



Jonas Michler

## GitHub und GitLab

Plattformen für das Teilen von Code

# GitHub und GitLab

Wer hat...

- ...davon bereits gehört?
- ...eines davon bereits genutzt?
- ...einen GitHub Account?

# Gliederung

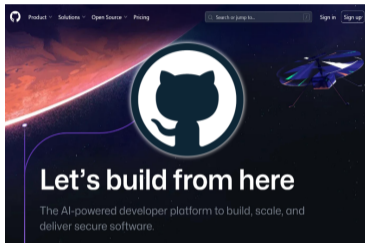
- 1 Cloud-basierte Entwicklung
- 2 GitHub: Erste Schritte
- 3 GitHub + GitLab
- 4 Contribution
- 5 Zusammenfassung

# Was sind GitHub und GitLab?

kurz:

- Cloud-Dienste zur Versionsverwaltung für Software-Entwicklungsprojekte

# Was ist Git?



# Vorteile von GitHub und GitLab

- Versionsverwaltung von Code
  - ▶ Effiziente Verwaltung und Nachverfolgung von Codeänderungen
  - ▶ Historie zur Wiederherstellung früherer Versionen
- Kollaborative Softwareentwicklung
  - ▶ Gleichzeitige Zusammenarbeit mehrerer Entwickler
  - ▶ Effektive Zusammenarbeit in dezentralen Teams
- Unterstützung von Open Source und Privaten Projekten
  - ▶ Plattformen ermöglichen öffentliche und private Repositories
  - ▶ Schutz des geistigen Eigentums in privaten Projekten

# Cloud-basierte Versionskontrolle

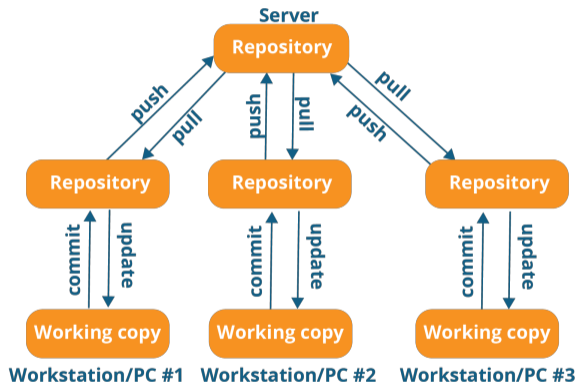
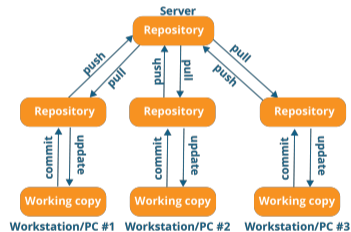


Figure: Distributed Version Control

# Cloud-basierte Versionskontrolle

- Lokales Arbeitsrepository
- Änderungen können hochgeladen (gepusht) werden
- Änderungen von Teammitgliedern können aus der Cloud heruntergeladen werden
- Konflikte können auftreten, wenn mehrere Personen gleichzeitig arbeiten





# GitHub vs. GitLab

## GitHub

- Beliebte Plattform für Versionsverwaltung
- Hauptsächlich für Open Source-Projekte
- Funktionen: Issues, Pull Requests, Actions
- Cloud-basiert

## GitLab

- Alternative mit ähnlichen Funktionen
- Geeignet für Open Source und private Projekte
- Umfassende CI/CD-Integration
- Cloud und lokale Installation

# GitHub vs. GitLab

## Hauptunterschiede:

### GitHub

- Nur Cloud
- Unternehmens- und OS-Lösungen
- Selbst nicht Open Source
- vielseitiges Ökosystem  
*Copilot, Workspaces ...*

### GitLab

- Cloud und On-Premises
- Unternehmensorientierte Lösung
- komplett Open Source
- reine Git-Lösung

