# Gitea

## Anleitung

https://docs.gitea.com/installation/install-with-docker

https://integrations.goauthentik.io/integrations/services/gitea/

https://docs.gitea.com/administration/config-cheat-sheet

Runner:

https://docs.gitea.com/usage/actions/act-runner#install-with-the-docker-image

https://docs.gitea.com/usage/actions/act-runner#configuration

## docker-compose.yml mit Runner

update mit v1.25.4 am 03.02.2026

```
services:
  server:
    image: docker.gitea.com/gitea:1.25.4
    container_name: gitea
    environment:
      - USER_UID=1000
      - USER_GID=1000
      - GITEA__database__DB_TYPE=postgres
      - GITEA__database__HOST=db:5432
      - GITEA__database__NAME=gitea
      - GITEA__database__USER=gitea
      - GITEA__database__PASSWD=<openssl rand -hex 24>
    restart: always
```

```
    networks:
      - gitea
    volumes:
      - app:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "3046:3000"
      - "222:22"
    depends_on:
      - db

  db:
    image: docker.io/library/postgres:14
    restart: unless-stopped
    environment:
      - POSTGRES_USER=gitea
      - POSTGRES_PASSWORD=<openssl rand -hex 24>
      - POSTGRES_DB=gitea
    networks:
      - gitea
    volumes:
      - db:/var/lib/postgresql/data

  runner:
    image: docker.io/gitea/act_runner:0.2.13
    environment:
      CONFIG_FILE: /config.yaml
      GITEA_INSTANCE_URL: "https://git.MEINEDOMAIN.DE"
      GITEA_RUNNER_REGISTRATION_TOKEN: "r5ES....ptLTD" # Token aus gitea web-ui Einstellungen
> Actions > Runner > create Runner
      GITEA_RUNNER_NAME: "gitea-runner-01"
      GITEA_RUNNER_LABELS: "ubuntu-22.04:host"
    volumes:
      - /mnt/fn-volume-01/docker-data/volumes/gitea_runner-cfg/config.yaml:/config.yaml
      - runner-data:/data
      - /var/run/docker.sock:/var/run/docker.sock

networks:
```

```
  gitea:
#    external: false

volumes:
  app:
  db:
  runner-data:
```

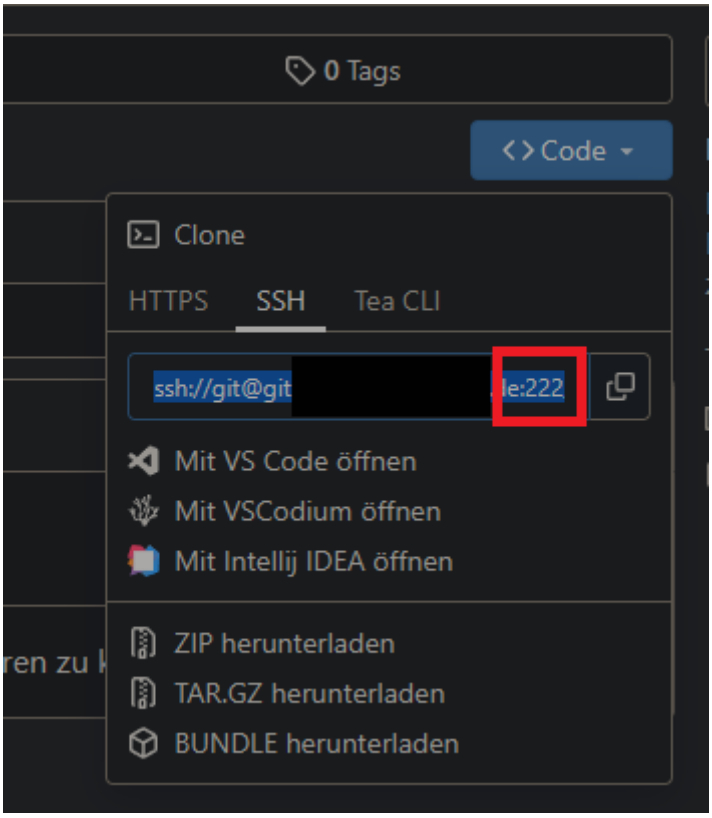
# SSH einrichten

Im meiner docker-compose.yml oben ist port 22 im Container auf Port 222 vom Host gemappt. Daher muss noch im Container in der app.ini der SSH-Port angepasst werden:

