



Tag 1: Einführung in Git und GitLab, Git-Workflow im Team

17.06.2024, Daniel Krämer & Malte Fischer

© Copyright 2024 anderScore GmbH

HECKER
CONSULTING

- **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

Einführung in **GitLab**

Inhalt

- Einführung
 - Was ist GitLab?
 - Git vs. GitLab
 - Versionen
- Live Demo
 - Projekte erstellen
 - Gruppen
 - Arbeiten im Projekt
 - Planing Tools
 - Branches und Merge Requests
- Übung

Was ist GitLab?

- 2011 von Dimitri Saparoschez und Valery Sizov entwickelt
- Service zur Softwareentwicklung und Versionsverwaltung
- Hosting von Git Remote Repositories als Projekte
- Zusätzliche Funktionen:
 - Issue Tracking: Verwaltung/Verfolgung von Aufgaben
 - Dokumentation
 - CI/CD (Continuous Integration/Continuous Deployment)



Git vs. GitLab

Git

- Verteiltes Versionskontrollsystem zur Verwaltung von Dateien
- Verfolgung und Protokollierung von Änderungen

GitLab

- Umfassende Entwicklungs- und DevOps-Plattform
- Erweitert Git um Tools und Funktionalitäten für Projektmanagement, Zusammenarbeit und Automatisierung
- Zentrale Plattform zur Verwaltung von Projekten

Versionen

- Free
 - Kostenlos und Open Source
 - Konzipiert für persönliche Projekte
 - 400 compute minutes für CI/CD
 - Support nur über GitLab Forum
- Premium
 - Für mittlere bis große Teams
 - Fortgeschrittene Features im Bereich Projektmanagement, Code Reviews sowie CI/CD
 - 10.000 compute minutes für CI/CD
 - Support bei Problemen

- Ultimate
 - Für Unternehmen und große Organisationen
 - Umfangreiche Statistiken zur Analyse
 - Erweiterte Funktionen insbesondere im Bereich Security und Automatisierung
 - 50.000 compute minutes für CI/CD
- Verwendung als SaaS (GitLab.com) oder selbst gehostete Instanz im eigenen Netzwerk
- Zusätzliche Add-ons mit Fokus auf AI-Unterstützung für Premium und Ultimate



GitLab

Startseite und Projekte



📁 Your work / Projects / New project

Create new project



Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.



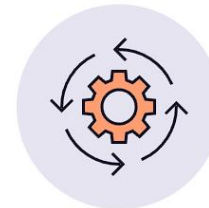
Create from template

Create a project pre-populated with the necessary files to get you started quickly.



Import project

Migrate your data from an external source like GitHub, Bitbucket, or another instance of GitLab.



Run CI/CD for external repository

Connect your external repository to GitLab CI/CD.

You can also create a project from the command line. [Show command](#)



 Your work / Projects / New project / Create blank project



Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

Project slug

Visibility Level

☒  Private

Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

☐  Internal

The project can be accessed by any logged in user except external users.

☐  Public

The project can be accessed without any authentication.

Project Configuration

☒ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

☐ Enable Static Application Security Testing (SAST)

Analyze your source code for known security vulnerabilities. [Learn more.](#)

Create project

Cancel



Neues Projekt erstellen


- URL
 - Project URL + Project Slug bestimmen die URL, unter der das neue Projekt aufgerufen werden kann
- Visibility Level
 - Gibt an, welche User Zugriff auf das Projekt erhalten
 - Internal als zusätzliche Option bei eigenen GitLab Instanzen für angemeldete User und Guest Accounts
- Project Configuration
 - Mit einer README wird eine erste Datei und ein initialer Commit angelegt
 - Sinnvoll wenn man zuerst das Remote Repository anlegt um direkt clonen zu können
 - Bei bestehendem lokalen Repository sollte man ein leeres Remote Repository anlegen



GitLab Groups



Groups



5

1

6

Search or go to...

Your work

Projects

Groups

Issues 5

Merge requests >

To-Do List 6

Milestones

Snippets

Activity

Workspaces

Environments

Operations

Security >

Help

Your work / Groups

Groups

[Explore groups](#)

[New group](#)

Search by name

Last created ▾

• E	example-group 🔒	1
Example group description		
• T	Trainings 🔒	6

DEMO

- Gruppen können zur Verwaltung von einem oder mehreren Projekten gleichzeitig verwendet werden
 - Projekte können innerhalb von Gruppen angelegt werden
 - Dadurch befinden diese sich im **Namespace** der Gruppe, bspw. https://gitlab.example.de/my_group/my_project
 - Verwaltung von Rechten auf Gruppenebene
 - Mitglieder erhalten Zugriff auf alle Projekte innerhalb einer Gruppe
- Können zur Kommunikation verwendet werden
- Abrufen von gruppenspezifischen Statistiken und Aktivitäten
- Können in Untergruppen aufgeteilt werden
- Gruppeneinteilungen und Strukturen auf individuelle Bedürfnisse anpassbar



🏠 Your work / Groups / New group / Create group



Create group

Groups allow you to manage and collaborate across multiple projects. Members of a group have access to all of its projects.