# GitHub

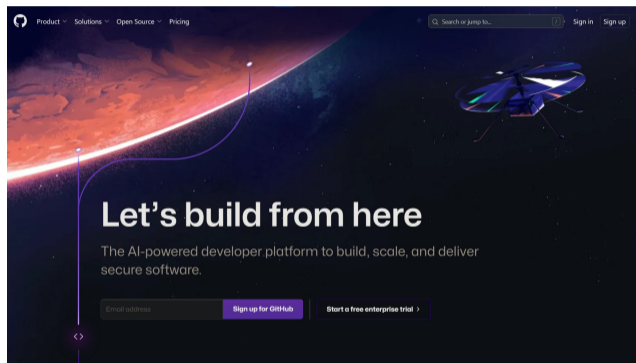


Figure: GitHub Homepage

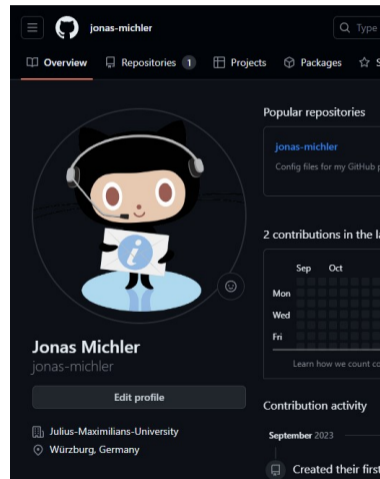
# GitHub: Anmeldung

- 1 <https://github.com> besuchen
- 2 Schaltfläche "Sign Up"
- 3 E-Mail-Adresse + sicheres Passwort
- 4 Benutzernamen wählen
- 5 E-Mail-Verifizierung



# GitHub: Profil einrichten

- 1 Profil-Icon (oben rechts), im Dropdown "Your profile"
- 2 Profilbild und eine kurze Beschreibung hinzufügen
- 3 Weitere Informationen hinzufügen
- 4 Soziale Links oder Website verknüpfen
- 5 Speichern



# GitHub-Profil: Überblick

## Was Ihr GitHub-Profil zeigt:

- Aktuelle Projekte
- Verwendete Programmiersprachen
- Ihre Follower und gefolgteten Entwickler

## Warum ist es wichtig?

- Arbeitgeber und Teams prüfen oft GitHub-Profile
- Zeigt Leidenschaft für Softwareentwicklung
- Dient als Portfolio Ihrer Fähigkeiten
- Ermöglicht Networking und Zusammenarbeit

# Repository-Oberfläche

- Code und Dateien im Repository
- README-Datei und Projektinformationen
- Contributors und Aktivitäten
- Branches und Pull Requests
- Projektstatistiken und Insights

Repos:

- `https://github.com/facebook/react`
- `https://github.com/signalapp`
- `https://github.com/audacity/audacity`

# GitLab: Repository-Oberfläche

The screenshot shows the GitLab repository page for the 'gitlab' project. The left sidebar contains navigation options: Projekt, Angehört, Tickets, Merge Requests, Verwalten, Aktivität, Mitglieder, Labels, Planen, Code, Build, Bereitstellung, Betreiben, Überwachen, Analysieren, and Hilfe. The main content area displays the repository name 'GitLab', project ID '278964', and statistics: 359,363 Commits, 15,830 Branches, 2,217 Tags, 67,4 TiB Projekt Speicher, 139 Veröffentlichungen, and 4,318 Umgebungen. A description states: 'GitLab is an open source end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD, and more. Self-host GitLab on your own servers, in a container, or on a cloud provider.' Below this is a progress bar for CI coverage (66.79%) and a 'Contribute' button. A recent commit is shown: 'Merge branch '416075-doc-continuous-vuln-scans' into 'master'' by Russell Dickenson. The bottom section shows a file tree with columns for Name, Letzter Commit, and Letzte Aktualisierung.

Name	Letzter Commit	Letzte Aktualisierung
github	Rename GitLab CE to FOSS in GitHub issue tem...	vor 3 Jahren
gitlab	Merge branch 'codeowners_auto_devops_temp...	vor 2 Tagen
lefthook/pre-push	Add security harness to Lefthook	vor 5 Monaten
rubocop_todo	Remove more redundant freeze	vor 3 Tagen
vscode	Add GitLab Workflow as recommended VS Cod...	vor 1 Jahr
app	Merge branch 'smrll-422028/salesforce_logo...	vor 1 Tag
bin	Add vite for javascript assets	vor 2 Wochen

