



Gitlab-CE-Server unter Linux installieren und betreiben

Vorwort zur Anleitung:

Diese Anleitung hat einen hohen Schwierigkeitsgrad. Installation von Linux und Kenntnisse in der Befehlszeile sind vorausgesetzt.

Was ist ein Gitlab-CE-Server?



Ein **GitLab CE Server** ist ein Server, auf dem die **Community Edition** (CE) der GitLab-Software installiert und betrieben wird. GitLab ist eine DevOps-Plattform, die Entwicklern hilft, Code zu verwalten und zu entwickeln.

GitLab CE (Community Edition)

- **Open Source:** Die Community Edition ist die kostenlose, Open-Source-Version von GitLab. Sie bietet Funktionen für die Versionskontrolle und DevOps-Prozesse..
- **Funktionsumfang:** GitLab CE umfasst Funktionen wie:
 - Git-Repository-Management
 - Code-Reviews
 - CI/CD-Pipelines (mit begrenztem Funktionsumfang im Vergleich zur kostenpflichtigen Version)
 - Issue-Tracking
 - Merge-Requests
 - Wiki und Snippets

Einsatzmöglichkeiten eines GitLab CE Servers

- **Entwicklerteams** können ihre Projekte und Code-Repositorys auf diesem Server verwalten.
- **CI/CD**-Prozesse für automatisierte Tests und Deployments werden auf dem GitLab CE Server eingerichtet.
- **Private Git-Instanzen:** Unternehmen oder Einzelpersonen, die GitLab für ihre eigene Infrastruktur verwenden möchten, ohne eine kommerzielle Lizenz (Enterprise Edition) zu benötigen, können GitLab CE auf einem eigenen Server betreiben.

Zusammengefasst ist ein GitLab CE Server einfach eine selbstgehostete Installation von GitLab in der Community Edition. Er kann auf einer Vielzahl von Betriebssystemen installiert werden und ermöglicht es Teams, GitLab mit den grundlegenden Funktionen zu nutzen.

Vorbereitungen / Anforderungen:

<p>Anforderungen</p> <ul style="list-style-type: none"> • Schwierigkeitsgrad-Installation: Hoch • Schwierigkeitsgrad-Anwendung: Hoch • Erforderliche Kenntnisse: Linux-Befehlszeile <p>Systemvoraussetzungen:</p> <ul style="list-style-type: none"> • Basis: Debian • Betriebssystem: Ubuntu Server 22.04 (Headless) • Festplattenspeicher: Mindestens 17 GB • CPU: Mindestens 2 CPU • RAM: Mindestens 5 GB • Internetverbindung: Verfügbar 	<p>Die Systemanforderungen wurden in einer virtualisierten Umgebung getestet.</p> <p>Tipp:</p> <p>Wenn du mit Virtualisierung, z.B. ProxMox, vertraut bist, kannst du deine eigenen Server effizient und stromsparend betreiben!</p> <p>Getestet auf:</p> <p>Ubuntu-Server 22.04 Ubuntu-Server 24.10</p>
--	--

<p>Installationsablauf:</p> <ol style="list-style-type: none"> 1. Durchführung von System-Upgrade 2. Statische IP-Adresse festlegen und anwenden 3. (Optional) Hostnamen festlegen 4. Installation und Einstellungen 5. Systembedienung 6. Resümee 7. GitLab-User anlegen 8. GitLab-Projekte anlegen 9. GitLab-Projekt klonen (Push&Pull) 10. Resümee 11. Schlussbemerkung 	<p>Optionale Schritte können weggelassen werden, da sie lediglich unterstützende Funktionen bieten, wie zum Beispiel die Zusammenfassung von Servern in einem Rechenzentrum.</p> <p>Für den Betrieb in einem Rechenzentrum sollten die optionalen Abläufe jedoch aus Gründen des beruflichen Stolzes in Betracht gezogen werden.</p>
--	--

1. Durchführung von System-Upgrade

Vor jeder Installation ist ein System-Update erforderlich:

\$> ssh DEINUSER@192.168.1.X	Stelle eine SSH-Verbindung zum Server her, um Remote-Operationen durchzuführen.
\$> sudo apt update	Aktualisiere die Paketquellen, um sicherzustellen, dass du die neuesten Versionen der Pakete erhältst
\$> sudo apt upgrade -y	Starte das System-Upgrade und verwende dabei die Option --yes-to-all, um alle Bestätigungsabfragen automatisch zu beantworten.
\$> sudo apt autoclean	Entferne Pakete, die nicht mehr benötigt werden, um Speicherplatz zu sparen und das System zu optimieren.
\$> sudo apt autoremove	Bereinige das System von überflüssigen Abhängigkeiten, die nach Paket-Deinstallationen übrig geblieben sind.

2. Statische IP-Adresse festlegen und anwenden

Ein Gitlab-Server benötigt zwingend eine statische IP-Adresse, weil Client-System ausfallen, wenn sich diese Adresse ändert.

Es besteht die Möglichkeit, sich selbst vom Server auszuschließen, wenn die IP-Adresse nicht korrekt konfiguriert ist.

\$> ip addr	Verwende den Befehl <code>ip addr</code> , um alle Netzwerk-Adapter und deren Namen anzuzeigen.
\$> sudo su	Stelle sicher, dass du über Root-Rechte verfügst, um Änderungen vorzunehmen.
\$> cd /etc/netplan	Gehe in das Verzeichnis, in dem sich die Netzwerk-Konfigurationsdateien befinden.
\$> ls	Zeige mit dem Befehl <code>ls</code> alle relevanten YAML-Dateien im Verzeichnis an.
\$> for i in \$(ls); do mv \$i \$i.bak; done	Sichere alle bestehenden Konfigurationsdateien, bevor du Änderungen vornimmst.
\$> touch /etc/netplan/01_static_ip.yaml	Erstelle eine neue Konfigurationsdatei
\$> nano /etc/netplan/01_static_ip.yaml	Öffne die Datei mit einem Texteditor
# Beispiel-Konfiguration von Linux-Schweiz.ch network: version: 2 renderer: networkd ethernets: ens18: #Edit this line according to your network interface name. dhcp4: no addresses: - 192.168.1.150/24 gateway4: 192.168.1.1 nameservers: addresses: - 8.8.8.8 - 8.8.4.4	(Copy-Paste) Ersetze ens18 durch den Namen deines Netzwerk-Adapters (siehe Ausgabe von <code>ip addr</code>). Ändere die IP-Adresse und die Netzmaske nach Bedarf. Passen die DNS-Adressen an, falls erforderlich. Überprüfe und passe ggf. die Routen (Gateway) an. Speichern und Verlassen <ul style="list-style-type: none">• Speichern der Datei: <code>Ctrl + O</code>• Editor verlassen: <code>Ctrl + X</code>
\$> netplan generate && netplan apply	Wende die neuen Netzwerkeinstellungen an
--- Verbindungsunterbruch - IP wird neu gesetzt	

	Der Server wechselt nun zu einer neuen IP-Adresse, was dazu führt, dass deine aktuelle SSH-Sitzung unterbrochen wird.
---	---

3. (Optional) Hostnamen festlegen

Durch das Ändern des Hostnamens der Server-Station vergibst du einen eindeutigen Namen für das System.