Groups can also be nested by creating [subgroups](#).

Group name

Must start with letter, digit, emoji, or underscore. Can also contain periods, dashes, spaces, and parentheses.

⚠️ Your group name must not contain a period if you intend to use SCIM integration, as it can lead to errors.

Group URL

Visibility level

Who will be able to see this group? [View the documentation](#)

- ☒ Private
The group and its projects can only be viewed by members.
- ☐ Internal
The group and any internal projects can be viewed by any logged in user except external users.
- ☐ Public
The group and any public projects can be viewed without any authentication.

Now, personalize your GitLab experience

We'll use this to help surface the right features and information to you.

Role

Who will be using this group?

- ☐ My company or team
- ☐ Just me

What will you use this group for?

Invite Members (optional)

Invited users will be added with developer level permissions. [View the documentation](#) to see how to change this later.

Email 1

+ Invite another member

Create group Cancel



- Anlegen eines Projektes innerhalb einer Gruppe über Button „New Project“ auf Startseite/Gruppenseite

Your work / Projects / New project / Create blank project



Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

Project name

My awesome project

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

https://gitlab.com/ Example_Group

Project slug

my-awesome-project

Project deployment target

Select the deployment target

Search

Groups

Example_Group

Users

Example_User

Visibility Level

☒ Private

Project access must be granted to members of the group.

☐ Public

The project can be accessed without any authentication.

this project is part of a group, access is granted to members of the group.

Project Configuration

☒ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

☐ Enable Static Application Security Testing (SAST)

Analyze your source code for known security vulnerabilities. [Learn more.](#)

Create project

Cancel

DEMO

GitLab

Clonen von Projekten



Clonen von Projekten

The screenshot shows the GitLab web interface for a project named "Example Project". The left sidebar contains navigation options: Project, Pinned, Issues, Merge requests, Manage, Plan, Code, Build, Secure, Deploy, Operate, Monitor, Analyze, and Settings. The main content area displays the project details, including the initial commit, a table of files, and the README content. The right sidebar shows project information, including commit count, branch count, tags, project storage, and a list of integrations.

Example User / Example Project

Project

- Example Project
- Pinned
- Issues
- Merge requests
- Manage
- Plan
- Code
- Build
- Secure
- Deploy
- Operate
- Monitor
- Analyze
- Settings

Example Project

main example-project /

Initial commit
Example User authored 11 minutes ago

f60432af

Name	Last commit	Last update
README.md	Initial commit	11 minutes ago

README.md

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!](#)

Add your files

- ☐ Create or upload files
- ☐ Add files using the command line or push an existing Git repository with the following command:

Project information

- 1 Commit
- 1 Branch
- 0 Tags
- 3 KiB Project Storage

README

- [Add LICENSE](#)
- [Add CHANGELOG](#)
- [Add CONTRIBUTING](#)
- [Add Kubernetes cluster](#)
- [Set up CI/CD](#)
- [Add Wiki](#)
- [Configure Integrations](#)

Created on
May 27, 2024

DEMO

Clonen von Projekten

- In Projektübersicht **Code** auswählen

Example User / Example Project

E

Example Project

Star 0 Fork 0

main

example-project /

History

Find file

Edit

Code

Project information

Initial commit
Example User authored 1 hour ago

Name	Last commit
README.md	Initial commit

README.md

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 even better? [See how](#).

Add your files

Created on
May 07, 2024

Clone with SSH

git@gitlab.example.de:exampleuse

Clone with HTTPS

git@gitlab.example.de/exampleuse

Open in your IDE

Visual Studio Code (SSH)

Visual Studio Code (HTTPS)

IntelliJ IDEA (SSH)

IntelliJ IDEA (HTTPS)

Download source code

zip

tar.gz

tar.bz2

tar

Project Storage

USE

IGELOG

TRIBUTING

to DevOps

netes cluster

CD

Integrations

DEMO

- Clonen mittels SSH Key oder über HTTPS möglich
- Im gewünschten Zielverzeichnis den Befehl `git clone` in Kombination mit der kopierten URL ausführen
- Zum Clonen über SSH muss ein öffentlicher SSH Schlüssel hinterlegt werden



SSH Keys

- Können zur verschlüsselten Kommunikation zwischen lokalem Repository und GitLab Server verwendet werden
- Sinnvolle Alternative zur wiederholten Verwendung von Username und Passwort
- SSH Keys besteht aus einem Schlüsselpaar
 - Öffentlicher Schlüssel wird auf den GitLab Server hinterlegt
 - Privater Schlüssel verbleibt geschützt auf Endgerät und wird nicht geteilt
- Unterstützte SSH Key Formate:
 - ED25519 (bevorzugt)
 - ECDSA
 - RSA (mind. 2048-bit)
 - DSA (nicht mehr empfohlen)



- SSH Schlüsselpaar generieren (ED25519)
 - Folgenden Befehl in Terminal ausführen:
`ssh-keygen -t ed25519 -C "<comment>"`
 - Dateiname und Speicherort angeben
 - Passphrase für Schlüssel einrichten (optional)
- Öffentlichen Schlüssel auf GitLab hochladen
 - Inhalt der erstellten .pub Datei kopieren
 - Navigieren **Startseite** → **Avatar** → **Edit profile** → **SSH Keys** (Sidebar)
 - **Add new key** auswählen
 - Öffentlichen Schlüssel in die Textbox kopieren
 - Optional **Usage Type** und **Expiration Date** festlegen
 - **Add key** auswählen



User Settings / SSH Keys

Search settings

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the [current instance configuration](#).