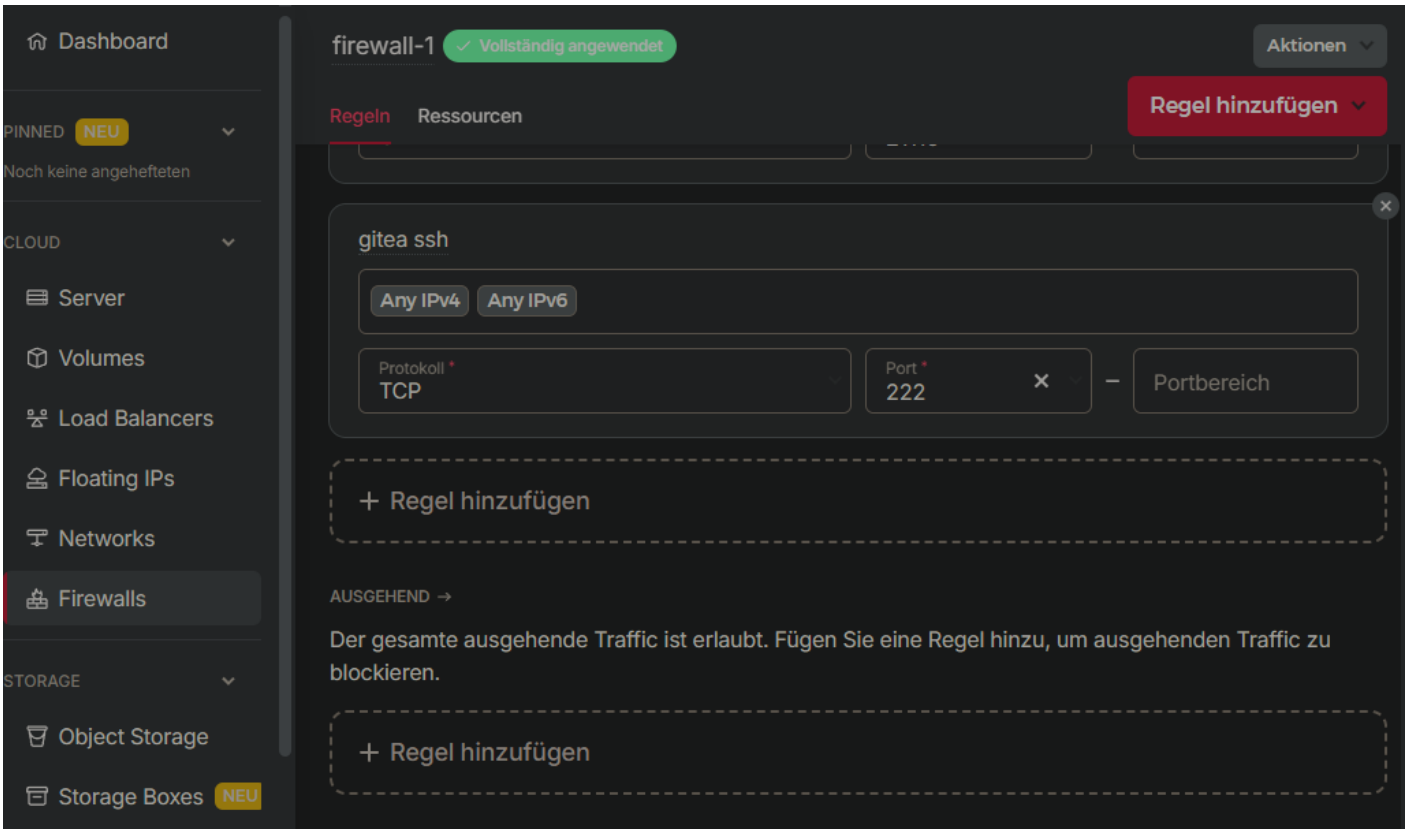
```
nano /var/lib/docker/volumes/gitea_app/_data/gitea/conf
```

```
[server]
...
SSH_PORT = 222
SSH_LISTEN_PORT = 222
...
```

Danach den Gitea-Container neustarten und in einem Repo unter Code > SSH die Adresse checken, ob der Port übernommen wurde:

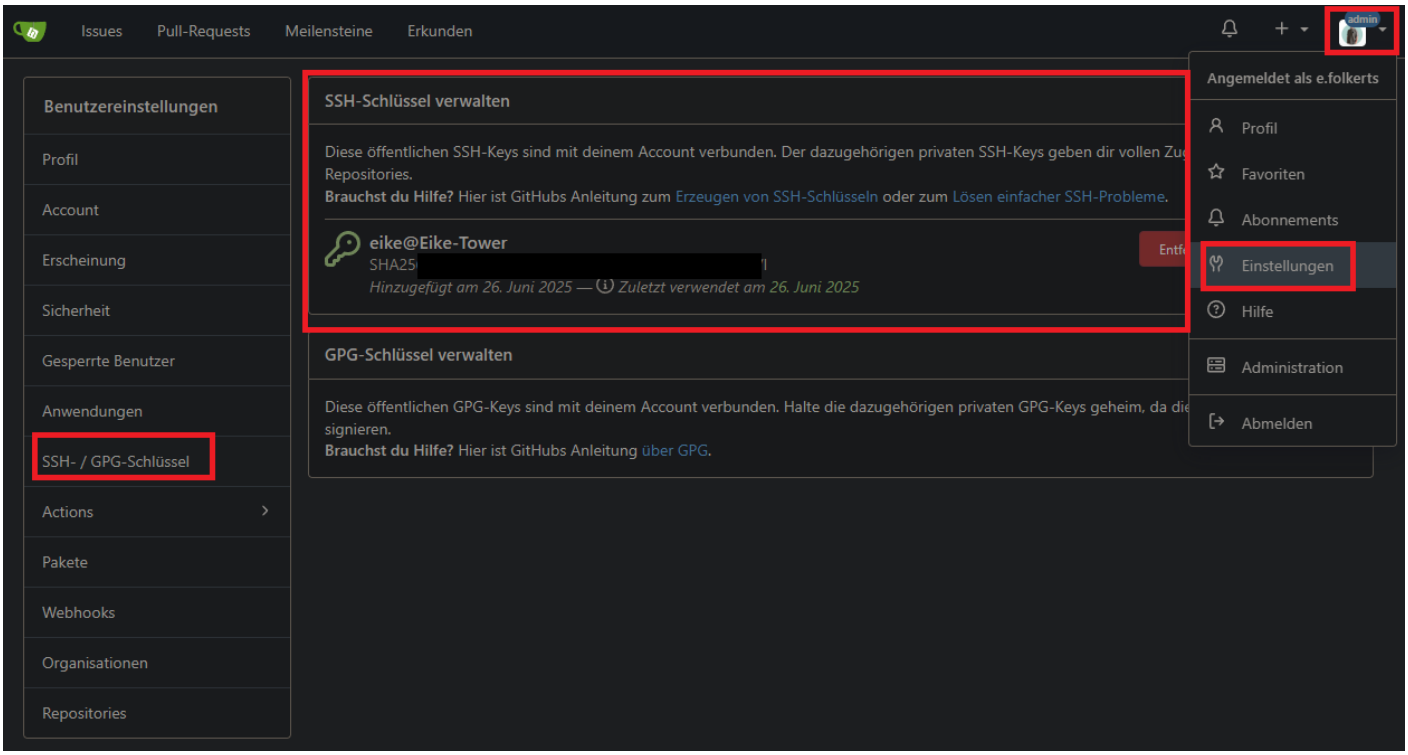



Nicht vergessen, den Port in der Firewall zu öffnen, in meinem Fall Hetzner, ansonsten auch in der ufw



Um den eigenen SSH-Key verwenden zu können, muss der Public-Key natürlich noch in Gitea eingepflegt werden unter

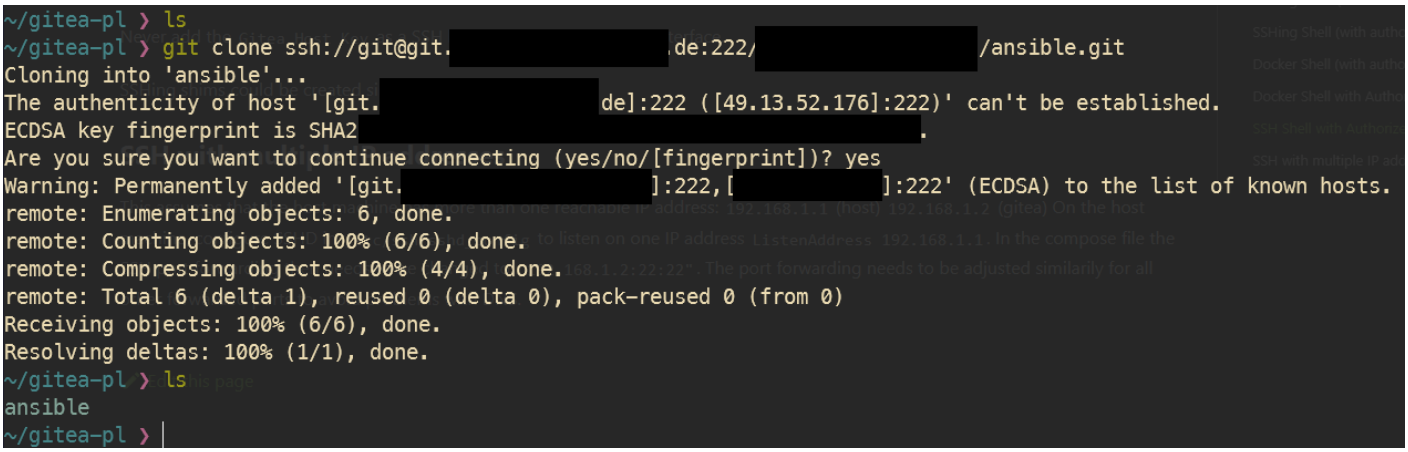
Profilbild > Einstellungen > SSH / GPG-Schlüssel > Schlüssel hinzufügen > .pub-Key einfügen und benennen