In unserem Fall könnte der Hostname beispielsweise „S5-Gitlab-Server“ lauten.

Die Hosts-Datei wird verwendet, um auf Server-Ebene IP-Adressen Namen zuzuordnen, die intern genutzt werden. Wenn du den Hostnamen änderst, muss auch der entsprechende Eintrag in der Hosts-Datei aktualisiert werden. Bei produktiven Systemen sollte diese Kleinigkeit unbedingt beachtet werden, da sie ein Zeichen von Berufsethos und Professionalität ist. Für Tests und Experimente ist dies jedoch nicht zwingend erforderlich.

Der neue Hostname muss zwingen der gleiche sein wie auch im Hosts-Files.

<pre>\$> ssh DEINUSER@192.168.1.X</pre>	Stelle eine SSH-Verbindung zum Server über die neue IP-Adresse her.
<pre>\$> sudo hostnamectl set-hostname DEINHOSTNAME</pre>	Setze einen neuen Hostnamen.
<pre>\$> sudo nano /etc/hosts</pre>	Bearbeite die Datei /etc/hosts oder die entsprechende Datei, in der der Hostname definiert ist.
<pre>127.0.0.1 localhost 127.0.1.1 DEINHOSTNAME # The following lines are desirable for IPv6 capable hosts ::1 ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters</pre>	Ändere ausschließlich den markierten Bereich, der den Hostnamen enthält. Achte darauf, dass du keine anderen Einträge unbeabsichtigt veränderst. Speichern und Verlassen <ul style="list-style-type: none">• Speichern der Datei: Ctrl + O• Editor verlassen: Ctrl + X
<pre>\$> sudo reboot</pre>	Starte den Server neu, damit die Änderungen wirksam werden.

4. Installation und Einstellungen

GitLab-CE ist keine Anwendung aus dem Repository von Ubuntu-Linux. Falls die Installation fehlschlägt muss die Installations-URL angepasst werden. Universe und Multiverse-Quellen müssen im Sources-File aktiviert sein. Mehr unter:

<https://packages.gitlab.com/gitlab/gitlab-ce/install>

<pre>\$> ssh DEINUSER@192.168.1.X</pre>	Stelle eine SSH-Verbindung zum Server über die neue IP-Adresse her.
<pre>\$> sudo apt install curl -y \$> sudo apt install openssh-server -y \$> sudo apt install ca-certificates -y \$> sudo apt install tzdata -y \$> sudo apt install perl -y \$> sudo apt install postfix -y</pre>	Unterstützende Pakete installieren
<pre>\$> cd \$> mkdir downloads && cd downloads</pre>	Optional – wechselt in ein anderes Verzeichnis für den Download
<pre>\$> curl -O https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh</pre>	Installationshilfe für den GitLab-Server herunterladen

\$> chmod +x script.deb.sh	Bash-Script ausführbar machen
\$> sudo bash script.deb.sh	Bash-Script ausführen
\$> sudo apt install gitlab-ce	Installieren des Gitlab-Servers

Die GitLab-Installation ist eine umfangreiche Software mit großem Dateivolumen, die eher mit einer eigenständigen Betriebssystem vergleichbar ist. Es wird in dieser Anleitung nur das nötigste zur Lauffähigkeit erläutert.

\$> sudo nano /etc/gitlab/gitlab.rb	Einstellungen bearbeiten, externe URL ändern – verwende deine IP-Adresse
Erstellen einer lokalen Domain oder einer Weltweiten Domain auf deine IP-Adresse. In der Konfiguration kann auch mit deiner IP verwendet werden, jedoch muss sie zwingend im Format http://DEINE-IP sein.	
... external_url 'http://192.168.1.151' ...	
\$> sudo gitlab-ctl reconfigure	Neue Einstellung laden
\$> sudo cat /etc/gitlab/initial_root_password	Initial-Passwort anzeigen
\$> sudo gitlab-rake "gitlab:password:reset[root]"	Setzen eines neuen Admin-Passworts
Das Weblogin wird den Benutzernamen root und dein gewähltes Passwort verlangen. Wähle ein starkes Passwort	
\$> sudo reboot	Server-Neustart (Optional)

Die Basis-Installation ist nun installiert und kann über das Weblogin, welches deine URL oder IP-Adresse sein wird, erreicht werden. Eine Menge an Einstellungen werden dich im Web-Gui erwarten.

Hinweise zur Installation

GitLab ist eine leistungsintensive Anwendung, die sich von klassischen Linux-Anwendungen unterscheidet. Viele gängige Verwaltungsbefehle werden hier durch ein eigenes Programm, **gitlab-ctl**, gesteuert.

Daher sollte man vorsichtig sein und nicht willkürlich am Datenstamm herumbasteln. Die Daten und Projekte, die in diesem System gespeichert sind, können nicht einfach kopiert und wiederhergestellt werden. Aus Gründen der Einfachheit empfiehlt es sich daher, immer den gesamten Server zu sichern.

Erfahrungen zeigen, dass es sinnvoller ist, den gesamten Server neu zu starten, anstatt nur die Anwendung lokal zu stoppen und neu zu starten.

5. Systembedienung

<code>\$> sudo gitlab-ctl restart</code>	Startet Gitlab-CE, um die Zeit-Synchronisation zu aktivieren.
<code>\$> sudo gitlab-ctl stop</code>	Stoppt Gitlab-CE, wenn du ihn vorübergehend deaktivieren möchtest.
<code>\$> sudo gitlab-ctl status</code>	Zeigt den Status aller GitLab-Dienste
<code>\$> sudo nano /etc/gitlab/gitlab.rb</code>	Bearbeitet die Gitlab-Konfigurationsdatei.
<code>\$> gitlab-ctl --help</code>	Zeigt die vielzahl an Steuerungsoptionen
<code>\$> sudo gitlab-ctl upgrade</code>	Führt ein Update aus
<code>\$> sudo apt install nmap -y</code> <code>\$> nmap localhost</code>	Listet alle offenen Ports auf, um sicherzustellen, dass der GitLab-Dienst ordnungsgemäß kommunizieren kann.

