

The background of the slide features a faded image of a multi-story brick building on the left and a church with a tall, ornate stone tower on the right. A large, white, stylized 'S' shape is overlaid on the right side of the image.

Tag 3: GitOps, Docker in der Entwicklung und Deployment-Strategien

19.06.2024, Daniel Krämer & Malte Fischer

© Copyright 2024 anderScore GmbH

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

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

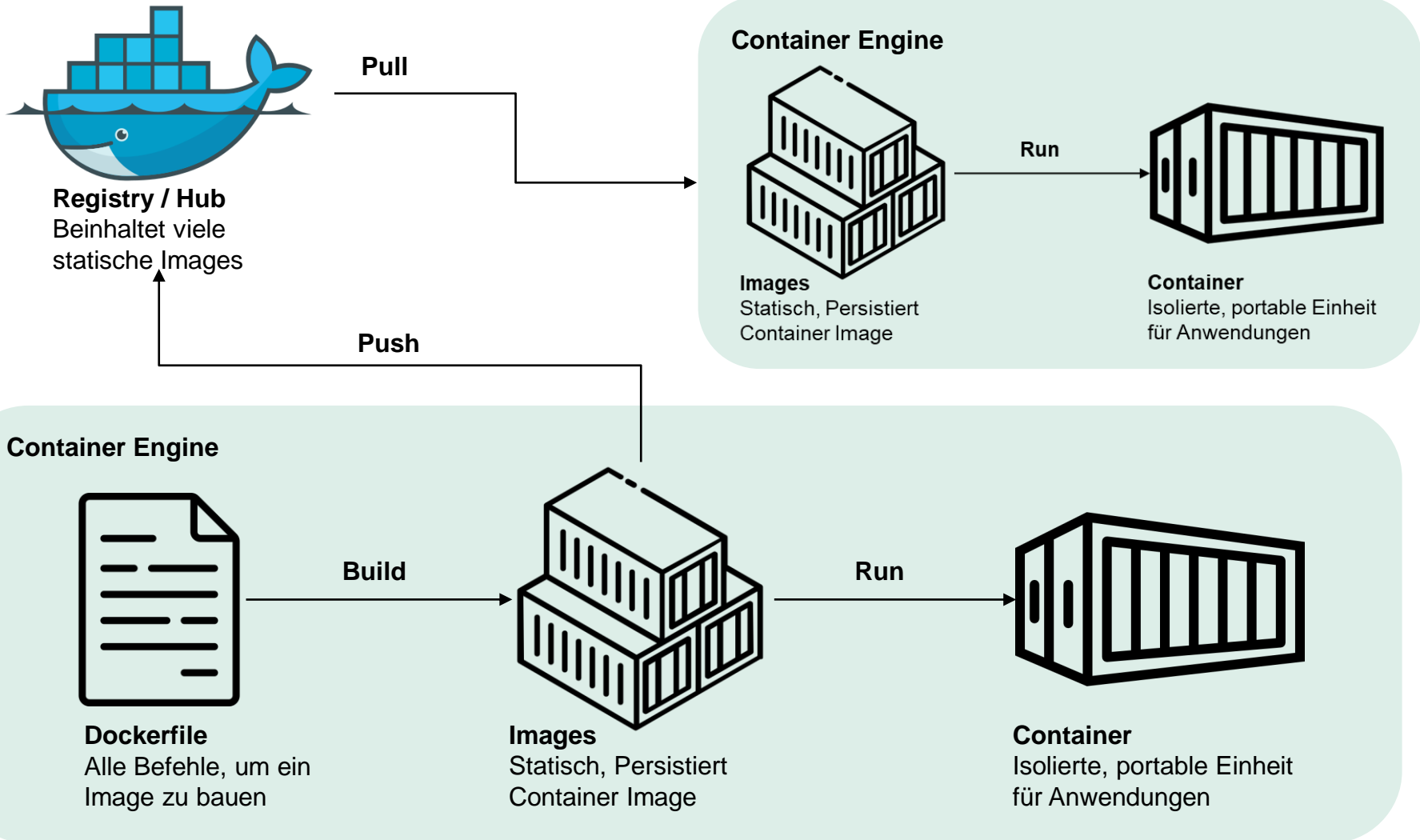