Your SSH keys 3

Add an SSH key

Add an SSH key for secure access to GitLab. [Learn more](#).

Key

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGkcePbmIo3PYxYHsrLZNFvYa97ZavFWHmuSAfrt79EF gitlab

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

Title

gitlab

Key titles are publicly visible.

Usage type

Authentication & Signing

Expiration date

2025-05-06

Optional but recommended. If set, key becomes invalid on the specified date.

Add key

Cancel

Title	Key	Usage type	Created	Last used	Expires	Actions
-------	-----	------------	---------	-----------	---------	---------

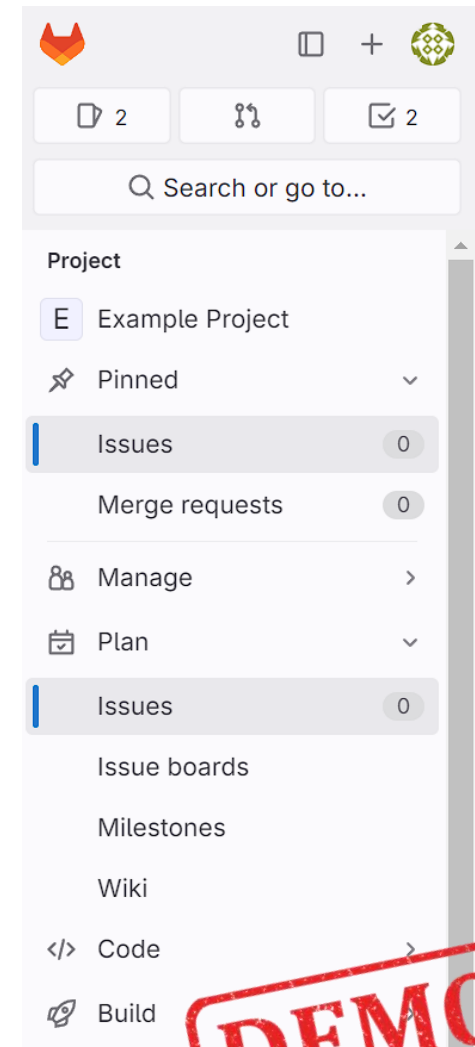
DEMO


GitLab

Planing Tools



- In GitLab können innerhalb eines Projektes Issues für verschiedene Aufgaben angelegt werden
- Issues werden in der Menübar links sowohl als Shortcut als auch unter **Plan → Issues** angezeigt
- Issues können mit Labels versehen werden
 - Eigene Labels können beliebig unter **Manage → Labels** hinzugefügt werden
 - Default Set kann ebenfalls unter **Manage → Labels** generiert werden
 - Im Board können einzelne Listen für verschiedene Labels angelegt werden
 - Labels können priorisiert werden





2

2

Search or go to...

Project

Example Project

Pinned

Issues

Merge requests

Manage

Activity

Members

Labels

Plan

Code

Build

Secure

Deploy

Operate

Monitor

Analyze

Settings

Help

Example User / Example Project / Labels


All

Subscribed

Labels can be applied to issues and merge requests. Star a label to make it a priority label.

Prioritized labels

Drag to reorder prioritized labels and change their relative priority.




No prioritized labels yet!
Star labels to start sorting by priority.

Other labels

<div>bug</div> <div>Example User / Example Project</div>	<div>Issues</div> <div>Merge requests</div> <div>☆</div> <div>Subscribe</div> <div>⋮</div>
<div>confirmed</div> <div>Example User / Example Project</div>	<div>Issues</div> <div>Merge requests</div> <div>☆</div> <div>Subscribe</div> <div>⋮</div>
<div>critical</div> <div>Example User / Example Project</div>	<div>Issues</div> <div>Merge requests</div> <div>☆</div> <div>Subscribe</div> <div>⋮</div>
<div>discussion</div> <div>Example User / Example Project</div>	<div>Issues</div> <div>Merge requests</div> <div>☆</div> <div>Subscribe</div> <div>⋮</div>
<div>documentation</div> <div>Example User / Example Project</div>	<div>Issues</div> <div>Merge requests</div> <div>☆</div> <div>Subscribe</div> <div>⋮</div>
<div>enhancement</div> <div>Example User / Example Project</div>	<div>Issues</div> <div>Merge requests</div> <div>☆</div> <div>Subscribe</div> <div>⋮</div>
<div>suggestion</div> <div>Example User / Example Project</div>	<div>Issues</div> <div>Merge requests</div> <div>☆</div> <div>Subscribe</div> <div>⋮</div>
<div>support</div> <div>Example User / Example Project</div>	<div>Issues</div> <div>Merge requests</div> <div>☆</div> <div>Subscribe</div> <div>⋮</div>

Neues Issue anlegen

 Next

2

2

Search or go to...