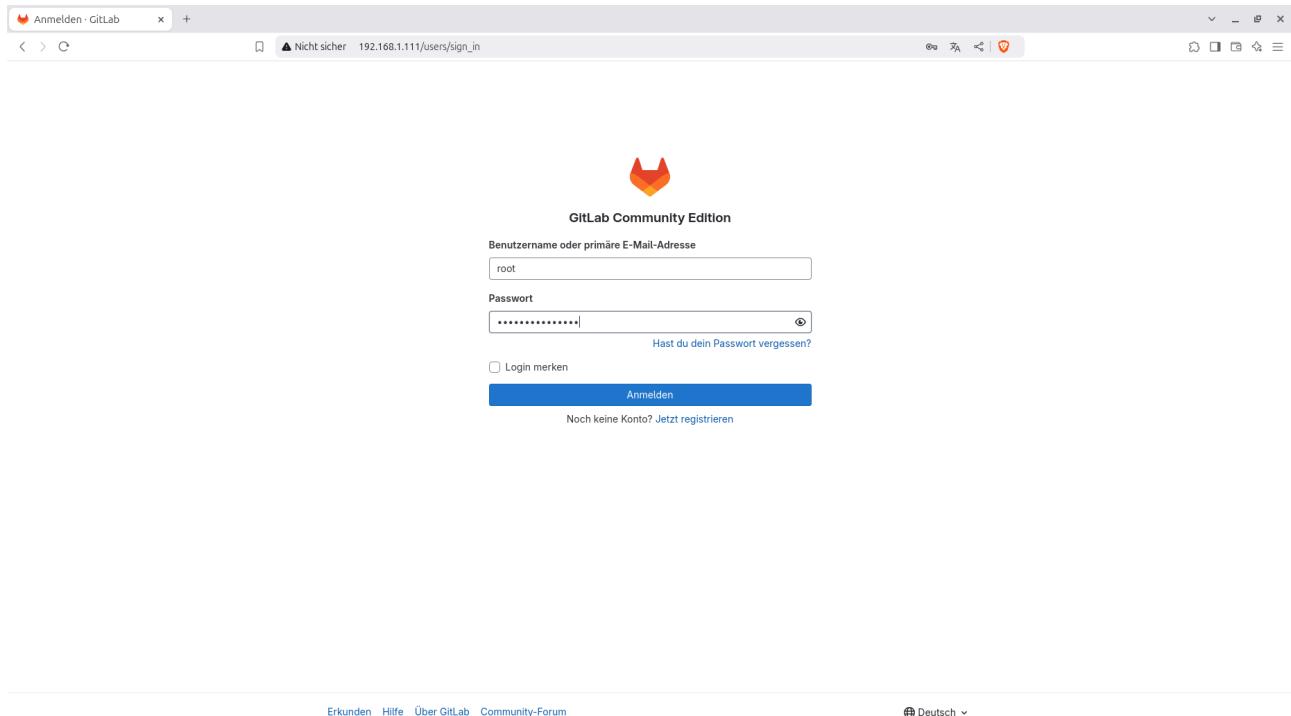
6. Resümee

Der Gitlab-Server ist nicht pflegeleicht und kein Leichtgewicht. Je nach Nutzungszweck ist nicht nur das Installations-Wissen oder Wartungs-Wissen nötig. Im Betrieb sind die Anwender-Kenntnisse und Features höher gewichtet.

7. GitLab-User anlegen

Lege auf dem Web-Gui einen neuen Gitlab-User an, der sich vom Root-Account unterscheidet.

Für das erste Login ist der Account «root» zu verwenden.



Anmelden - GitLab

Nicht sicher 192.168.1.111/users/sign_in

GitLab Community Edition

Benutzername oder primäre E-Mail-Adresse

root

Passwort

Hast du dein Passwort vergessen?