ssh testen mit

```
ssh -p 222 -i /home/BENUTZER/.ssh/id_MEINNAME_ed25519 git@git.DEINEDOMAIN.de
```

wenn successfully authenticated, testen mit git clone:




Falls ein anderer ssh-key angegeben werden muss, kann dies in der ~/.ssh/config eingestellt werden (Datei anlegen, falls nicht vorhanden):

```
~ ❯ cat /home/eike/.ssh/config

Host git.MEINEDOMAIN.de
  HostName git.MEINEDOMAIN.de
  Port 222
  User git
  IdentityFile ~/.ssh/id_MEINNAME_ed25519
```

# OIDC-Settings in Authentik

## app.ini Configuration

https://docs.gitea.com/administration/config-cheat-sheet

nano /var/lib/docker/volumes/gitea_app/_data/gitea/conf/app.ini:

```
[service]
# ...
DISABLE_REGISTRATION = false
REQUIRE_SIGNIN_VIEW = true
ALLOW_ONLY_EXTERNAL_REGISTRATION = true
ENABLE_AUTO_REGISTRATION = true
# ...

[openid]
ENABLE_OPENID_SIGNIN = true
ENABLE_OPENID_SIGNUP = true

# ...
```

## Anwendungen > Provider > gitea

# Aktualisieren Sie den OAuth2-Provider

**Name** * Provider for gitea

**Autorisierungs-Flow** * default-provider-authorization-implicit-consent (Authorize Application)

Flow der zur Authorisierung des Provider verwendet wird.

## Protokolleinstellungen

**Clienttyp** *

- (•) Vertraulich
  Vertrauliche Clients sind in der Lage, die Vertraulichkeit ihrer Anmeldedaten, beispielsweise des Client-Geheimnisses, zu wahren
- ( ) Öffentlich
  Öffentliche Clients können die Vertraulichkeit nicht gewährleisten und sollten Methoden wie PKCE nutzen.

**Client ID** * `UA3smB.................canzuz`

**Client Geheimnis** `BksUUt8M4myUtOF.....................................................FIMfXtHwkr`

**Redirect URIs/Origins (RegEx)** * Strict `https://git.MEINEDOMAIN.de/user/oauth2/authentik/callback` −

+ Add entry

Valid redirect URIs after a successful authorization flow. Also specify any origins here for Implicit flows.
Wenn keine expliziten Umleitungs-URIs angegeben sind, wird die erste erfolgreich verwendete Umleitungs-URI gespeichert.
To allow any redirect URI, set the mode to Regex and the value to ".*". Be aware of the possible security implications this can have.

**Signaturschlüssel** authentik Self-signed Certificate

Schlüssel zum Signieren der Token.

