





18.06.2024, Daniel Krämer & Malte Fischer

© Copyright 2024 anderScore GmbH

### Agenda



#### 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

### Agenda



#### 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



Git

# Tags

#### Tags & Releases



#### **Inhalt**

- Tags in Git
  - Tagging Konzept
  - Erstellen von Git Tags
  - Annotated Tags
- Tags in GitLab
- GitLab Releases



- Tagging ist ein wichtiges Konzept der Versionverwaltung
- Wird verwendet, um einen spezifischen Entwicklungsstand zu markieren und mit einem Label zu versehen
- Entwicklungsstand wird als "snapshot" bezeichnet
- Einsatz häufig, um spezifische Releases in einem Projekt zu markieren



 Mit den Befehl git tag <tag-name> lässt sich zum letzten Commits des aktiven Branches ein Tag hinzufügen

```
$ git log --oneline
c3a059e (HEAD -> main) Add final_file
a908be7 Add file3
722eaf0 Add file from main
c61ef14 Initial commit

$ git tag "release-v0.0.1"

$ git log --oneline
c3a059e (HEAD -> main, tag: release-v0.0.1) Add final_file
a908be7 Add file3
722eaf0 Add file from main
c61ef14 Initial commit
```

- Fügt leichtgewichtigen Tag hinzu
- Tag wird an allen Files im Repository gesetzt, nicht nur solche, die beim Commit verändert wurden



git tag zeigt alle Tags im aktuellen Projekt

```
$ git log --oneline
89c1108 (HEAD -> main, tag: release-v0.0.2) Add release2_file
c3a059e (tag: release-v0.0.1) Add final_file
a908be7 Add file3
722eaf0 Add file from main
c61ef14 Initial commit

$ git tag release-0.0.0 a908be7

$ git log --oneline
89c1108 (HEAD -> main, tag: release-v0.0.2) Add release2_file
c3a059e (tag: release-v0.0.1) Add final_file
a908be7 (tag: release-0.0.0) Add file3
722eaf0 Add file from main
c61ef14 Initial commit
```



git show <tag-name> zeigt den Commit zum angegebenen Tag

```
$ git show release-v0.0.1
commit c3a059e2d6e4acfdca1295104b4d6a55f8bd00e4 (tag: release-v0.0.1)
Author: Example User <example.user@example.de>
Date: Tue May 28 14:47:28 2024 +0200

Add final_file

diff --git a/final_file.txt b/final_file.txt
new file mode 100644
index 0000000..e69de29
```



Rückwirkendes Tagging ist mit Angabe der Commit-ID möglich

```
$ git log --oneline
89c1108 (HEAD -> main, tag: release-v0.0.2) Add release2_file
c3a059e (tag: release-v0.0.1) Add final_file
a908be7 Add file3
722eaf0 Add file from main
c61ef14 Initial commit

$ git tag release-v0.0.0 a908be7

$ git log --oneline
89c1108 (HEAD -> main, tag: release-v0.0.2) Add release2_file
c3a059e (tag: release-v0.0.1) Add final_file
a908be7 (tag: release-v0.0.0) Add file3
722eaf0 Add file from main
c61ef14 Initial commit
```



git checkout <tag-name> ermöglicht das Auschecken eines getaggten Commit

```
$ git checkout release-v0.0.1
Note: switching to 'release-v0.0.1'.
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:
  git switch -c <new-branch-name>
Or undo this operation with:
  git switch -
Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at c3a059e Add final file
```



 git checkout -b <branch-name> <tag-name> ermöglicht das Erstellen eines Branches abzweigend vom getaggten Commit

```
$ git checkout -b hotfix release-v0.0.1
Switched to a new branch 'hotfix'
```

 Löschen eines Tags ist mit dem Befehl git tag -d <tag-name> möglich

```
$ git tag -d release-v0.0.0
Deleted tag 'release-v0.0.0' (was a908be7)
```