Login merken

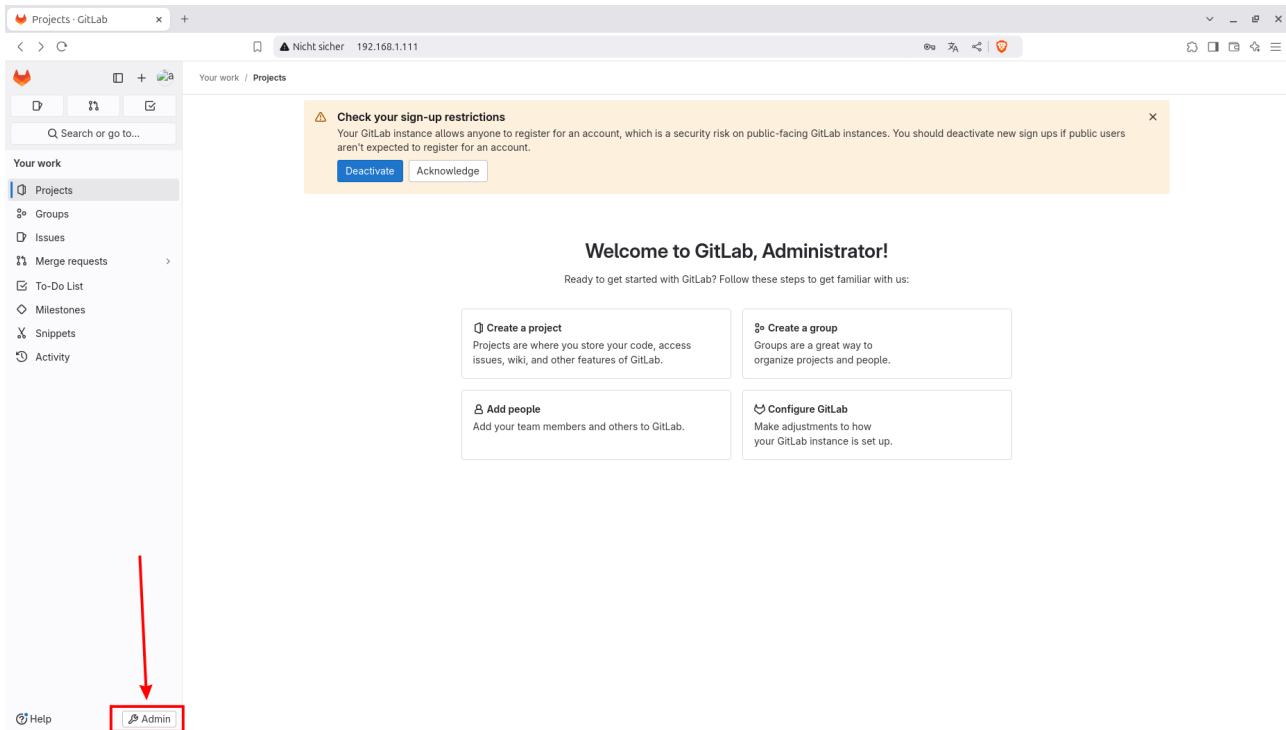
Anmelden

Noch keine Konto? Jetzt registrieren

Erkunden Hilfe Über GitLab Community-Forum

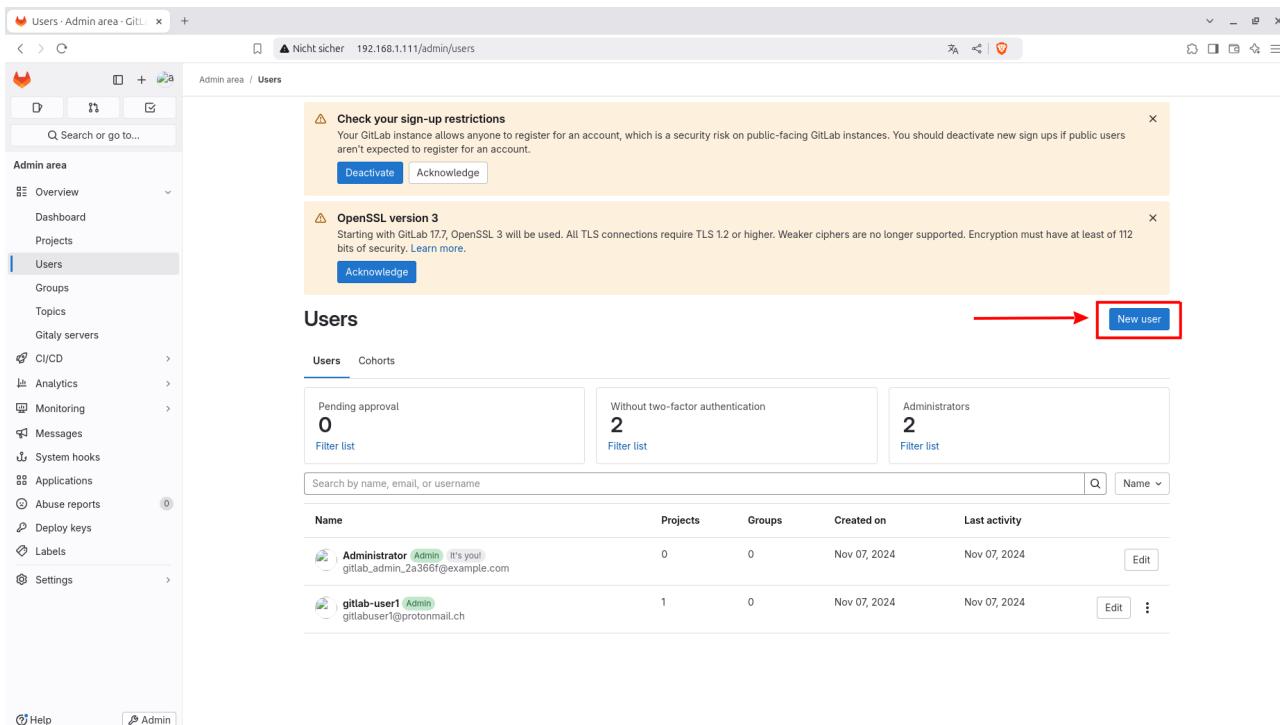
Deutsch

Wechsle auf die Admin-Seite wo du auf eine Nutzer-Übersicht gelangst.



The screenshot shows the GitLab main interface. On the left, a sidebar titled 'Your work' lists 'Projects', 'Groups', 'Issues', 'Merge requests', 'To-Do List', 'Milestones', 'Snippets', and 'Activity'. On the right, a central area displays a 'Welcome to GitLab, Administrator!' message with four initial steps: 'Create a project', 'Create a group', 'Add people', and 'Configure GitLab'. At the bottom left, there is a 'Help' link and an 'Admin' link. A red arrow points from the 'Admin' link in the sidebar to the 'Admin' link in the bottom right corner of the main content area.

Erstelle einen neuen GitLab-Benutzer:



The screenshot shows the GitLab Admin area. The sidebar on the left includes 'Admin area', 'Overview', 'Dashboard', 'Projects', 'Users' (which is selected and highlighted in blue), 'Groups', 'Topics', 'GitLab servers', 'CI/CD', 'Analytics', 'Monitoring', 'Messages', 'System hooks', 'Applications', 'Abuse reports', 'Deploy keys', 'Labels', and 'Settings'. The main content area is titled 'Users' and shows a summary of user status: 'Pending approval: 0', 'Without two-factor authentication: 2', and 'Administrators: 2'. Below this is a table of users with columns: Name, Projects, Groups, Created on, and Last activity. Two users are listed: 'Administrator Admin' (it's you) and 'gitlab-user1 Admin'. A red arrow points from the 'New user' button in the top right of the user summary to the 'New user' button in the bottom right of the user table.

Name	Projects	Groups	Created on	Last activity
Administrator Admin (it's you) gitlab_admin_2a366f@example.com	0	0	Nov 07, 2024	Nov 07, 2024
gitlab-user1 Admin gitlabuser1@protonmail.ch	1	0	Nov 07, 2024	Nov 07, 2024

Wähle passende Credentials und siehe dir die möglichen Einstellungen genau an:

New user

Account

Name:

Username:

Email:

Password

Reset link will be generated and sent to the user. User will be forced to set the password on first sign in.

Access

Projects limit: 100000

Can create top level group

Private profile

Access level

Regular

Regular users have access to their groups and projects.

Administrator

The user has unlimited access to all groups, projects, users, and features.

[Create user](#) [Cancel](#)

In unserem Fall existiert noch kein Mail-Server, richte für den Benutzer X also ein Passwort ein:

(Wenn ein Mail-Server definiert ist, erhält die Person eine Email mit einem Link. Das Passwort wird dann von der betreffenden Person selber festgelegt.)

gitlabuser2

User was successfully created.