**Privater Schlüssel** Wählen Sie ein Objekt aus.

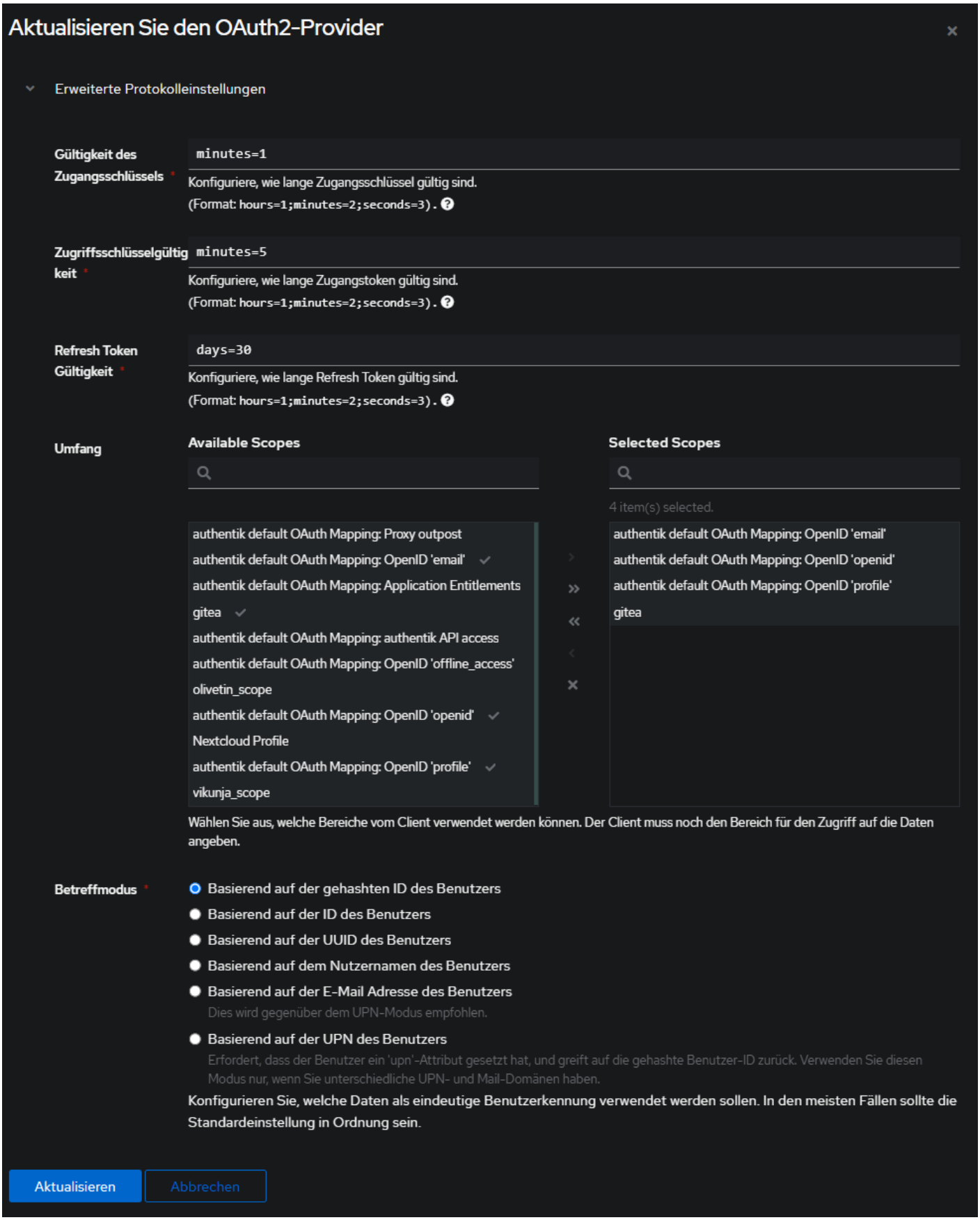
Schlüssel, der zum Verschlüsseln der Tokens benutzt wird.