Project

E Example Project

Pinned

Issues 0

Merge requests 0

Manage

Plan

Issues 0

Issue boards

Milestones

Wiki

</> Code

Build

Secure

Deploy

Operate

Monitor

Analyze

Settings
































Example User / Example Project / Issues / New

New Issue

Title (required)

Type ?

Description

Preview B I  |  </>                             

Write a description or drag your files here...

Switch to rich text editing

Add [description templates](#) to help your contributors to communicate effectively!

☐ This issue is confidential and should only be visible to team members with at least Reporter access.

Assignee

[Assign to me](#)

Due date

Milestone

Labels

Create issue

Cancel

DEMO

5

5

Search or go to...

Project

Example Project

Pinned

Issues 3

Merge requests 0

Manage

Plan

Issues 3

Issue boards

Milestones

Wiki

Code

Build

Secure

Deploy

Operate

Monitor

Analyze

Settings

Example User / Example Project / Issues / #3

Add unit tests

Open Issue created 7 minutes ago by Example User

0 0

Create merge request

Drag your designs here or [click to upload](#).

Child items 0

Show labels ☒ Add

No child items are currently assigned. Use child items to break down this issue into smaller parts.

Linked items 0

Add

Link issues together to show that they're related. [Learn more](#).

Activity

Sort or filter

- Example User added testing label 7 minutes ago
- Example User assigned to [@Example_User](#) 7 minutes ago

Preview **B** *I* U `</>`

Write a comment or drag your files here...

Switch to rich text editing

☐ Make this an internal note

Comment Close issue

Mark as done

Assignee Edit

Example User

Labels Edit

testing

Milestone Edit

None

Weight

This feature is locked. [Learn more](#)

Due date Edit

None

Time tracking +

No estimate or time spent

Confidentiality Edit

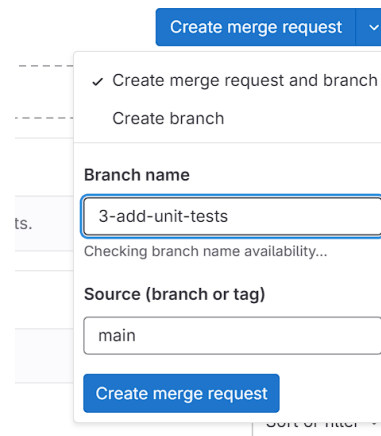
Not confidential

1 Participant

Move issue

DEMO


- Issues können an Personen „assigned“ werden, also zugewiesen werden
- Milestones, Fälligkeitsdatum und Time Tracking können ebenfalls hinzugefügt werden
- GitLab bietet die Möglichkeit, direkt aus Issues Merge-Requests oder Branches anzulegen



- Issues werden nach Erledigung geschlossen und damit in die „closed“ Sektion verschoben
- Templates für Issues können als Datei unter `.gitlab/issue_templates` mittels Markdown in eine `.md` Datei gespeichert werden



Issue-Liste



5

5

Project

E

Example Project

📌

Pinned

Issues

3

Merge requests

0

👤

Manage

📅

Plan

Issues

3

Issue boards

Milestones

Wiki

</>

Code

Example User / Example Project / Issues

Open 3

Closed 0

All 3

Bulk edit

New issue

⋮

🕒

Search or filter results...

Q

Created date

↓

📄

Add unit tests

#3 · created just now by Example User

testing

🔗

📄

Fix redirect after login

#2 · created 2 minutes ago by Example User

bug critical

🔗

📄

Redesign landing page

#1 · created 2 minutes ago by Example User

Release-v1.0.0

enhancement

🔗

[Email a new issue to this project](#)

DEMO

- GitLab besitzt mit dem Issue Board ein Software Management Tool
- Kann zur Planung, Organisation und Visualisierung genutzt werden
 - Nutzung als Kanban oder Scrum Board
 - Effektive Aufgabenerfassung und Aufgabenverteilung
- Für Aufgaben werden einzelne Issues angelegt
 - Grundlegend in offene und geschlossen Labels aufteilt
- Board kann nach Labels strukturiert werden



Issue-Board

Example User / Example Project / Issue Boards

Development Search

Search or go to...

