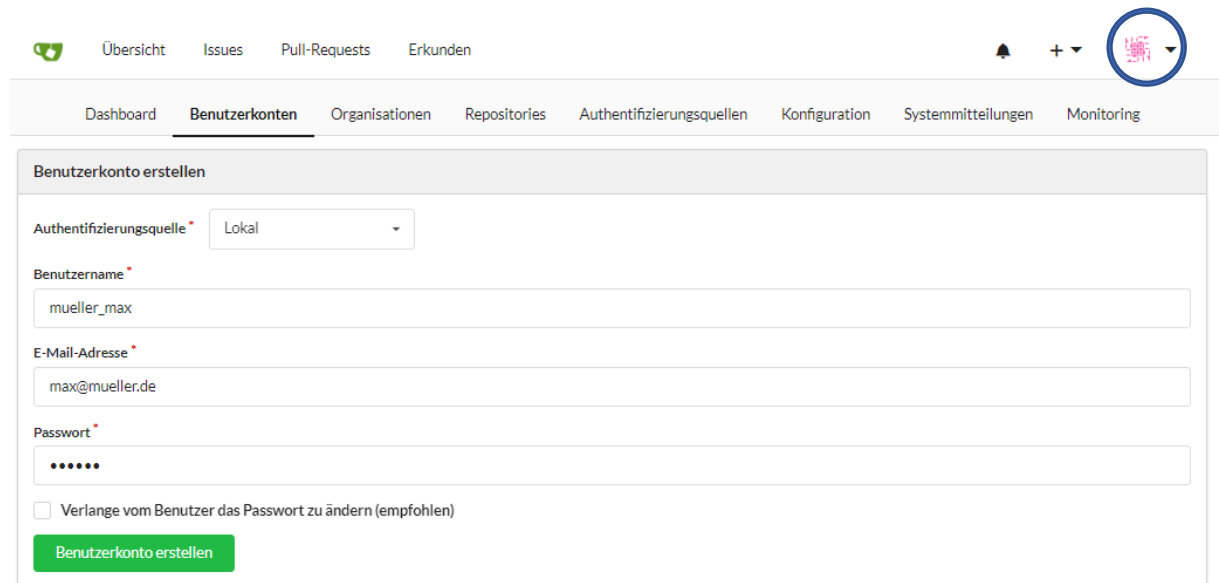


Anleitung-Projekt anlegen

Giteanutzer anlegen – Optional

Unter Webseiten-Administration → Benutzerkonten → Benutzerkonto erstellen



Übersicht Issues Pull-Requests Erkunden

Dashboard **Benutzerkonten** Organisationen Repositories Authentifizierungsquellen Konfiguration Systemmitteilungen Monitoring