## Advanced flow settings

## Erweiterte Protokolleinstellungen

## Machine-to-Machine-Authentifizierungseinstellungen

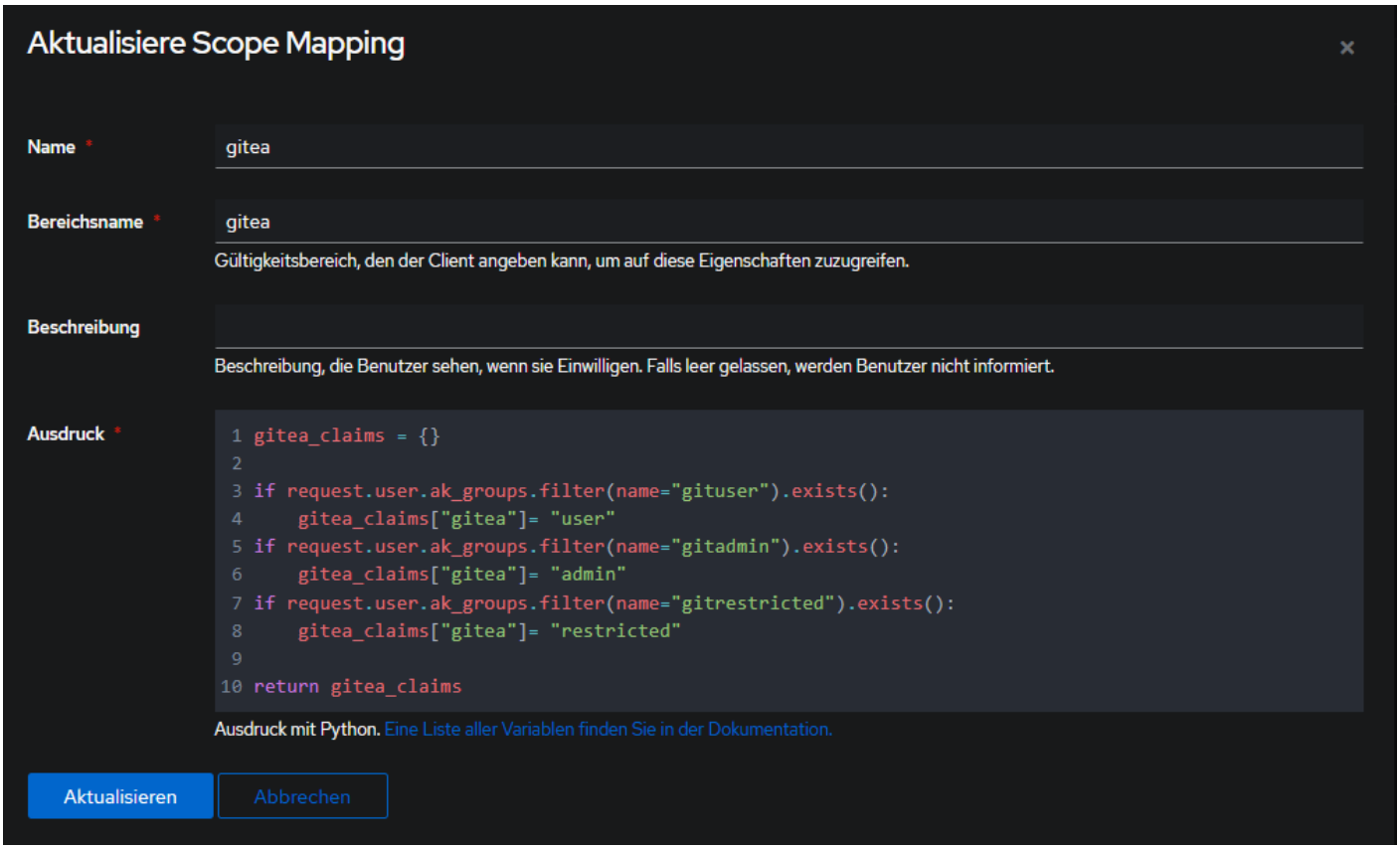
Aktualisieren | Abbrechen




Anwendungen > Anwendungen > Gitea

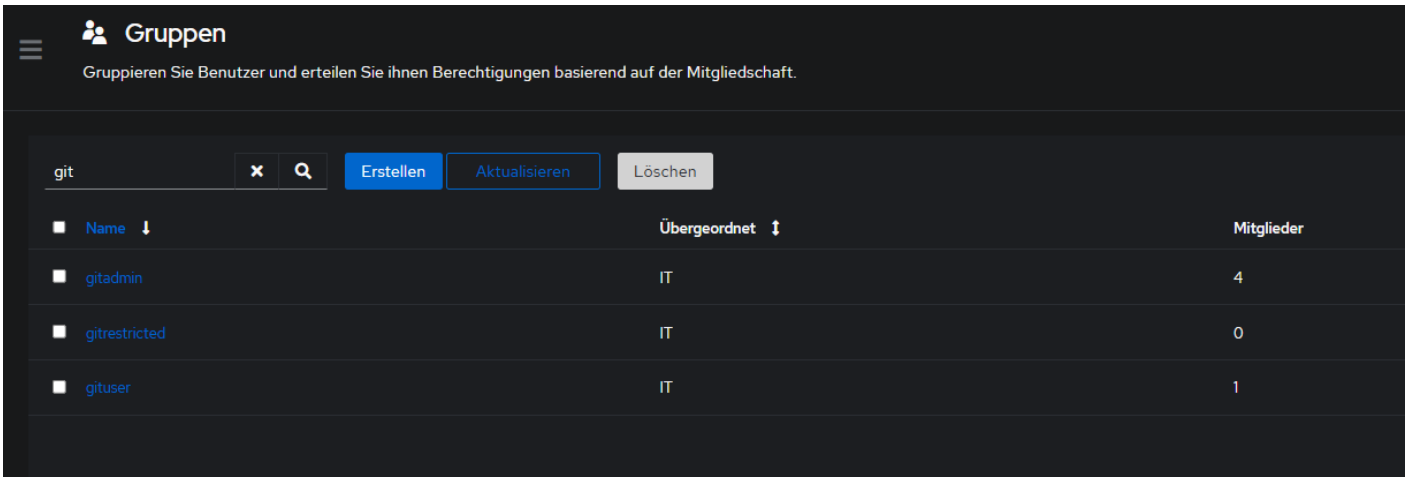

# Customization > Eigenschaften > Scope Mapping

gitea scope für gruppen gituser, gitadmin und gitrestricted