[Impersonate](#) [New identity](#) **Edit** [...](#)

gitlabuser2

Profile page: [gitlabuser2](#)

Account

Name: gitlabuser2

Username: gitlabuser2

Email: gitlabuser2@protonmail.ch Verified

Email confirmation code last sent at: never

Locked account email verification code last sent at: never

ID: 3

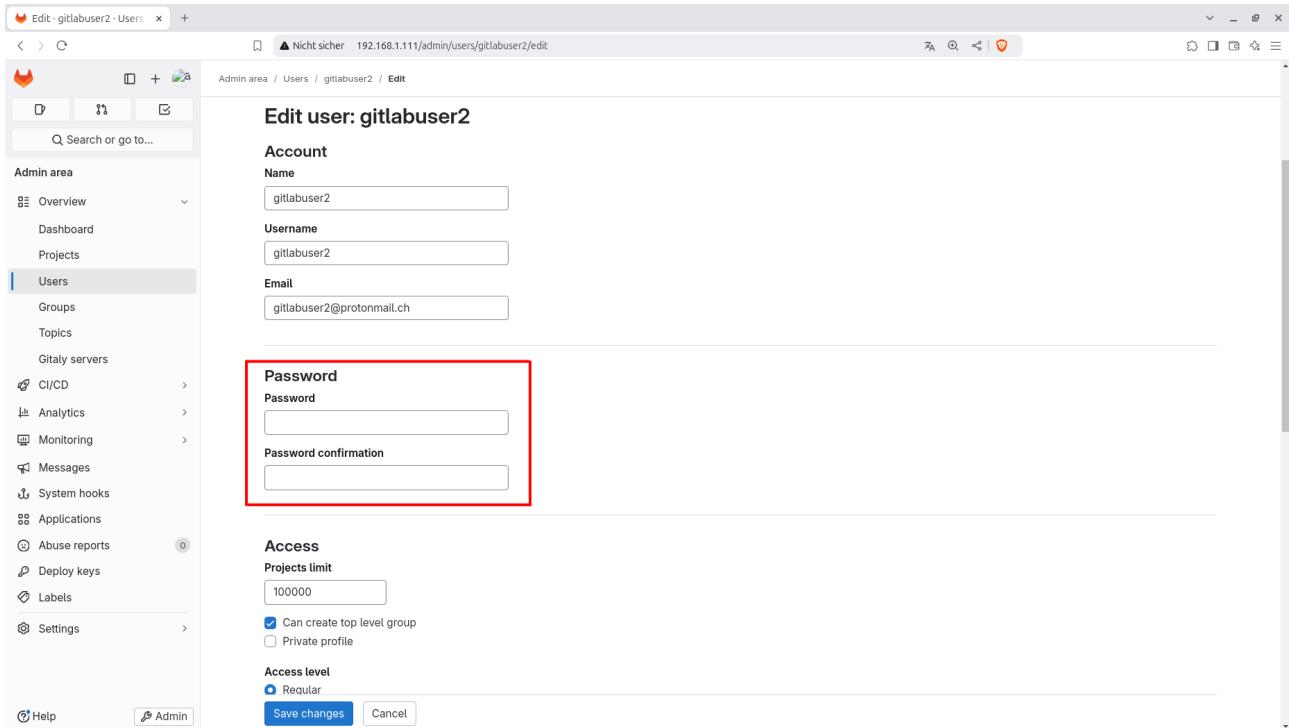
Namespace ID: 4

Two-factor Authentication: **Disabled**

Profile

Member since Nov 7, 2024 6:36pm

Auch hier - Sieh dir die möglichen Einstellungen genau an!



The screenshot shows the 'Edit user: gitlabuser2' page in the GitLab Admin Area. The 'Password' section is highlighted with a red box. The 'Access' section is also visible. The 'Save changes' button is at the bottom.

Account

- Name: gitlabuser2
- Username: gitlabuser2
- Email: gitlabuser2@protonmail.ch

Password

- Password: [redacted]
- Password confirmation: [redacted]

Access

- Projects limit: 100000
- Can create top level group:
- Private profile:

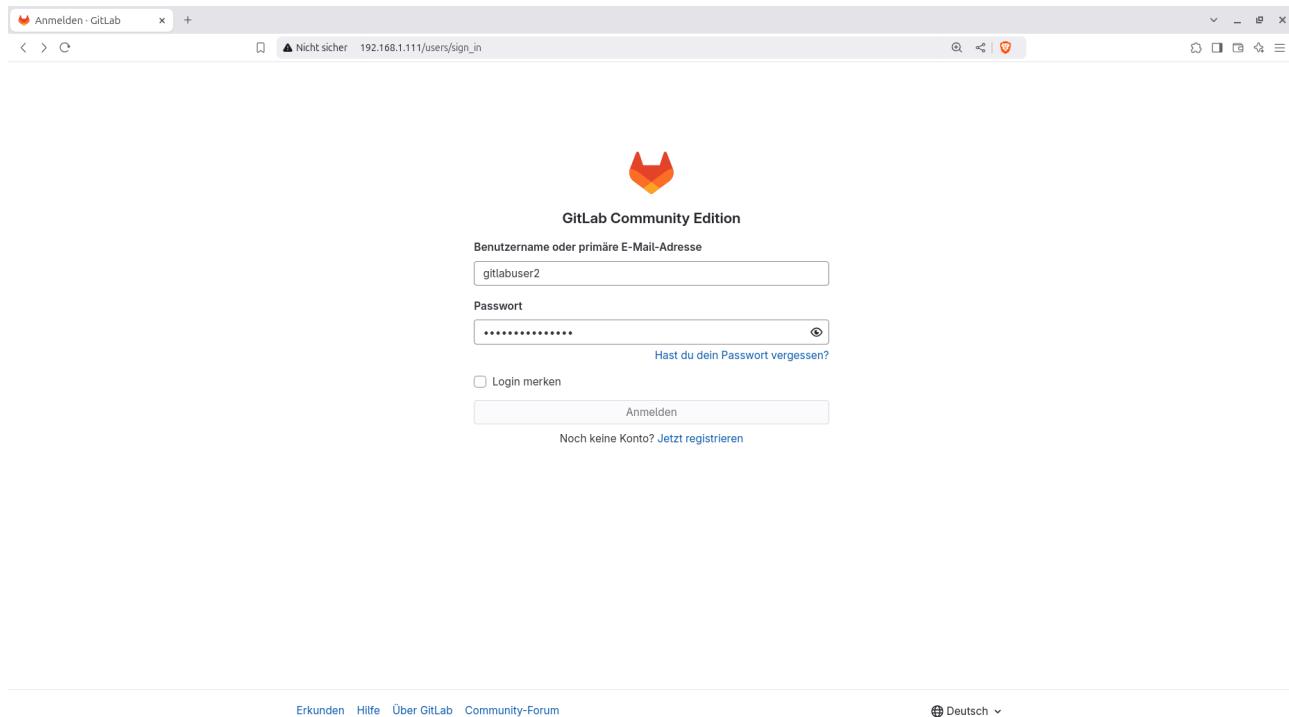
Access level

- Regular

Buttons

- Save changes
- Cancel

Logge dich mit dem neuen Benutzer ein:



The screenshot shows the 'Anmelden - GitLab' page. The 'Password' field is filled with '*****'. The 'Anmelden' button is at the bottom.