#### **Annotated und Lightweight Tags**

- Lightweight Tags geben dem jeweiligen Commit nur ein Alias
- Annotated Tags sind schwergewichtiger und enthalten daher mehr Informationen
  - Tagger Name
  - Tagger E-Mail
  - Datum
  - Tagging Message
  - Checksumme
  - Signierung
- Sinnvoll bei Mulit-Developer oder Mulit-Repository Umgebungen, um bspw. Releases zu markieren



#### Annotated Tag erstellen mittels

```
git tag -a <tag-name>
```

```
$ git tag -a release-v1.0.0 -m "Latest v1 release"

$ git show release-v1.0.0
tag release-v1.0.0
Tagger: Christopher Keutner <christopher.keutner@alumni.fh-aachen.de>
Date: Tue May 28 15:13:44 2024 +0200

Latest v1 release

commit 11f6b62f8bc7a3403ceae4e8923d7a54abab7b6d (HEAD -> main, tag: release-v1.0.0)
Author: Christopher Keutner <christopher.keutner@alumni.fh-aachen.de>
Date: Tue May 28 15:13:10 2024 +0200

Add new file

diff --git a/file8.txt b/file8.txt
new file mode 100644
index 0000000.e69de29
```



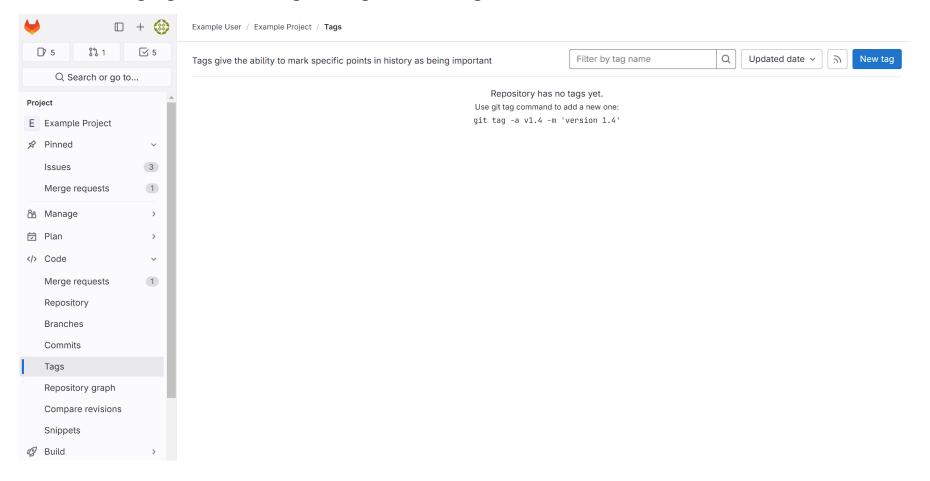
Git

## Tags in GitLab

### Tags in GitLab



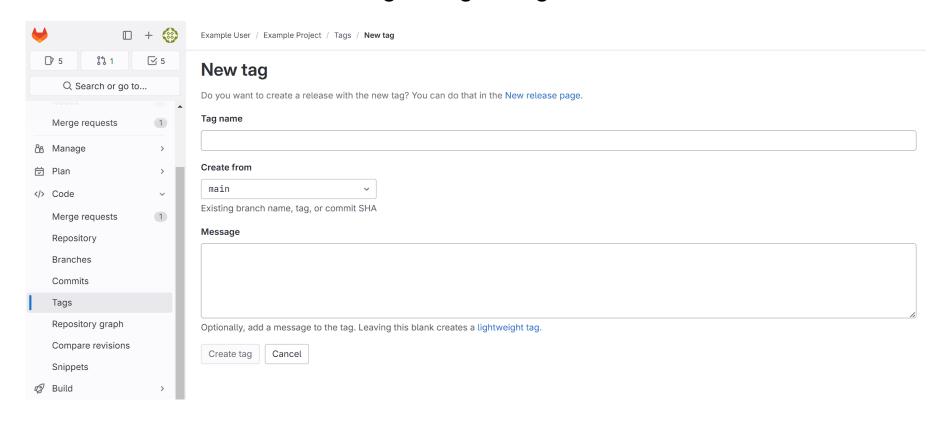
- Tags können ebenfalls in GitLab zum Repository hinzugefügt werden
- Erzeugt gleichwertige Tags im Vergleich zu Command Line