Project

- Example Project
- Pinned
- Issues (3)
- Merge requests (0)
- Manage
- Plan
- Issues (3)
- Issue boards**
- Milestones
- Wiki
- Code
- Build
- Secure
- Deploy
- Operate
- Monitor
- Analyze
- Settings

Open (3)

- Redesign landing page (enhancement) #1 Release-v1.0.0
- Fix redirect after login (bug, critical) #2
- Add unit tests (testing) #3

Closed (0)

+ New list

Help

DEMO

Neue Liste anlegen


The screenshot displays the GitLab Issue Board interface. On the left is a sidebar with navigation options: Project (Example Project), Pinned, Issues (3), Merge requests (0), Manage, Plan, Issues (3), Issue boards (selected), Milestones, Wiki, Code, Build, Secure, Deploy, Operate, Monitor, Analyze, and Settings. The main area shows three columns: 'Open' (3 items), 'critical' (1 item), and 'Closed' (0 items). The 'Open' column lists: 'Redesign landing page' (enhancement, #1, Release-v1.0.0), 'Fix redirect after login' (bug, critical, #2), and 'Add unit tests' (testing, #3). The 'critical' column lists: 'Fix redirect after login' (bug, #2). The 'Closed' column is empty. A red box highlights the '+ New list' button in the top right corner of the board.

DEMO

- Milestones können als Ziel angelegt werden
- Issues und Merge-Requests können an Milestones gekoppelt und dadurch gruppiert werden
- Besitzen optionalen Start- und Endpunkt



Neuen Milestone anlegen



2

2

2

Q Search or go to...

Project

E Example Project

Pinned 0

Issues 0

Merge requests 0

Manage >

Plan >

Issues 0

Issue boards

Milestones

Wiki

</> Code >

Build >

Secure >

Deploy >

Operate >

Monitor >

Analyze >

Settings >

Example User / Example Project / Milestones / New

New Milestone

Title

Start Date

Select start date

[Clear start date](#)

Due Date

Select due date

[Clear due date](#)

Description

Preview | B I S | I= </> | | | | | | |

Write milestone description...

Switch to rich text editing

Create milestone

Cancel

anderScore GmbH • Frankenwerft 35 • 50667 Köln
Daniel Krämer & Malte Fischer

Tag-1_5-Gitlab_Light.pptx
12.06.2024

39

- Ermöglicht Dokumentation innerhalb des Projekts
- Wird in einem separaten Git-Repository gespeichert
 - Keine direkte Versionierung zusammen mit dem Code
- Ist in einzelne Seiten aufgeteilt, welche in der Sidebar rechts angezeigt werden
- Ermöglicht Strukturierung in Unterseiten

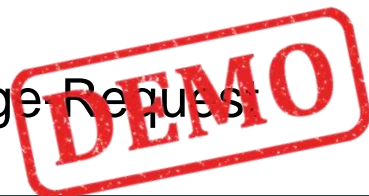


GitLab


Branches, Commits & Merging



- Beim Erstellen eines Projektes legt GitLab eine Default Branch an
 - Einstellungen bezüglich Default Branch können auf Projekt, (Sub-)Gruppen oder Instanz Level vorgenommen werden
- Default Branch wird als *protected* bezeichnet und kann nicht gelöscht werden
 - Mit Standardeinstellungen ist ebenfalls das force pushen auf einen Default Branch nicht möglich
 - Einstellungen hierzu können unter **Settings** → **Repository** feingranular vorgenommen werden
- Neue Branches können angelegt werden über
 - Projekt-Startseite
 - Unterpunkt Code → Branches
 - Über Issues
 - Lokal und per Push in GitLab übertragen
- Ist ein Branch abgeschlossen, wird üblicherweise ein Merge Request (MR) gestellt, um Änderungen ins Projekt zu integrieren



Branches



5

1

5

Q Search or go to...

Project

E Example Project

Pinned

Issues 3

Merge requests 1

Manage

Plan

</> Code

Merge requests 1

Repository

Branches

Commits

Tags

Repository graph

Compare revisions

Snippets

Build

Example Users / Example Project / Branches

OverviewActiveStaleAll

Filter by branch name

View branch rules








New branch

See all branch-related settings together with branch rules

You can now find an overview of settings for protected branches, merge request approvals, status checks, and security approvals conveniently in one spot.

View branch rulesDismiss

Active branches

3-add-unit-tests 66b511a1 · Add conflict file tests · 5 minutes ago	1 2	 1		
main 212e031b · Add conflict file main · 5 minutes ago				
feature/my_feature db15c779 · Add new file · 28 minutes ago	1 0	 New		

DEMO

- Commits bietet die Möglichkeit, alle Commits einen ausgewählten Branches und deren Änderungen anzuzeigen

The screenshot shows the GitLab interface for the 'Example Project' under the user 'Christopher Keutner'. The left sidebar contains navigation options: Project (Example Project), Pinned, Issues (3), Merge requests (1), Manage, Plan, Code, Merge requests (1), Repository, Branches, Commits (selected), Tags, Repository graph, Compare revisions, Snippets, and Build. The main content area displays a list of commits for the '3-add-unit-tests' branch. The commits are:

- 66b511a1**: Add conflict file tests (Christopher Keutner authored 6 minutes ago)
- 28094c25**: Add new file (Christopher Keutner authored 28 minutes ago)
- db15c779**: Add new file (Christopher Keutner authored 30 minutes ago)
- f60432af**: Initial commit (Christopher Keutner authored 5 hours ago)


Each commit entry includes a commit icon, the commit message, the author's name, the time since authored, and a box with the commit hash and icons for viewing the commit and its files. At the top of the commit list, there is a date separator for 'May 27, 2024'. Above the commit list, there are filters for 'Author' and 'View open merge request', and a search bar for 'Search by message'.