GitLab Community Edition

Benutzername oder primäre E-Mail-Adresse
gitlabuser2

Passwort

Hast du dein Passwort vergessen?

Login merken

Anmelden

Noch keine Konto? [Jetzt registrieren](#)

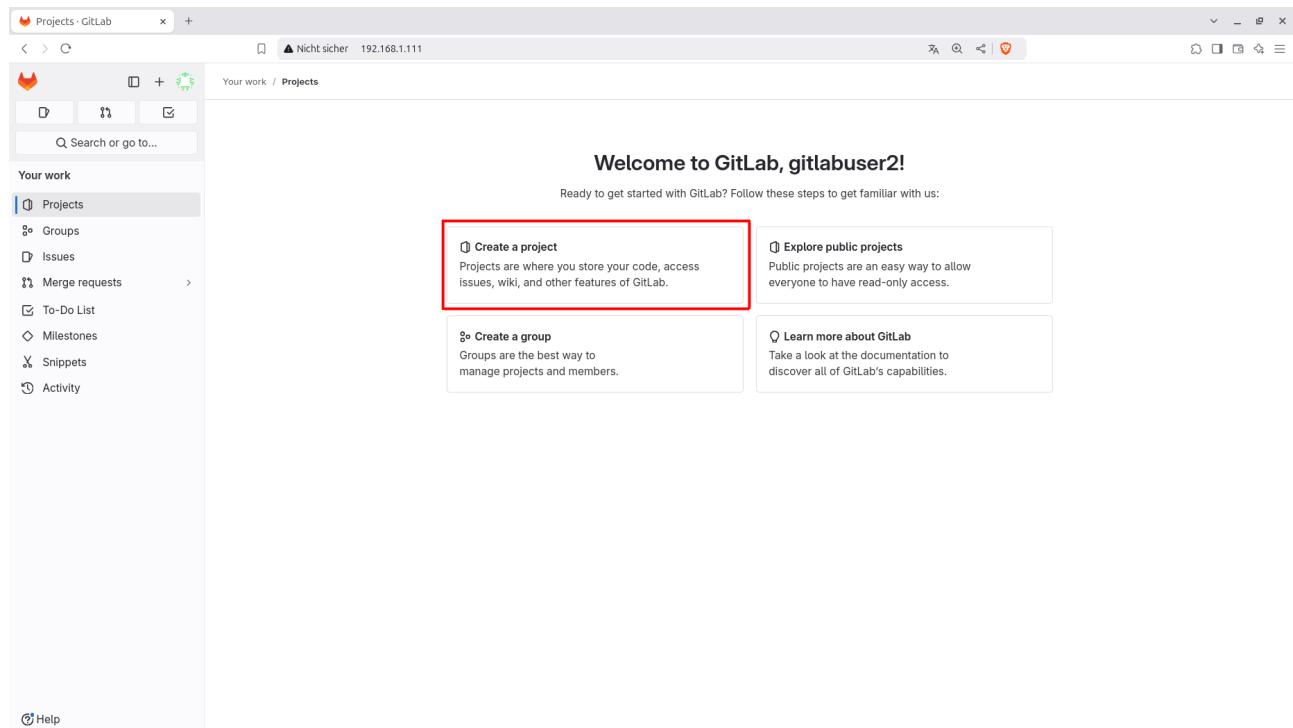
[Erkunden](#) [Hilfe](#) [Über GitLab](#) [Community-Forum](#)

Deutsch

8. GitLab-Projekte anlegen

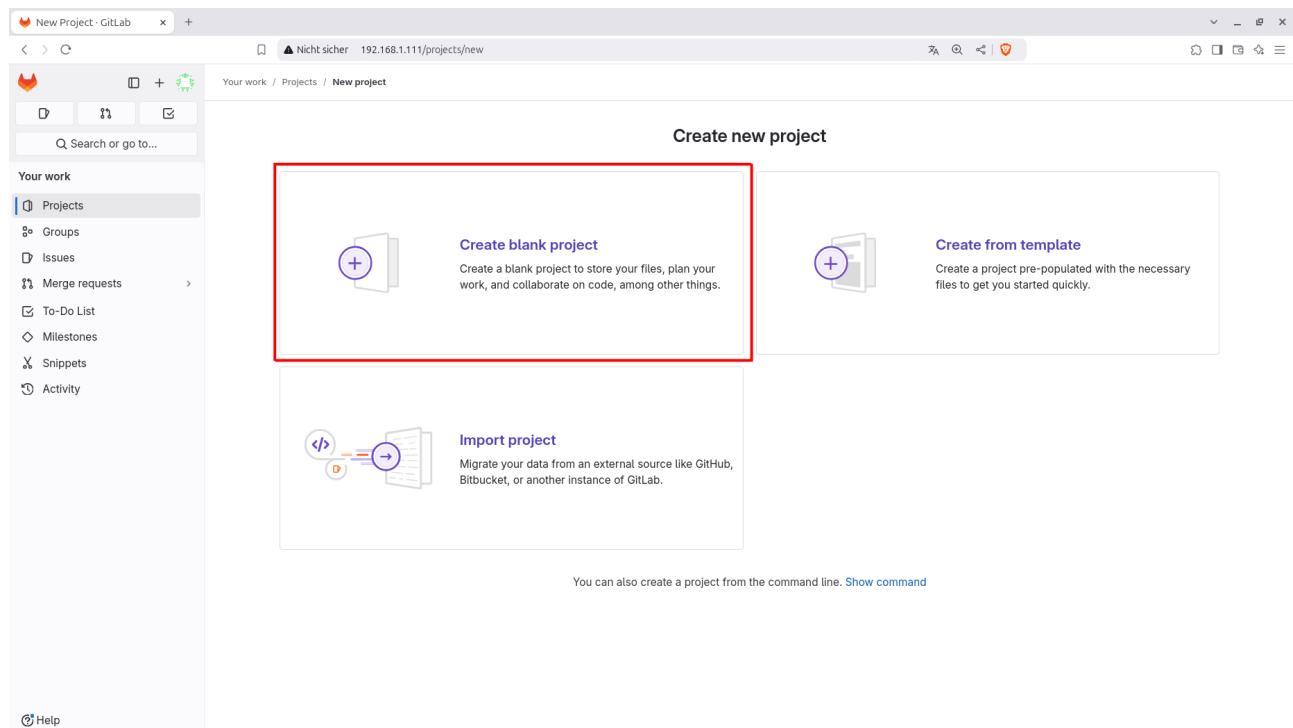
Lege unter dem neuen User ein Projekt an.