Benutzerkonto erstellen

Authentifizierungsquelle * Lokal

Benutzername *
mueller_max

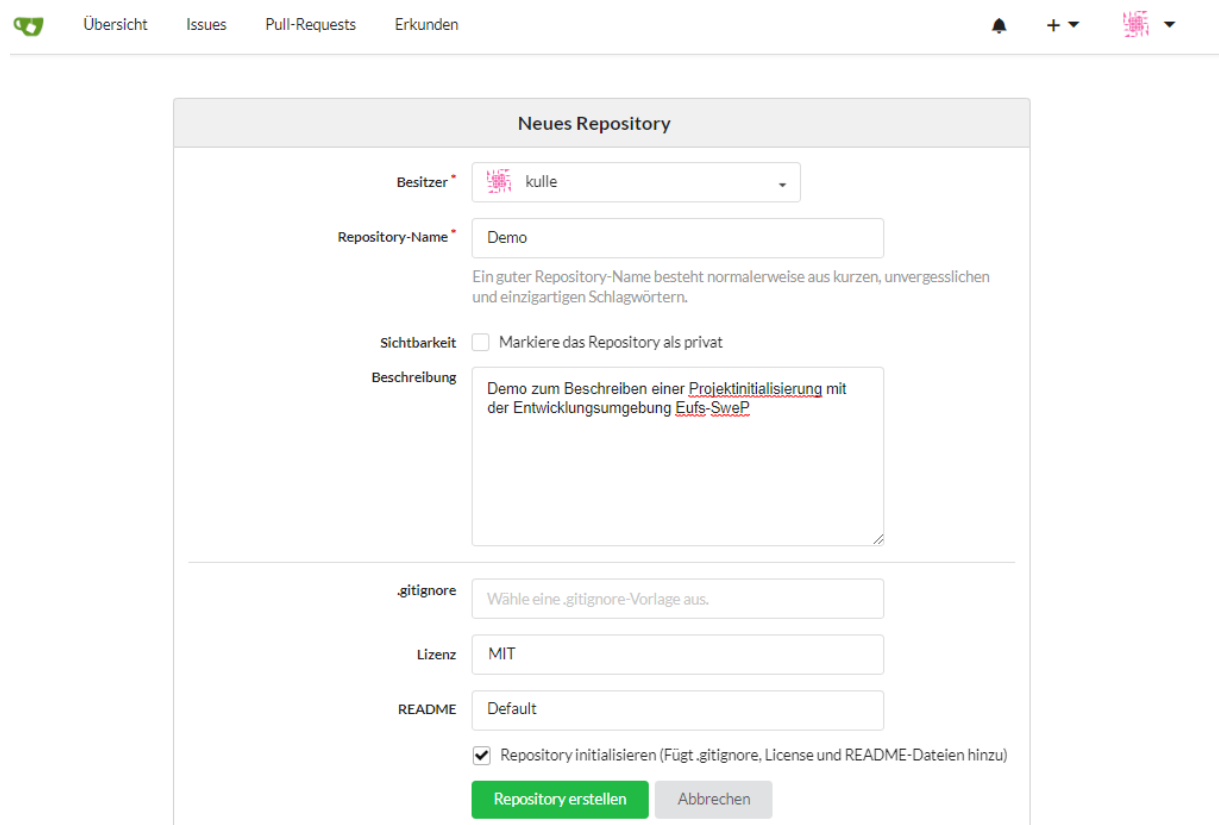
E-Mail-Adresse *
max@mueller.de

Passwort *
•••••

Verlange vom Benutzer das Passwort zu ändern (empfohlen)

Benutzerkonto erstellen

Repository initialisieren



Übersicht Issues Pull-Requests Erkunden

Neues Repository

Besitzer * kulle

Repository-Name * Demo
Ein guter Repository-Name besteht normalerweise aus kurzen, unvergesslichen und einzigartigen Schlagwörtern.

Sichtbarkeit Markiere das Repository als privat

Beschreibung
Demo zum Beschreiben einer Projektinitialisierung mit der Entwicklungsumgebung Eufs-SweP

.gitignore Wähle eine .gitignore-Vorlage aus.

Lizenz MIT

README Default

Repository initialisieren (Fügt .gitignore, License und README-Dateien hinzu)

Repository erstellen Abbrechen

Repository an Giteauser übergeben

Einstellungen → Besitz übertragen

Besitz übertragen

- Du wirst keinen Zugriff mehr haben, wenn der neue Besitzer ein individueller Benutzer ist.
- Du wirst weiterhin Zugriff haben, wenn der neue Besitzer eine Organisation ist und du einer der Besitzer bist.

Gib den Repository-Namen zur Bestätigung ein: **Demo**

Repository-Name *

Neuer Besitzer *

Abbrechen

Transfer durchführen

GITEA-Link kopieren

Übersicht Issues Pull-Requests Erkunden

student / Demo

Beobachten beenden 1 Favorisieren 0 Fork 0

Code Issues 0 Pull-Requests 0 Releases 0 Wiki Aktivität Einstellungen

Demorepository
Themen verwalten

4 Commits 1 Branch

Branch: master Demo

Neue Datei Datei hochladen HTTP SSH http://192.168.39.220:3000

Projekt in der Umgebung anmelden

Neues Projekt

Projektname:

GIT-Repo:

Anzahl Clients:

Wechsel in die IDE

Willkommen auf dem Projektserver

Projektübersicht

Name	Status	IPv4	IPv6	Start/Stop	Verbinden
uniDocker	RUNNING	192.168.39.241 (wHO)		<input type="button" value="Stopf"/>	
uniGit	RUNNING	192.168.39.220 (wHO)		<input type="button" value="Stopf"/>	Git-Server
uniIDE	RUNNING	192.168.39.230 (wHO)		<input type="button" value="Stopf"/>	IDE-Server
uniIDEdemo1	RUNNING	192.168.39.114 (wHO)		<input type="button" value="Stopf"/>	IDE-Server
uniIDEdemo2	RUNNING	192.168.39.115 (wHO)		<input type="button" value="Stopf"/>	IDE-Server
uniIDEWeb	RUNNING	192.168.39.107 (wHO)		<input type="button" value="Stopf"/>	IDE-Server
uniJen	RUNNING	192.168.39.221 (wHO)		<input type="button" value="Stopf"/>	Jenkins-Server
uniTicket	RUNNING	192.168.39.222 (wHO)		<input type="button" value="Stopf"/>	Ticket-System
uniVNC	RUNNING	192.168.39.240 (wHO)		<input type="button" value="Stopf"/>	VNC-Viewer: 192.168.39.240:5901

Neues Projekt

Projektname:

GIT-Repo:

Anzahl Clients:

Tutorials

Willkommen auf dem Projektserver

Name	Status	IPv4	IPv6	Start/Stop	Verbinden
uniDocker	RUNNING	192.168.39.114 (wHO)		<input type="button" value="Stopf"/>	
uniGit	RUNNING	192.168.39.220 (wHO)		<input type="button" value="Stopf"/>	Git-Server
uniIDE	RUNNING	192.168.39.230 (wHO)		<input type="button" value="Stopf"/>	IDE-Server
uniIDEWeb	RUNNING	192.168.39.108 (wHO)		<input type="button" value="Stopf"/>	IDE-Server
uniJen	RUNNING	192.168.39.221 (wHO)		<input type="button" value="Stopf"/>	Jenkins-Server
uniTicket	RUNNING	192.168.39.222 (wHO)		<input type="button" value="Stopf"/>	Ticket-System
uniVNC	RUNNING	192.168.39.240 (wHO)		<input type="button" value="Stopf"/>	VNC-Viewer: 192.168.39.240:5901

Neues Projekt

Projektname:

GIT-Repo:

Anzahl Clients:

Projektkalibrierung
Die nächsten Schritte sind Anlegen des Projektes innerhalb des GITEA bis hin zur Aufnahme der persönlichen Code der Studenten.