DEMO


- Merge-Requests sind eine Anfrage, um neue Änderungen in das Projekt einzubringen
- Git-Merge von Branches als Grundkonzept
- Erweitert einfachen Merge um folgende Features
 - Beschreibung des Merge-Request
 - Anzeige der Codeänderungen für Reviews
 - Auflistung der einzelnen Commits des Requests
 - Verknüpfungen mit Issues oder Milestones
 - Kommentarsektion zur Diskursion
- Bietet Option den Source Branch nach dem Merge zu löschen
- Ermöglicht Squashing von Commits
 - Zielbranch enthält dann nicht alle einzelnen Commits des Source Branches und bleibt ggf. übersichtlicher



Merge-Requests



5




4

Search or go to...

Project


E


Example Project


 Pinned


Issues 3


Merge requests 0


 Manage


 Plan


 Code


 Build


 Secure

 Deploy

 Operate

 Monitor

 Analyze

 Settings

Example User / Example Project / Merge requests / New

New merge request

Source branch

Example_User/example-project

Select source branch

Select a branch to compare

Compare branches and continue

Target branch

Example_User/example-project

main



Add new file

Example User authored May 27, 2024

db15c779



DEMO

anderScore
trust in competence

New merge request

Drafts cannot be merged until marked ready.


Cancel




- **Draft** – Merge Request ist für andere Personen einsehbar, kann aber noch nicht gemerged werden
- **Assignee** – Zugewiesene Person, die für den Merge-Request bzw. für das Einpflegen der Änderungen zuständig ist
- **Reviewer** – Person(en), die die Änderungen reviewen und durch einen *Approve* bestätigen können. Approves können als verpflichtend sowie als Optional eingestellt werden
- **Milestone** – Zugehöriger Milestone
- **Labels** – Dienen zur Kategorisierung und Einordnung des MR
- **Merge Options**
 - Delete Source Branch → Source Branch wird nach Durchführung des MR gelöscht
 - Squash Commits → Commits des Source Branches werden gesquashed zum Target-Branch hinzugefügt



- GitLab versucht durch Templates den MR bereits mit Informationen anzureichern
 - So führt die 3 beim Branch 3-add-unit-tests im vorherigen Beispiel dazu, dass in die Beschreibung ein Closes #3 eingefügt wird und die Labels des Issues übernommen werden
 - Generelle Templates können als Datei unter .gitlab/merge_request_templates mittels Markdown in eine .md Datei gespeichert werden
- Merge-Konflikte werden in der Übersicht angezeigt

 Merge blocked: 1 check failed

 Merge conflicts must be resolved. [Resolve locally](#) [Resolve conflicts](#)

Merge details

- The source branch is [1 commit behind](#) the target branch.
- 1 commit and 1 merge commit will be added to main (squashes 2 commits).
- Source branch will be deleted.
- Closes issue [#3](#)



- Auch bei Merge-Requests können, analog zu lokalen Konflikten, ebenfalls Merge-Konflikte auftreten
- Kleinere Konflikte können über GitLab aufgelöst werden, größere müssen jedoch lokal behoben werden

The screenshot displays the GitLab Merge Request interface for a conflict resolution. The left sidebar shows the project navigation menu with 'Merge requests' selected. The main content area is titled 'Resolve "3-add-unit-tests"' and shows a conflict in the file 'conflict_file.txt'. The conflict is between the 'HEAD//our changes' (unit-tests) and the 'main' branch (origin//their changes). The interface provides options to 'Use ours' or 'Use theirs'. Below the conflict, there is a section for 'Resolve conflicts on source branch' with instructions on how to resolve the conflict. A 'Commit message' field is also present, containing the text 'Merge branch 'main' into '3-add-unit-tests'' and '# Conflicts: # conflict_file.txt'. The bottom right corner features a large red 'DEMO' stamp.

GitLab Übung

Aufgabe 1: Neues Projekt erstellen

1. (Optional) Beim Arbeiten mit GitLab ist es sinnvoll, mittels SSH Keys auf Remote Repositories zuzugreifen, statt immer wieder Username und Passwort zu verwenden.
Falls Sie noch keinen SSH Key in GitLab hinterlegt haben, so können Sie entweder die Anleitung aus GitLab nutzen oder dem Tutorial unter <https://docs.gitlab.com/ee/user/ssh.html> folgen.
2. Legen Sie ein neues privates Projekt **Playground** in Ihrem eigenen Bereich an, sodass nur Sie selbst Zugriff darauf haben. Achten Sie darauf, dass Sie das Repository mit einer Readme initialisieren, um dieses direkt klonen zu können.
3. Klonen Sie das Repository in ein lokales Verzeichnis auf Ihrem Rechner.