Grundlagen von

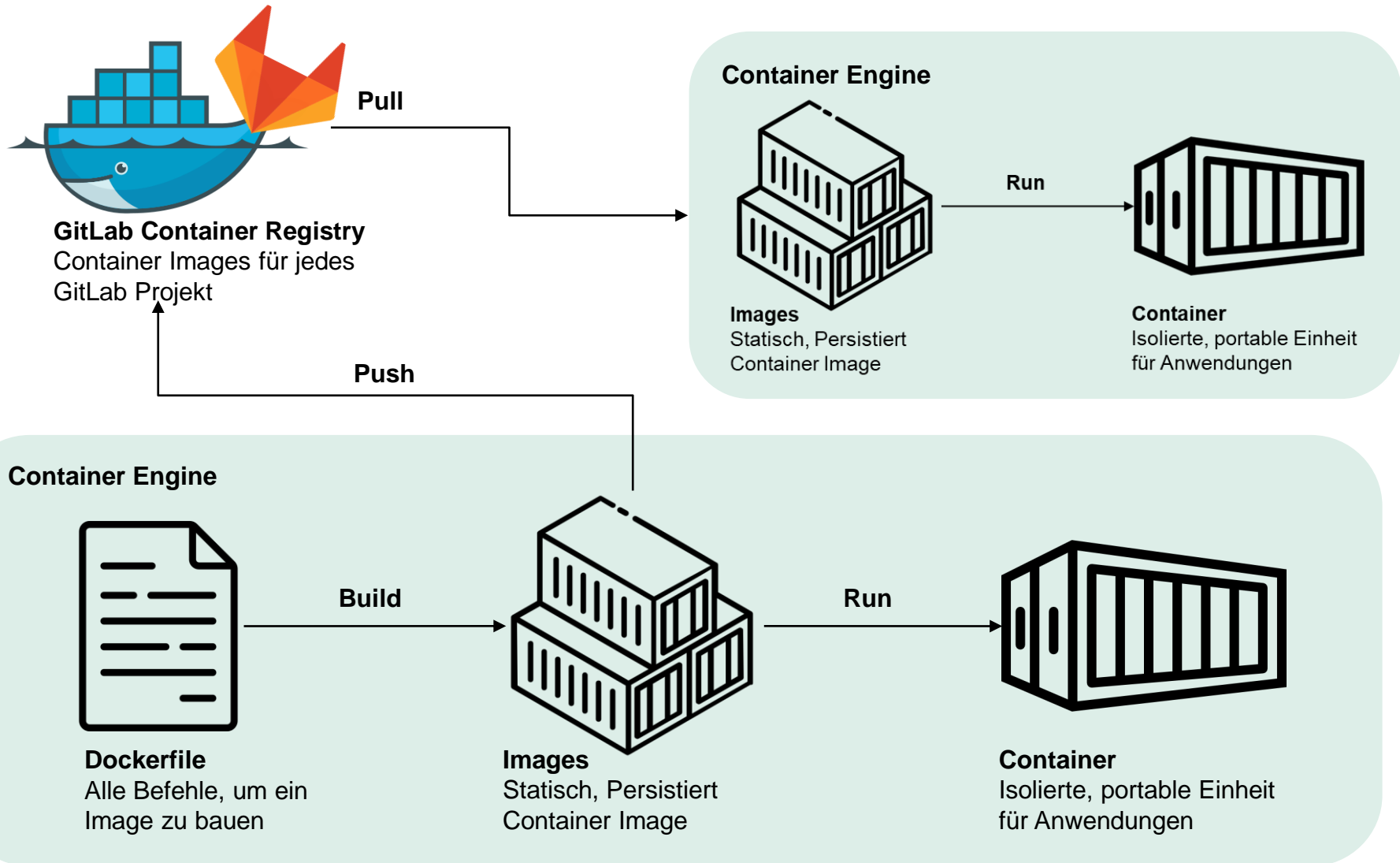
Container/Docker Registry





Container Registry





- Dockerhub
 - öffentliche Registry
- GitLab Container Registry
 - Integriert in GitLab, private Registry
 - Speichert Docker-Anwendungen
 - Eigenes Docker Image innerhalb GitLab CI Pipeline verwenden
- Verwaltung / Aktivierung in GitLab
 - https://docs.gitlab.com/ee/administration/packages/container_registry.html
- GitLab Dependency Proxy
 - Falls Images von Docker Hub
 - Vermeidet „rate limits“ und beschleunigt Pipelines