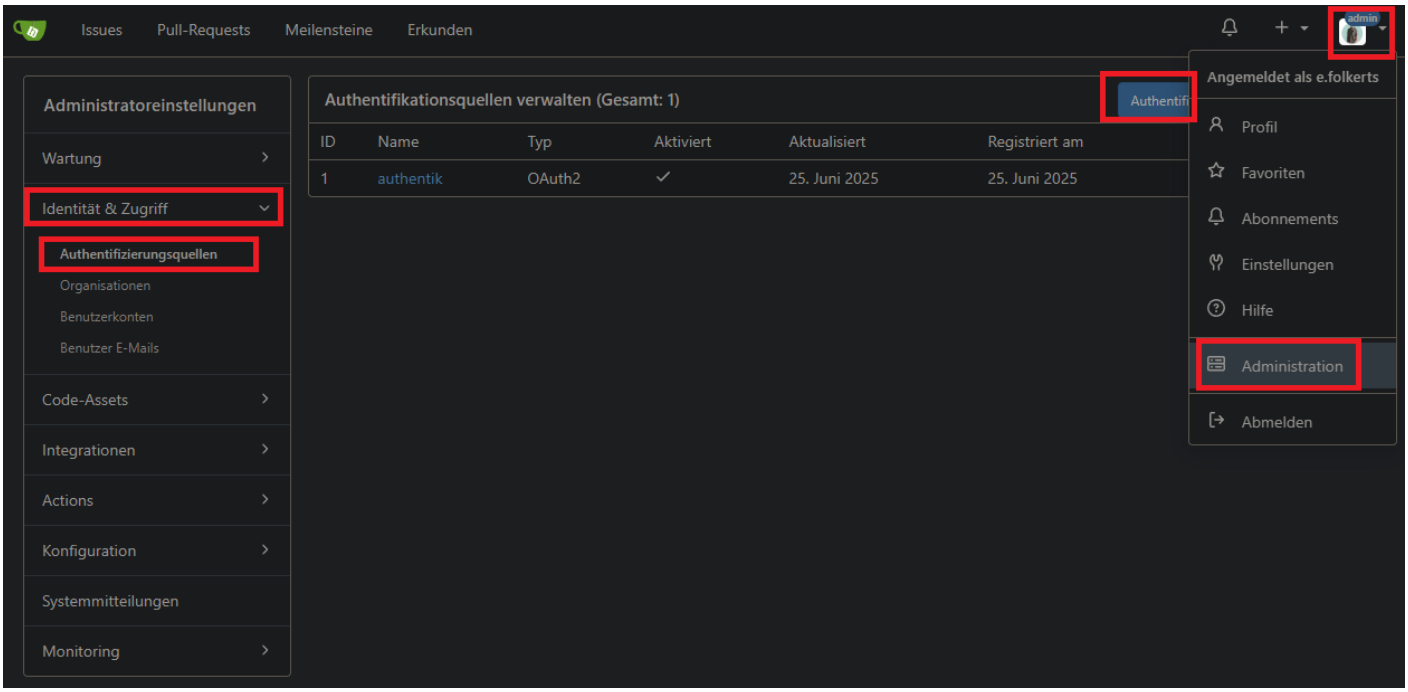

```
gitea_claims = {}

if request.user.ak_groups.filter(name="gituser").exists():
    gitea_claims["gitea"]= "user"
if request.user.ak_groups.filter(name="gitadmin").exists():
    gitea_claims["gitea"]= "admin"
if request.user.ak_groups.filter(name="gitrestricted").exists():
    gitea_claims["gitea"]= "restricted"

return gitea_claims
```