Hinweis: ein Projekt kann auch über die Kommando-Zeile angelegt werden, jedoch ist es für Einsteiger einfacher dies über das Web-Gui zu erledigen:



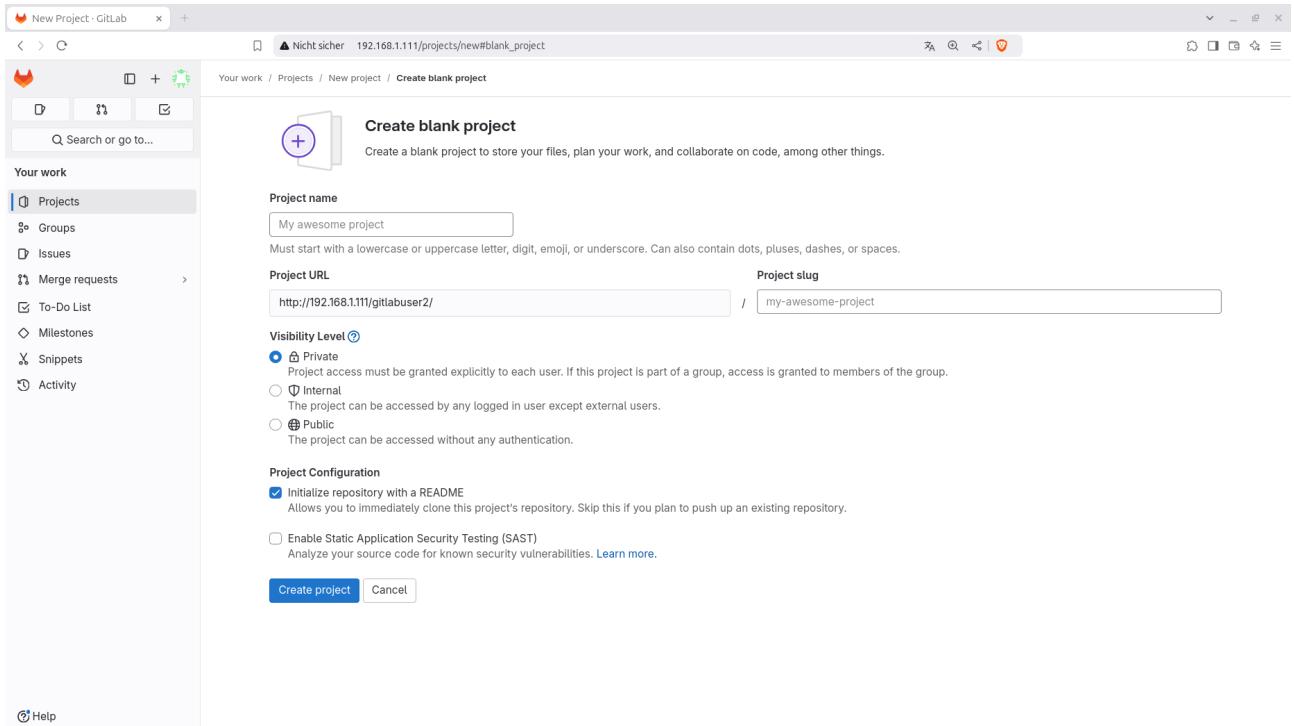
The screenshot shows the GitLab 'Welcome to GitLab' page for the user 'gitlabuser2!'. The left sidebar is titled 'Your work' and includes 'Projects' (selected), 'Groups', 'Issues', 'Merge requests', 'To-Do List', 'Milestones', 'Snippets', and 'Activity'. The main content area is titled 'Welcome to GitLab, gitlabuser2!' and contains a 'Ready to get started with GitLab? Follow these steps to get familiar with us:' section. It includes four cards: 'Create a project' (highlighted with a red box), 'Explore public projects', 'Create a group', and 'Learn more about GitLab'. The 'Create a project' card describes projects as storage for code, access issues, and other features.

Immer schön easy bleiben... und ein leeres Projekt erstellen – man kann die Funktionalitäten damit unverfälscht testen und erkunden:



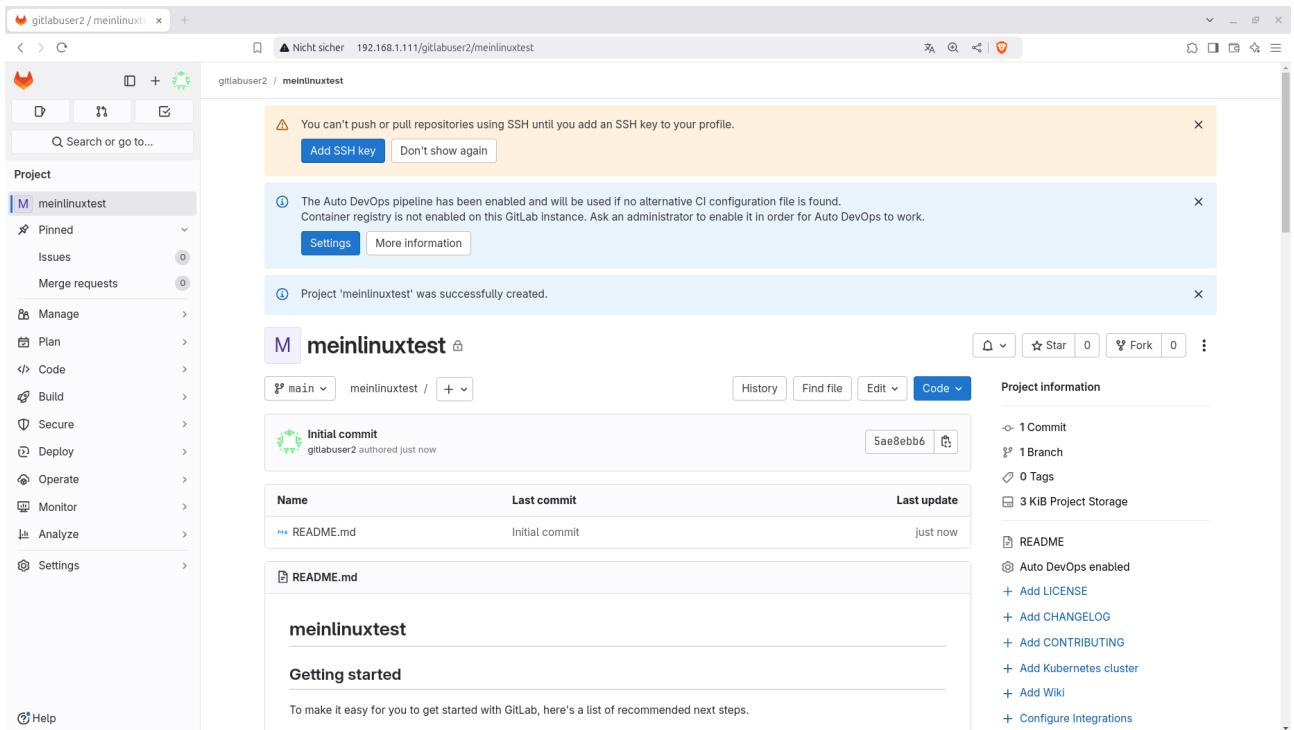
The screenshot shows the 'Create new project' page in GitLab. The left sidebar is identical to the previous screenshot. The main content area is titled 'Create new project' and features three options: 'Create blank project' (highlighted with a red box), 'Create from template', and 'Import project'. The 'Create blank project' card includes a description: 'Create a blank project to store your files, plan your work, and collaborate on code, among other things.' At the bottom, it says 'You can also create a project from the command line. [Show command](#)'.