Figure: GitLab Repository

sample source: <https://gitlab.com/gitlab-org/gitlab>



# GitLab: Profil-Oberfläche

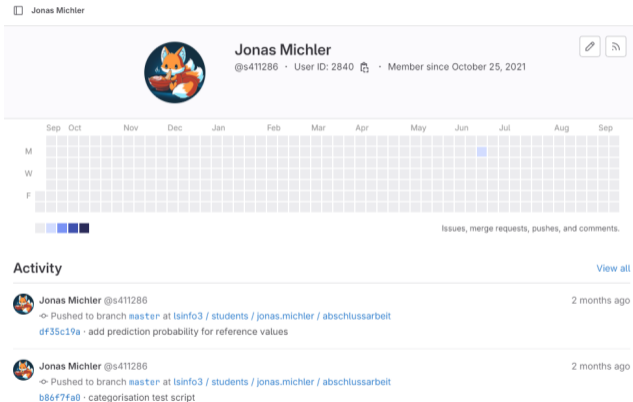


Figure: GitLab Profil

# Contribution: Diverse Möglichkeiten zur Mitarbeit

## ■ **Code-Beitrag:**

- ▶ Schreiben neuer Funktionen
- ▶ Beheben von Fehlern
- ▶ Issues bearbeiten

## ■ **Fehlerberichte (Bug Reports):**

- ▶ Melden von Softwarefehlern oder Problemen
- ▶ Hilft bei der Fehlerbehebung und Verbesserung der Stabilität

## ■ **Dokumentation, Support und Diskussion:**

- ▶ Dokumentation/Wiki/FAQ ausarbeiten
- ▶ Beantworten von Fragen anderer Benutzer
- ▶ Teilnahme an Diskussionen und Entscheidungsfindung

# Erstellen detaillierter Fehlerbeschreibungen/Issues

- Issues sind der Schlüssel zur Verfolgung von Fehlern und Verbesserungen
- Beschreiben Sie Fehler so detailliert wie möglich:
  - ▶ Was ist das fehlerhafte Verhalten?
  - ▶ Welche Schritte führen zum Fehler?
  - ▶ Was ist das zu erwartende Verhalten?
  - ▶ Welche Umgebung verwenden Sie?
  - ▶ -> Guidelines beachten
- Klare Beschreibungen erleichtern das Verständnis und die Replizierbarkeit
- Issues sind **nicht** der Ort, um seine persönliche Meinung zum Produkt zu veröffentlichen

## Code-Beiträge an spezifische Probleme/Issues binden

- Beiträge, Forks und Pull Requests sollten immer zu einem spezifischen Problem gehören
- Diese Verknüpfung ermöglicht eine klare Zuordnung von Änderungen
- Verbessert die Transparenz und Zusammenarbeit innerhalb des Projekts
  
- Viele Projekte besitzen eine Dokumentation, wie Beiträge einzureichen sind
  - ▶ z.T. extern gelöst
  
- Contribution.md, CodeOfConduct.md, SPDX

# Code-Beitrag auf GitHub/GitLab

## ■ Schritt 1: Fork des Projekts

- ▶ Erstellen einer Kopie des Repositorys auf Ihrem eigenen Konto

## ■ Schritt 2: Code ändern

- ▶ Bearbeiten des Codes in Ihrem Fork

## ■ Schritt 3: Pull/Merge Request erstellen

- ▶ Reichen Sie Ihre Änderungen als Pull Request (PR) beim Hauptprojekt ein

## ■ Schritt 4: Review Phase

- ▶ Andere Entwickler überprüfen und diskutieren Ihre Änderungen.

## ■ Schritt 5: Merge

- ▶ Wenn Ihre Änderungen genehmigt werden, werden sie in das Hauptprojekt integriert

# Repository erstellen

- 1 Klicken Sie auf die Schaltfläche “New” oder “+”
- 2 Geben Sie einen Projektnamen und eine Beschreibung ein
- 3 Legen Sie die Sichtbarkeit (öffentlich oder privat) fest
- 4 Fügen Sie eine README-Datei hinzu, um Informationen über Ihr Projekt bereitzustellen
- 5 Fügen Sie Ihrem Projekt eine Lizenz bei

# GitHub + GitLab: Ihre Plattformen für kollaborative Entwicklung



## Warum GitHub und GitLab nutzen?

- Gemeinschaftliche Arbeit an einem Projekt
- Dokumentation von Entwicklungsschritten
- Öffentliche Bereitstellung des Source-Code
- Möglichkeit spezifisches Feedback zu erhalten