Hinweis:

Man kann grundsätzlich viele Änderungen am Projekt sowohl *lokal* als auch über die *GitLab Weboberfläche* vornehmen. Für gewöhnlich bearbeitet man eher selten Dateien in der GitLab Weboberfläche, sondern nimmt Änderungen lokal in einer IDE vor und pusht diese danach ins Remote Repository.

Um das Arbeiten mit mehreren Entwicklern im Projekt etwas nachzustellen, werden wir im Folgenden auch Änderungen über die Weboberfläche vornehmen, um damit Änderungen von anderen Entwicklern zu simulieren.

Dadurch lassen sich verschiedene Szenarien erzeugen, deren Behandlung wir an unserem lokalen Repository üben werden.

Aufgabe 2: Erste Schritte im Repository

1. Navigieren Sie in einem Terminal in das neue geklonte Projekt auf Ihrem Rechner.
2. Mit dem Befehl `git status` können Sie Ihren aktuellen Stand abfragen.
Sie sollten sich auf dem Branch **main** befinden, der sich auf demselben Stand wie **origin/main** befindet.
3. `git status` vergleicht nur den lokalen Remote Tracking Bereich mit Ihrem aktuellen Branch. Es gibt keinen Aufschluss darüber, ob remote neue Änderungen verfügbar sind.
Nutzen Sie `git fetch`, um ihren lokalen Remote Tracking Bereich mit dem GitLab Repository zu synchronisieren.
Es sollten keine neuen Änderungen verfügbar sein.

Aufgabe 2: Erste Datei hinzufügen

1. Legen Sie eine Datei **random_numbers.sh** an und füllen Sie diese mit dem folgenden Inhalt

```
#!/bin/bash

random_number=$(( RANDOM % 10 + 1 ))

if [ $random_number -lt 11 ]; then
    echo "Number is less than 11"
fi
```

2. Comitten Sie Ihre Änderungen auf dem **main** Branch.
3. Pushen Sie Ihre Änderungen ins Remote Repository. Vergewissern Sie sich über die Web-GUI, dass Ihr Push erfolgreich war.

Aufgabe 3: Issue erstellen

1. Über das Issue Board in GitLab können Aufgaben definiert und an bestimmte Personen zugewiesen werden.
Legen Sie im Issue Board ein neues Issue **Add output if number greater than 5** an.
Diese Aufgabe sollen Sie nun übernehmen, tragen Sie sich daher als Assignee ein.
2. GitLab bietet die Möglichkeit, aus Issues direkt Merge-Requests und Branches zu erstellen.
Klicken Sie auf das Dropdown beim Button „Create merge request and branch“ und wählen Sie nur „Create branch“ aus.
Erstellen Sie nun eine Branch, um die Änderungen für das Feature vorzunehmen.

Aufgabe 4: Lokale Änderungen

1. Führen Sie in Ihrem lokalen Repository den Befehl `git status` sowie `git branch -a` aus.
Ihnen sollte der neu angelegte Branch aus GitLab nicht angezeigt werden, da Sie diesen erst vom Remote abrufen müssen.
2. Fetchen Sie alle Änderungen vom Remote.
`git branch -a` sollte nun den neuen Branch anzeigen.
3. Wechseln Sie in den Feature Branch.
4. Öffnen Sie die `random_numbers.sh` Datei und nehmen Sie die geforderten Änderungen vor, indem Sie die Datei um ein weiteres *if-Statement* ergänzen, indem geprüft wird, ob die Zahl größer als 5 ist.
5. Comitten Sie Ihre Änderungen und pushen Sie diese ins Repository.

Aufgabe 5: Merge Request

Nun wollen wir unsere Änderungen des Features in den main Branch übernehmen. Lokal würde man den Feature Branch in den main Branch mergen. Der Ablauf bei GitLab ist funktional sehr ähnlich, erweitert jedoch das reine Merging um einige Aspekte.

1. Navigieren Sie über die Sidebar in der GitLab Web-GUI auf den Punkt Merge Requests.
2. Erstellen Sie einen neuen Merge-Request, um Ihren Feature Branch in den **main** Branch zu mergen.
Weisen Sie sich als Assignee und Reviewer zu (normalerweise würde man logischerweise jemand anderen als Reviewer eintragen).
3. Im Merge-Request haben Sie oben die Option, sich Commits und Changes anzuschauen. Reviewen Sie Ihre Änderungen und approben Sie den Merge-Request.

Aufgabe 5: Merge Request

4. Mergen Sie nun den Merge Request mit der Option „Delete source branch“ und verifizieren Sie, dass die Änderungen auf dem **main** Branch vorhanden sind.
5. Löschen Sie Ihren lokalen Branch, da dieser abgeschlossen ist und keine Verwendung mehr besitzt.