### Tags in GitLab



- Tags können auch hier aus Commits, Branches oder anderen Tags erstellt werden
- Ohne Nachricht wird ein Lightweight Tag erstellt





GitLab

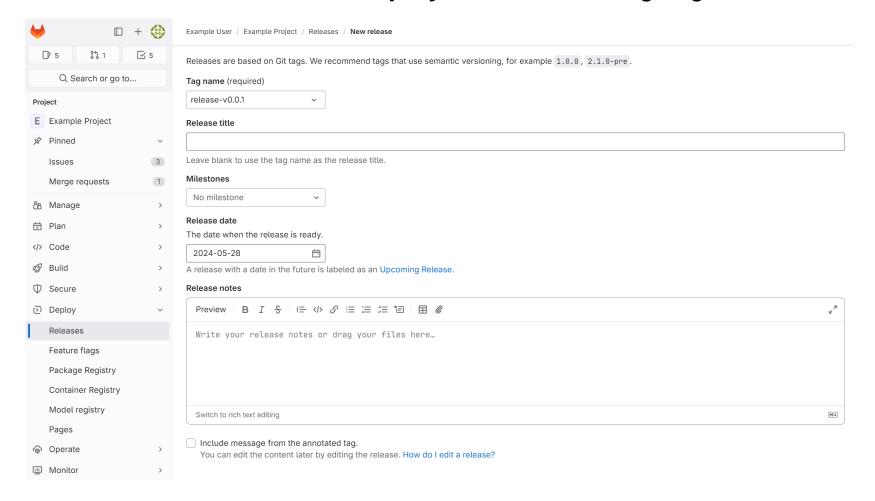
### Releases



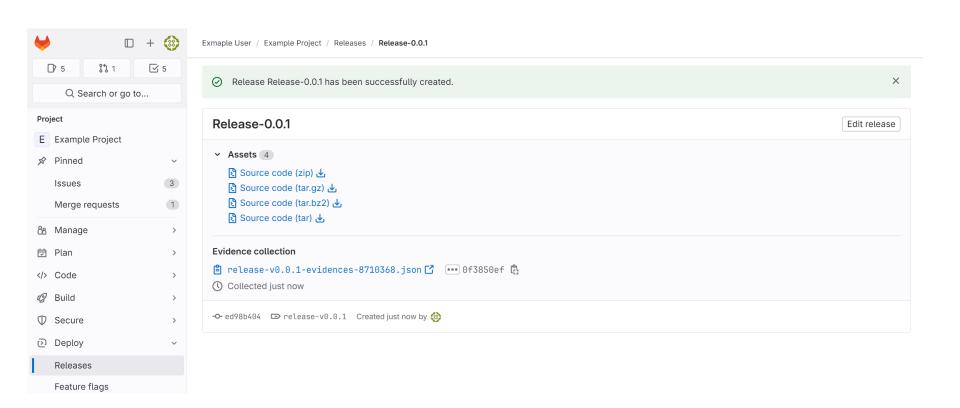
- GitLab Releases erstellen einen Snapshot des aktuellen Projektes
- Werden aus einem bestimmten Tag erstellt
- Ein GitLab Release kann beinhalten
  - Einen Snapshot des Source Codes im Repository
  - Packages aus Job Artefakten
  - Metadaten
  - Release Notes
- GitLab archiviert beim Erstellen eines Releases automatisch den Source Code und verknüpft diesen mit dem Release
- GitLab erstellt ebenfalls eine JSON Datei, welche den gesamten Release Inhalt auflistet (release evidence)
- Löschung des Tags führt auch zur Löschung des Releases



#### Releases können unter Deploy → Releases angelegt werden









Releases werden in der Projekt-Übersicht angezeigt