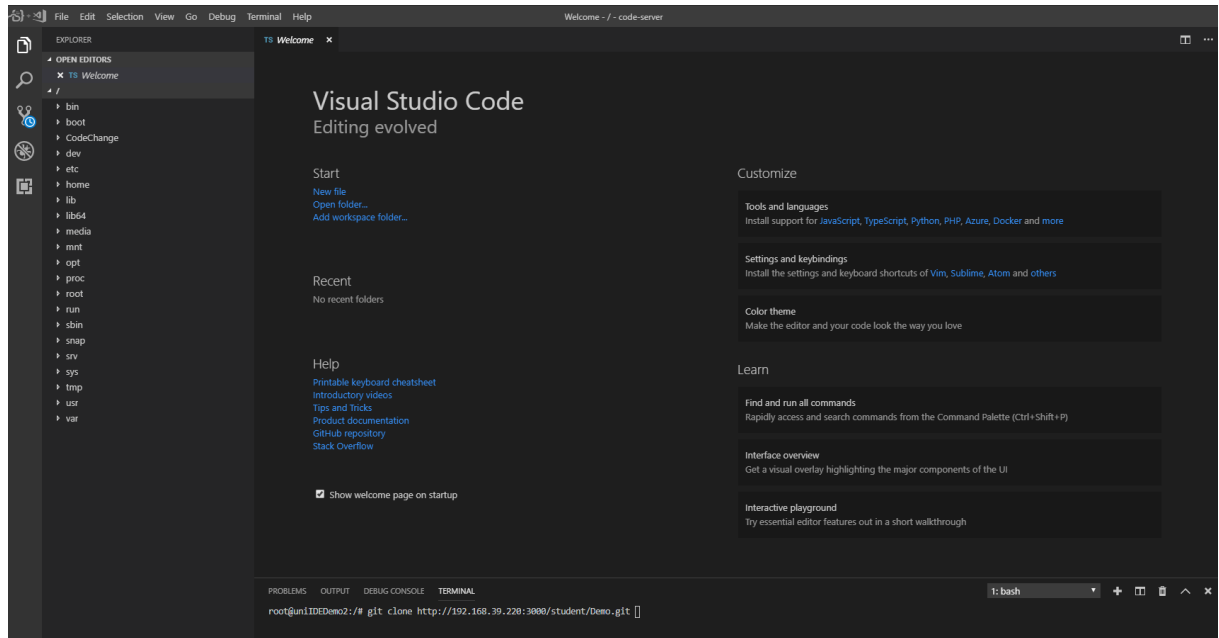
0:00 / 7:54

Coder initialisieren, dazu das GITEA-Repository clonen – Dazu:

File → OpenFolder → CONFIRM

Terminal → NEW TERMINAL

Im Terminal wird dann git clone <GITEA-LINK> eingegeben (siehe Abbildung)



Wechsel in das neue Verzeichnis dazu File → OpenFolder → <Ordner mit dem Repositorynamen> öffnen

Terminal → New Terminal

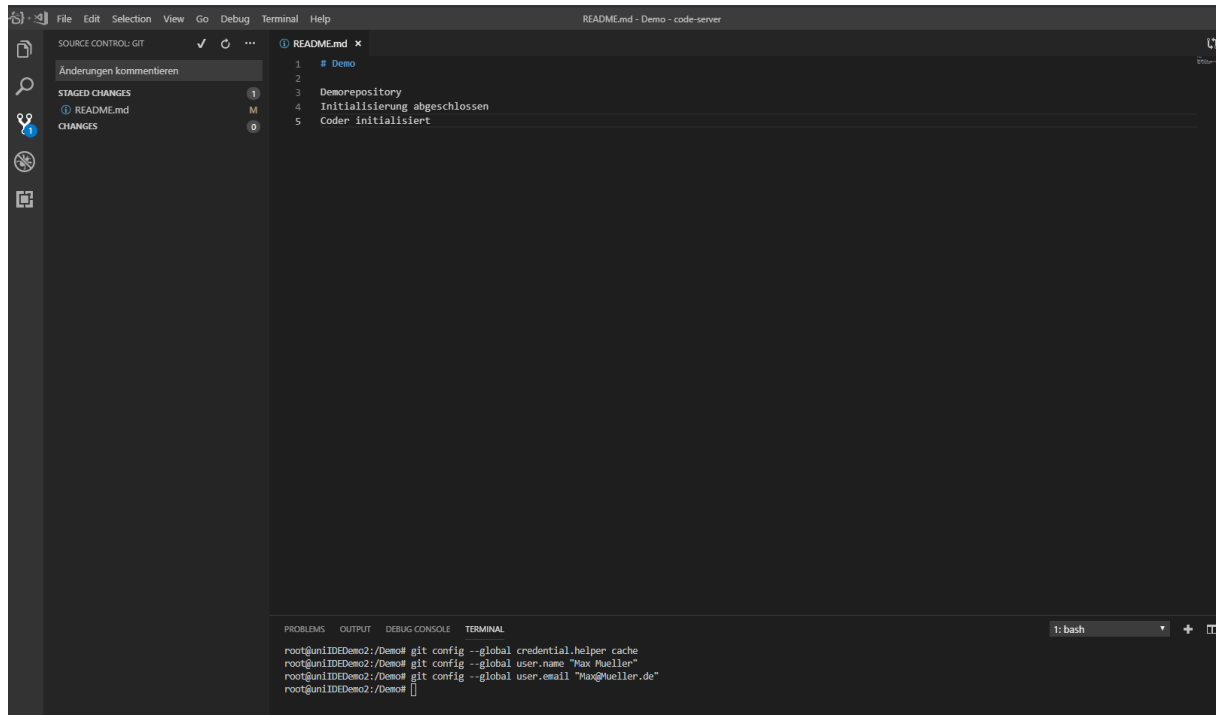
```
TERMINAL: git config --global credential.helper cache
```

```
TERMINAL: git config --global user.name "<studentName>"
```

```
TERMINAL: git config --global user.email "<StudentMail>"
```

Änderung an einer Datei vornehmen, speichern und diese Änderung kommentieren Sie wechseln in das GIT-Modul des Coders (1)