Aufgabe 6: Rebase von lokalen Branches

Nun wollen wir ein Rebase nutzen, um Remote Änderungen in unseren lokalen Branch zu integrieren.

1. Erstellen Sie ein neues Issue **Add output if number is maximum**. Dabei soll eine Ausgabe erfolgen, wenn die Random Nummer die größtmögliche Zahl ist.
2. Man muss nicht zwingend den Branch aus einem Issue heraus erstellen.
Legen Sie dieses Mal lokal einen Branch **2-add-output-if-max** an und führen Sie die notwendigen Änderungen durch.
3. Committen Sie Ihre Änderungen und pushen Sie diese zum Remote Repository.

Aufgabe 6: Rebase von lokalen Branches

In der Zwischenzeit hat ein Kollege die Aufgabe bekommen, den Zahlenbereich von 10 auf 1000 zu erhöhen.

Um die parallelen Änderungen von Ihm zu simulieren, führen wir diese über die Web-GUI durch. Ein zugehöriges Issue bzw. einen Branch überspringen wir zur Simplifizierung.

Hinweis: Ein gemeinsames Arbeiten über eine Datei sollte im Optimalfall vermieden werden, um Konflikte zu vermeiden.

4. Ändern Sie über GitLab den Random Number Bereich auf 1000 und committen Sie die Datei direkt aus dem Editor auf den **main** Branch.
5. Um im lokalen Repository die Änderungen zu übernehmen, müssen Sie den **main** Branch updaten. Wechseln Sie dazu in diesen und verwenden Sie `git pull`, um diese in Ihren lokalen **main** Branch zu übernehmen.

Aufgabe 6: Rebase von lokalen Branches

7. Um die Änderungen nun auch in den Feature Branch zu übernehmen, müssen Sie in den Feature Branch wechseln und dort auf den **main** Branch rebasen.
8. Rebasen Sie Ihren Feature Branch auf den **main** Branch. Dabei sollten keine Konflikte auftreten, da sich die betreffenden Zeilen bei Ihnen nicht geändert haben.
9. Passen Sie Ihre Ausgabe bezüglich der Random Number an und committen Sie Ihre Änderungen.
10. Versuchen Sie Ihren Branch zum Remote zu pushen. GitLab wird hierbei den Push wegen nicht zueinander passenden Commit-Historien ablehnen. Pushen Sie daher Ihren Branch mit der Option `--force`.

Aufgabe 6: Rebase von lokalen Branches

Wichtig: --force bewirkt, dass der Remote Branch durch die lokale Version überschrieben wird.

Falls andere Entwickler auf diesem Branch arbeiten würden, so könnten Sie weder push noch pull auf diesem ausführen, da die Commit-Historien nicht vereinbar sind. Ihre Branches wären damit kaputt und müssten komplett neu vom Remote abgerufen werden.

Daher niemals auf Public Branches rebasen!

Aufgabe 7: Merge Konflikte

In dieser Aufgabe wollen wir uns das Auflösen von Merge-Konflikten anschauen.

Ein Kollege von Ihnen hat ebenfalls das Feature bezüglich Ausgabe bei maximaler Zahl umgesetzt und dieses schon in den **main** Branch integriert.

1. Simulieren Sie die Änderungen des Kollegen, indem Sie auch hier wieder über die Web-GUI direkt auf dem **main** Branch arbeiten. Kopieren Sie Ihre lokalen Änderungen bezüglich der Ausgabe und fügen sie diese über die Web-GUI in die **random_numbers.sh** Datei ein. Verändern Sie dabei die Konsolenausgabe, sodass sich die Änderungen unterscheiden.
2. Committen Sie Ihre Änderungen direkt auf **main**.

Aufgabe 7: Merge-Konflikte

3. Stellen Sie für Ihren Feature Branch einen Merge Request. Nach der Erstellung sollte GitLab anzeigen, dass es Konflikte beim Mergen gibt.
4. Kleinere Konflikte lassen sich über die Web-GUI auflösen, größere Konflikte lassen sich nur lokal auflösen.
Zur Übung lösen wir den Konflikt lokal auf.
5. Updaten Sie den **main** Branch auf den neusten Stand aus dem Remote Repository.
6. Lösen Sie die Konflikte nun, indem Sie entweder den **main** Branch in Ihren Feature Branch mergen oder indem Sie den Feature Branch auf den aktuellen Stand des **main** rebasen. Da Sie alleine auf ihrem Branch arbeiten, bietet sich auch hier ein Rebase an.
7. Lösen Sie den Konflikt, indem Sie sich für Ihre Ausgabe entscheiden.

Aufgabe 7: Merge-Konflikte

8. Pushen Sie nach Auflösen des Konflikts Ihre Änderungen ins GitLab.
9. Der Merge-Request sollte nun ohne Konflikte umsetzbar sein. Mergen Sie Ihr Feature in den **main** Branch, um die Änderungen abzuschließen.