



Tag 2: Vertiefung Git-Workflow, CI/CD & GitLab CI

18.06.2024, 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

Git Tags

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

The screenshot displays the GitLab web interface for a project named 'Example Project'. The left sidebar contains a navigation menu with the following items: Project, E Example Project, Pinned, Issues (3), Merge requests (1), Manage, Plan, Code, Merge requests (1), Repository, Branches, Commits, Tags (selected), Repository graph, Compare revisions, Snippets, and Build. The main content area shows the 'Tags' page. At the top, there is a breadcrumb trail: 'Example User / Example Project / Tags'. Below this, there is a search bar and a 'New tag' button. The main message states: 'Repository has no tags yet. Use git tag command to add a new one: git tag -a v1.4 -m 'version 1.4''.

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

Example User / Example Project / Tags / New tag

New tag

Do you want to create a release with the new tag? You can do that in the [New release page](#).

Tag name

Create from

main

Existing branch name, tag, or commit SHA

Message

Optionally, add a message to the tag. Leaving this blank creates a [lightweight tag](#).

Create tag Cancel

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

- 5

1

5

Search or go to...

Project

E

Example Project

Pinned

Issues

3

Merge requests

1

Manage

>

Plan

>

Code

>

Build

>

Secure

>

Deploy

>

Releases

Feature flags

Package Registry

Container Registry

Model registry

Pages

Operate

>

Monitor

>

Example User / Example Project / Releases / **New release**

Releases are based on Git tags. We recommend tags that use semantic versioning, for example `1.0.0`, `2.1.0-pre`.

Tag name (required)

release-v0.0.1

Release title

Leave blank to use the tag name as the release title.

Milestones

No milestone

Release date

The date when the release is ready.

2024-05-28

A release with a date in the future is labeled as an [Upcoming Release](#).

Release notes





Preview | **B** | *I* | U | ~~ABC~~ | `</>` | 🔗 | ☰ | ☷ | 📎 | 📄 | ✎

Write your release notes or drag your files here...

Switch to rich text editing

☐ Include message from the annotated tag.

You can edit the content later by editing the release. [How do I edit a release?](#)

5

1


5

Q Search or go to...

Project


E


Example Project


 Pinned


Issues 3


Merge requests 1


 Manage

 Plan

 Code

 Build

 Secure

 Deploy

Releases

Feature flags









Example User / Example Project / Releases / **Release-0.0.1**

✓ Release Release-0.0.1 has been successfully created. ✕





Release-0.0.1

[Edit release](#)

Assets 4

-  [Source code \(zip\)](#) 
-  [Source code \(tar.gz\)](#) 
-  [Source code \(tar.bz2\)](#) 
-  [Source code \(tar\)](#) 

Evidence collection

 [release-v0.0.1-evidences-8710368.json](#)   0f3850ef 

 Collected just now

 ed98b404  release-v0.0.1 Created just now by 

- Releases werden in der Projekt-Übersicht angezeigt

The screenshot shows the GitLab interface for a project named 'Example Project'. The left sidebar contains navigation links for Project, Pinned, Issues (3), Merge requests (1), Manage, Plan, Code, Build, Secure, Deploy, Operate, Monitor, Analyze, and Settings. The main content area displays the project name, a recent commit 'Update conflict_file.txt' by Example User, and a table of files. The 'Project information' sidebar on the right shows 4 commits, 3 branches, 1 tag, 5 KiB project storage, and 1 release (highlighted with a red box). Below this are links to add various project elements like LICENSE, CHANGELOG, CONTRIBUTING, etc. The bottom section shows the 'Getting started' guide for the project.

Exmable User / Example Project

Example Project

main example-project / +

History Find file Edit Code

Update conflict_file.txt
Example User authored 13 hours ago
ed98b404

Name	Last commit	Last update
README.md	Initial commit	1 day ago
conflict_file.txt	Update conflict_file.txt	13 hours ago
main_file1.txt	Add new file	21 hours ago

Project information

- 4 Commits
- 3 Branches
- 1 Tag
- 5 KiB Project Storage
- 1 Release**

README

- + Add LICENSE
- + Add CHANGELOG
- + Add CONTRIBUTING
- + Enable Auto DevOps
- + Add Kubernetes cluster
- + Set up CI/CD
- + Add Wiki
- + Configure Integrations

Example Project

Getting started

To make it easy for you to get started with GitLab, here's a list of recommended next steps.

Already a pro? Just edit this README.md and make it your own. Want to make it easy? [Use the template at the bottom!](#)

Created on
May 27, 2024