Unter Changes klicken sie auf das Plus um die Dateien in die Staged Changes aufzunehmen, dann oben den Kommentar eintragen und mit dem Hacken bestätigen



Im Terminal git push eingeben und die Benutzerdaten eingeben

Danach das Jenkinsfile wie folgt anlegen

```
pipeline {
  agent any
  stages {
    stage('Build') {
      steps {
        sh 'echo "Buildng.."'
        sh '''
        echo "Zeile 1"
        echo "Zeile 2"
        '''
      }
    }
    stage('Transfer') {
      steps {
        sh 'echo "Transfer.."'
        sh '''
        echo "Zeile 1"
        echo "Zeile 2"
        '''
      }
    }
    stage('Run') {
```

```

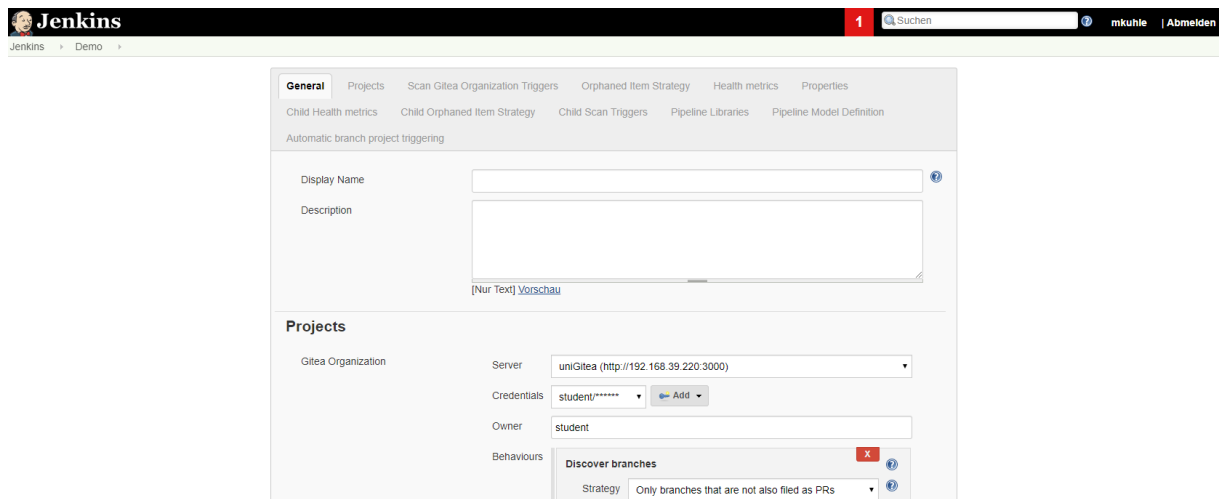
steps {
  sh 'echo "Run.."'
  sh '''
  echo "Zeile 1"
  echo "Zeile 2"
  '''
}
}
}
}
}

```

Danach Wechsel in den Jenkins

Element anlegen → Namen eingeben und GITEA Organization wählen mit OK bestätigen

Siehe Abbildung (unter ADD Credentials anlegen) Owner ebenfalls der Gitea-Account



Das Ergebnis sollte so aussehen

Pipeline master
Vollständiger Projektname: Demo/Demo/master

[Recent Changes](#)

Stage View

	Declarative: Checkout SCM	Build	Transfer	Run
Average stage times: (Average full run time: ~3s)	279ms	632ms	644ms	648ms
#2 Aug 26 17:36 1 commit	272ms	627ms	653ms	642ms
#1 Aug 26 16:42 No Changes	286ms	637ms	635ms	654ms

Build-Verlauf [Trend](#)

suchen

- #2 26.08.2019 15:36
- #1 26.08.2019 14:42

[RSS aller Builds](#) [RSS der Fehlschläge](#)

Permalinks

- [Letzter Build \(#2\) vor 4 Minuten 36 Sekunden](#)
- [Letzter stabiler Build \(#2\) vor 4 Minuten 36 Sekunden](#)
- [Letzter erfolgreicher Build \(#2\) vor 4 Minuten 36 Sekunden](#)
- [Neuester abgeschlossener Build \(#2\) vor 4 Minuten 36 Sekunden](#)