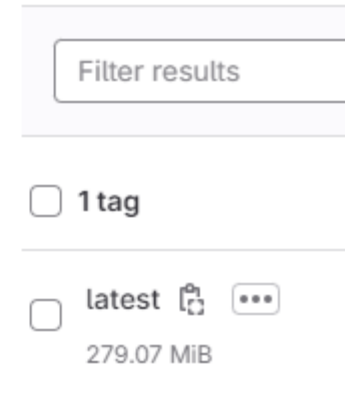
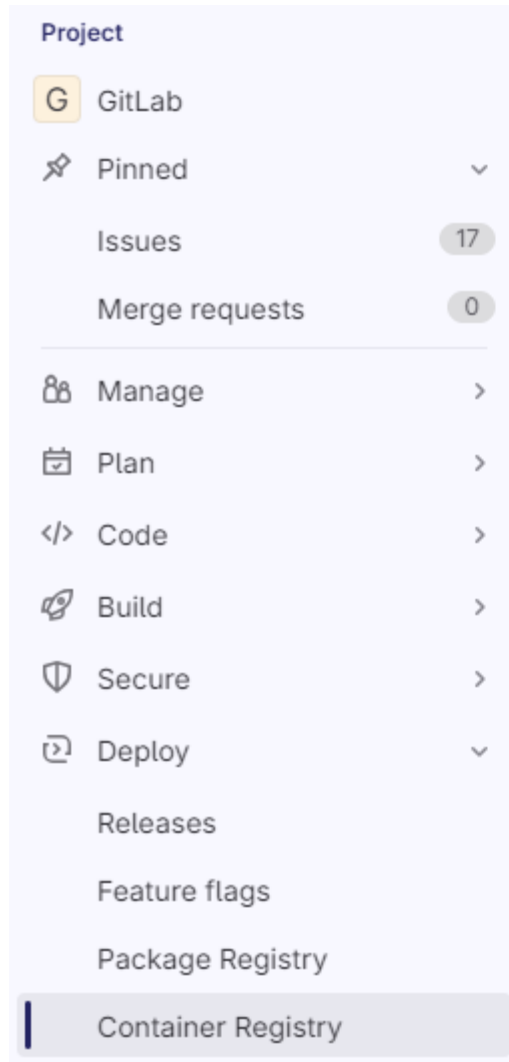


Tags eines bestimmten Container Images ansehen

- In GitLab kann man die „Tag Details“-Seite einsehen, um eine Liste der Tags zu erhalten, welche mit dem Image in Verbindung stehen
1. Gewünschtes Projekt oder Gruppe in GitLab auswählen
 2. „Deploy“ → „Container Registry“
 3. Gewünschtes Container Image selektieren
 4. „Tag-Details“-Seite wird angezeigt



Container Registry



DEMO

gitlab ⓘ

1 tag ⓘ Cleanup disabled ⓘ Created May 30, 2024 11:44

Filter results



Name ▾



☐ 1 tag

Delete selected

☐ latest ⓘ ⓘ

279.07 MiB

Published 21 minutes ago

Digest: 4a5f163 ⓘ

⌚ Published to the `trainings/gitlab` image repository at 13:44:14 GMT+0200 on 2024-05-30

📄 Manifest digest: `sha256:4a5f163f6dc759d64ed9d239af01ffe642f7d523263c557de9fbccdb883b3595` ⓘ


🔗 Configuration digest: `sha256:a481ef39210251411a17f981c877909880491081d71f824d6d260652667f568` ⓘ

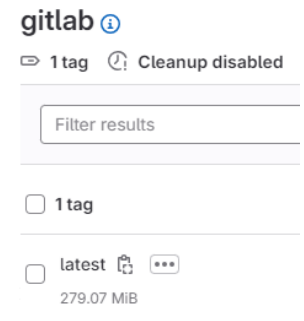


- Container Registry ansehen
- Tags eines bestimmten Container Images ansehen
- Container Images von der Container Registry nutzen
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern



Container Images von der Container Registry nutzen

- Um ein Container Image aus der GitLab Container Registry herunterzuladen und nutzen zu können:
 1. Gewünschtes Projekt oder Gruppe in GitLab auswählen
 2. „Deploy“ → „Container Registry“
 3. Gewünschtes Container Image auswählen und „Copy image path“  auswählen
 4. „docker run“ mit dem Link ausführen
 - `docker run [options]`
`gitlab.ads.anderscore.com:5006/trainings/gitlab:latest`
`[arguments]`
- `docker ps -a` sollte nun einen weiteren Container anzeigen



- Container Registry ansehen
- Tags eines bestimmten Container Images ansehen
- Container Images von der Container Registry nutzen
- Naming Convention für Container Images
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern



Container Registry für ein Projekt deaktivieren

- Default: enabled
1. Gewünschtes Projekt in GitLab auswählen
 2. „Settings“ → „General“
 3. „Visibility, project features, permissions“ aufklappen
 4. „Container registry“ deaktivieren
 5. „Save changes“ auswählen
 6. Daraufhin wird „Deploy“ → „Container Registry“ aus der linken Sidebar entfernt



- Container Registry ansehen
- Tags eines bestimmten Container Image ansehen
- Container Images von der Container Registry nutzen
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern



Sichtbarkeit der Container Registry ändern

- Default: Für jeden sichtbar mit Zugriff aufs Projekt
- Sichtbarkeit kann jedoch pro Projekt geändert werden

1. Gewünschtes Projekt in GitLab auswählen
2. „Settings“ > „General“
3. „Visibility, project features, permissions“ aufklappen
4. Unter „Container registry“ eine Auswahl treffen
 - „Everyone With Access“ (Default)
 - Container Registry hat das Sichtbarkeitslevel des Projektes!
 - „Only Project Members“
 - Vergleichbar mit: private project + everyone with access
- „Save changes“ auswählen

