dataport

anderScore
trust in competence

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

18.06.2024, Daniel Krämer & Malte Fischer

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

anderScore trust in competence

Inhalt

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

Git – Tags

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

Git – Tags

- `git tag <tag-name>` um zum letzten Commits des aktiven Branches ein Tag hinzuzufügen

```
$ 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

Git – Tags

- `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 – Tags

- `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
```

Git – Tags

- 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 – Tags

- `git checkout <tag-name>` ermöglicht 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 – Tags

- `git checkout –b <branch-name> <tag-name>` ermöglicht 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 `git tag -d <tag-name>`

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

Git – Tags

anderScore trust in competence

Annotated und Lightweight Tags

- Lightweight Tags geben Commit nur ein Alias
- Annotated Tags enthalten mehr Informationen
  - Tagger Name
  - Tagger E-Mail
  - Datum
  - Tagging Message
  - Checksumme
  - Signierung
- Annotated Tags sinnvoll bei Mulit-Developer oder Mulit-Repository Umgebungen

Git – Tags

- Annotated Tag erstellen

```
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

GitLab

Releases

GitLab Releases

- GitLab Releases erstellen Snapshot des aktuellen Projektes
- Werden aus Tag heraus erstellt
- GitLab Release kann beinhalten
  - Einen Snapshot des Source Codes im Repository
  - Packages aus Job Artefakten
  - Metadaten
  - Release Notes
- GitLab archiviert Source Code und verknüpft diesen mit Release
- GitLab erstellt JSON Datei, mit Auflistung von Inhalt (release evidence)
- Löschung von Tags führt zur Löschung des Releases

Releases in GitLab