Erstelle ein Projekt nach deinem Geschmack! Sieh dir die möglichen Einstellungen genau an. Wir kratzen nur an der Oberfläche und haben den Server für professionellen Bedarf noch nicht komplett eingerichtet. Wichtig: Vermeide Tippfehler und wähle die Projekt-Namen klug!



The screenshot shows the 'Create blank project' page in GitLab. The 'Project name' field contains 'My awesome project'. The 'Visibility Level' is set to 'Private'. The 'Project URL' is 'http://192.168.1.111/gitlabuser2/'. The 'Project slug' is 'my-awesome-project'. The 'Create project' button is highlighted in blue.

Dein Test-Projekt wurde erfolgreich erstellt. Hier findest du die Projekt-Übersicht:



The screenshot shows the project overview for 'meinlinuxtest'. The project has 1 commit, 1 branch, and 3 kB of storage. A 'Getting started' section is present.

Für weiter Schritte ist es hilfreich unter «Code» die http-Sektion des Projektes zu kopieren:

The screenshot shows a GitLab project page for 'meinlinuxtest'. The 'Code' section is highlighted with a red box. It contains links for cloning the repository via SSH and HTTP, as well as options for opening it in various IDEs (Visual Studio Code, IntelliJ IDEA) and downloading the source code in zip, tar.gz, tar.bz2, or tar formats.

9. GitLab-Projekt klonen (Push&Pull)

\$> ssh CLIENT1@192.168.1.X	Stelle eine SSH-Verbindung zum Client über eine IP-Adresse her.
\$> sudo apt install git	Installiere Git auf deinem Client
\$> mkdir projekte	Erstelle einen Ordner für deine Projekte
\$> cd projekte	Wechsle in das neue Verzeichnis
\$> git clone http://DeineIP/User/PROJEKT.git	Klone dein Git-Repository in das neue Verzeichnis
\$> touch neuedatei1	Erstelle eine neue Datei
Dein Projekt-Repository ist nun vom Server-Stand abweichend. Dein lokales Projekt hat jetzt einen aktuelleren Status als der GitLab-Server.	
\$> git status	Zeige den aktuellen Status an
\$> git add .	Füge alle neuen Dateien der Änderung zu
\$> git commit -m «neue-Datei»	Kommentiere deine Änderung
\$> git status	Zeige den aktuellen Status an
\$> git push	Upload «pushe» dein Änderung
\$> git status	Zeige den aktuellen Status an

GitLab-Projekt von einem anderen eingerichteten Client auf den neuesten Stand bringen
(einfaches Pull-Beispiel wenn mehrere Benutzer das gleiche Projekt betreuen)

\$> ssh CLIENT1@192.168.1.Y	Stelle eine SSH-Verbindung zum Client über eine IP-Adresse her.
-----------------------------	---

\$> cd projekte	Wechsle in das Projekt-Verzeichnis
\$> git pull	Download «pull» dein Änderung vom Server.

10. Resümee

Wer zum ersten Mal mit GitLab in der Kommandozeile arbeitet, muss einiges an Geduld und Durchhaltevermögen mitbringen – vor allem, wenn man sich das Thema alleine erarbeiten muss. Fehlerbehandlung und unterschiedliche Betriebsmodi wurden in diesem Zusammenhang noch gar nicht angesprochen. Zwar gibt es viele Anleitungen im Internet, doch die Herangehensweisen variieren oft stark. Dieses Thema könnte problemlos eine eigene Anleitung füllen. Mein Tipp: Wenn du dich mit GitLab vertraut machst, geh es spielerisch an – etwa mit einem Testprojekt. Experimentiere, mache Fehler und lerne daraus.

11. Schlussbemerkung

Die GitLab-Technologie wurde hier anhand eines einfachen Beispiels vorgestellt. Mit GitLab können Projekte versioniert und geteilt werden – in diesem Fall anschaulich über die Kommandozeile. Viele grafische Entwicklungsumgebungen nutzen ebenfalls GitLab. Dadurch können mehrere Entwickler gleichzeitig an einem Projekt arbeiten und den Versionsstand aktuell halten.

Allerdings erfordert dies eine koordinierte Zusammenarbeit. Mit dieser Technologie hat GitLab nicht nur einen neuen Ansatz in der Softwareentwicklung etabliert, sondern auch eine neue Ideologie des Entwicklungsprozesses geprägt. Die Entwicklung kann nun nicht nur dezentral organisiert werden, sondern es ist auch möglich, in alle Richtungen zu arbeiten.

Entwickler können unabhängig an Projekten arbeiten oder diese mit anderen teilen – unabhängig davon, ob nur ein einzelner Entwickler oder ein ganzes Team beteiligt ist oder ob einfach nur Entwicklungen ausgetauscht werden.

Hier wurde lediglich eines der einfachsten Beispiele sowie die Installation demonstriert. Dabei wurden Fehlerbehandlung und Versionskonflikte noch nicht berücksichtigt.