Live Demo

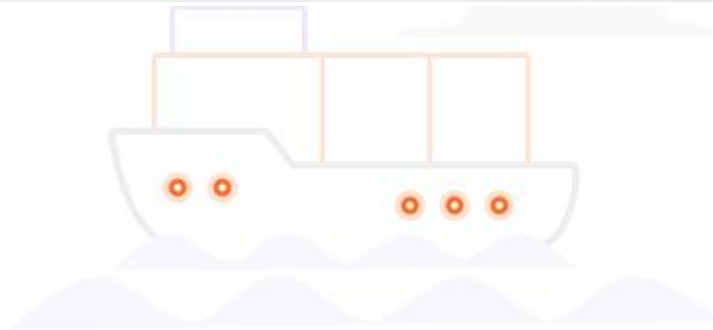
- Container Registry ansehen
- Tags eines bestimmten Container Image ansehen
- Container Images von der Container Registry nutzen
- Naming Convention für Container Images
- Verschieben oder Umbenennen einer Container Registry
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern



- Container Registry ansehen
- Tags eines bestimmten Container Image ansehen
- Container Images von der Container Registry nutzen
- Naming Convention für Container Images
- Verschieben oder Umbenennen einer Container Registry
- 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 Projekts 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 Vorschlagstext 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
```

Project

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

Container Registry

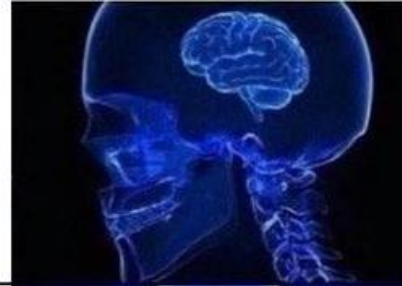
 1 Image repository  Cleanup is not scheduled.

trainings/gitlab 

1 tag

- Das Pushen hat lange gedauert...
 - → Größe des Images beachten!
- Bad practise
 - :latest bei einem Docker Image zu verwenden
 - Weil sich dadurch das Image immer ändert! (durch die neue Version)
 - Keine Version → z.B. docker
 - Das latest tag → z.B. docker:latest
 - Das stable tag → z.B. docker:stable
 - Major version → z.B. docker:26
- Good practise
 - Spezifische Version nutzen → z.B. docker:26.1.3-dind
 - --version nutzen, um ggf. die zuletzt funktionierende Version anzuzeigen

:latest



:1-stable



:1.41.1



@sha256:82becede498899e

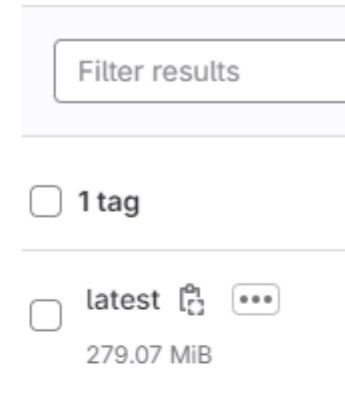
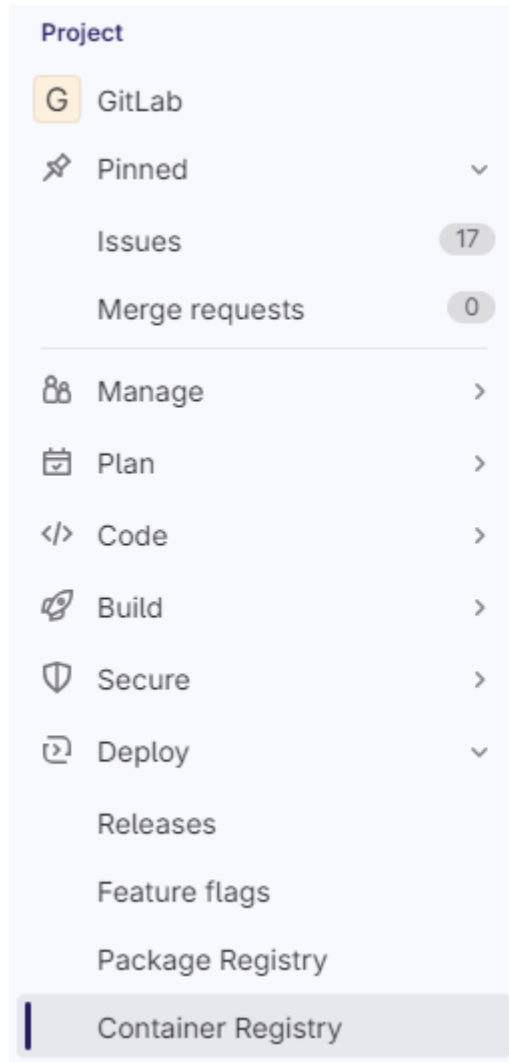


imgflip.com

- Container Registry ansehen
- Tags eines bestimmten Container Image ansehen
- Container Images von der Container Registry nutzen
- Naming Convention für Container Images
- Verschieben oder Umbenennen einer Container Registry
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern

Tags eines bestimmten Container Image 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



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

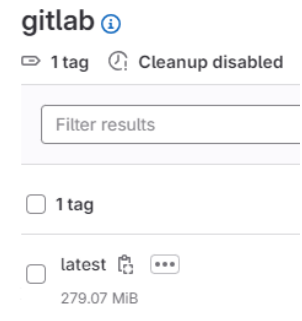
📄 Manifest digest: `sha256:4a5f163f6dc759d64ed9d239af01ffe642f7d523263c557de9fbccdb883b3595` ⓘ

🔗 Configuration digest: `sha256:a481ef39210251411a17f981c877909880491081d71f824d6d260652667f568` ⓘ

- Container Registry ansehen
- Tags eines bestimmten Container Image ansehen
- Container Images von der Container Registry nutzen
- Naming Convention für Container Images
- Verschieben oder Umbenennen einer Container Registry
- 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 kopierten 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 Image ansehen
- Container Images von der Container Registry nutzen
- Naming Convention für Container Images
- Verschieben oder Umbenennen einer Container Registry
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern

Naming Convention für Container Images

- `<registry server>/<namespace>/<project>[/<optional path>]`
- `gitlab.example.com/mynamespace/myproject`
- Project: `gitlab.ads.anderscore.com/trainings/gitlab`
- Image → `gitlab.ads.anderscore.com/trainings/gitlab`
- Zusätzliche Namen ans Ende des Images sind 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`

- Container Registry ansehen
- Tags eines bestimmten Container Image ansehen
- Container Images von der Container Registry nutzen
- Naming Convention für Container Images
- Verschieben oder Umbenennen einer Container Registry
- Container Registry für ein Projekt deaktivieren
- Sichtbarkeit der Container Registry ändern

Verschieben oder Umbenennen einer Container Registry

- Der Pfad der Registry matched immer dem zugehörigen Projekt
 - Daher muss man entweder das Projekt verschieben oder umbenennen
- Wird von GitLab.com gehosteten Instanzen unterstützt
- Self-managed:
 - Alle Container Images müssen vorher gelöscht werden
 - <https://gitlab.com/gitlab-org/gitlab/-/issues/18383#possible-workaround>
 - <https://gitlab.com/groups/gitlab-org/-/epics/9459>

- Container Registry ansehen
- Tags eines bestimmten Container Image ansehen
- Container Images von der Container Registry nutzen
- Naming Convention für Container Images
- Verschieben oder Umbenennen einer Container Registry
- 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
- Naming Convention für Container Images
- Verschieben oder Umbenennen einer Container Registry
- 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 Projekts!
 - „Only Project Members“
 - Vergleichbar mit: private project + everyone with access
- „Save changes“ auswählen

Unterstützte Image Formate

- Docker V2
- Open Container Initiative (OCI)
- Container Registry entspricht OCI-Verteilungsspezifikation