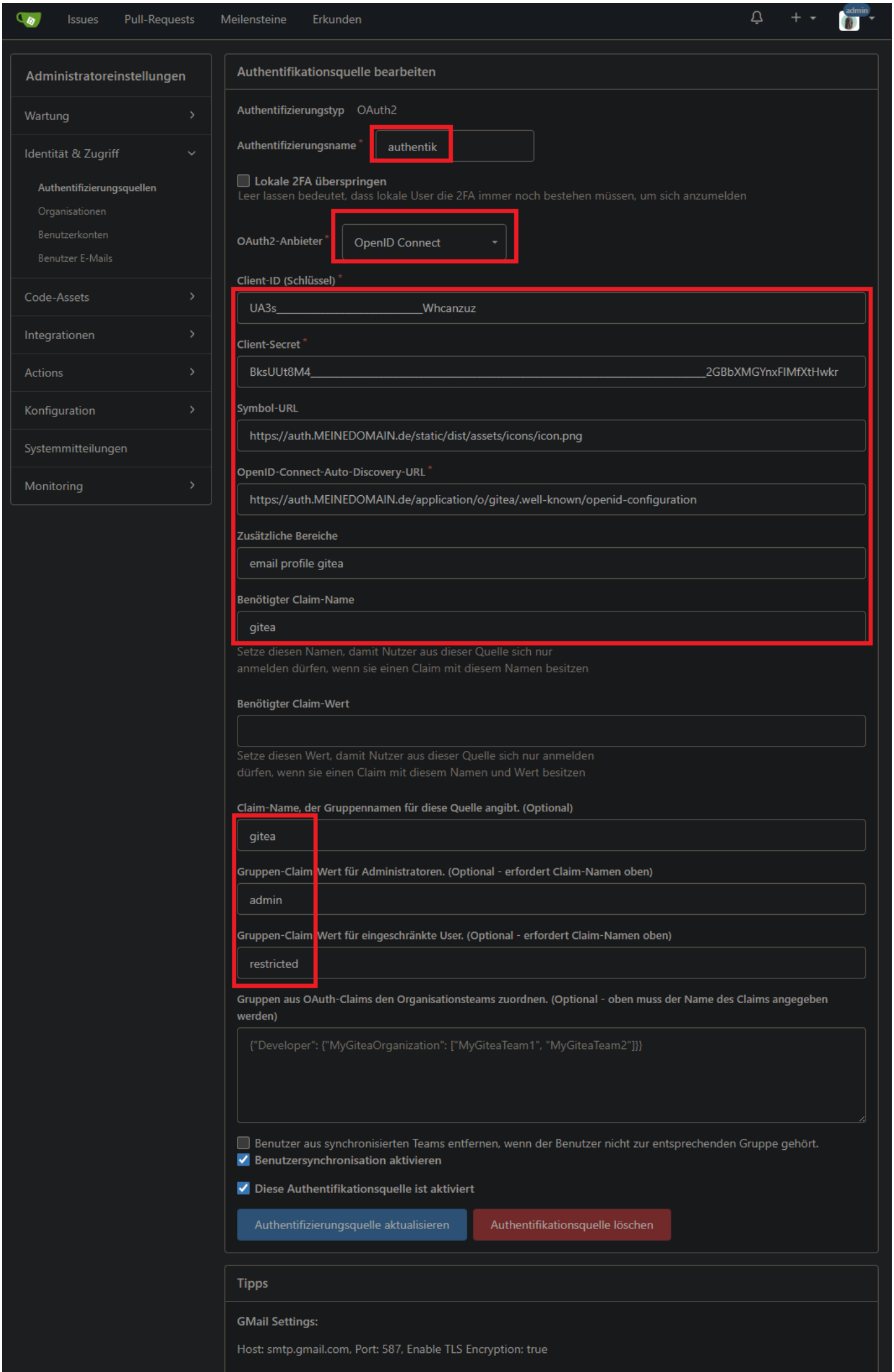
# Verzeichnis > Gruppen




# OIDC-Settings in Gitea

Administration > Identität & Zugriff > Authentifizierungsquellen > Neu:




Issues
Pull-Requests
Meilensteine
Erkunden
admin
Administratoreinstellungen
Wartung
Identität & Zugriff
Authentifizierungsquellen
Organisationen
Benutzerkonten
Benutzer E-Mails
Code-Assets
Integrationen
Actions
Konfiguration
Systemmitteilungen
Monitoring
Authentifikationsquelle bearbeiten
Authentifizierungstyp OAuth2
Authentifizierungsname *
authentik
Lokale 2FA überspringen
Leer lassen bedeutet, dass lokale User die 2FA immer noch bestehen müssen, um sich anzumelden
OAuth2-Anbieter *
OpenID Connect
Client-ID (Schlüssel) *
UA3s_____Whcanzuz
Client-Secret *
BksUUt8M4_____2GBbXMGYnxFIMfXtHwkr
Symbol-URL
https://auth.MEINEDOMAIN.de/static/dist/assets/icons/icon.png
OpenID-Connect-Auto-Discovery-URL *
https://auth.MEINEDOMAIN.de/application/o/gitea/.well-known/openid-configuration
Zusätzliche Bereiche
email profile gitea
Benötigter Claim-Name
gitea
Setze diesen Namen, damit Nutzer aus dieser Quelle sich nur
anmelden dürfen, wenn sie einen Claim mit diesem Namen besitzen
Benötigter Claim-Wert
Setze diesen Wert, damit Nutzer aus dieser Quelle sich nur anmelden
dürfen, wenn sie einen Claim mit diesem Namen und Wert besitzen
Claim-Name, der Gruppennamen für diese Quelle angibt. (Optional)
gitea
Gruppen-Claim-Wert für Administratoren. (Optional - erfordert Claim-Namen oben)
admin
Gruppen-Claim-Wert für eingeschränkte User. (Optional - erfordert Claim-Namen oben)
restricted
Gruppen aus OAuth-Claims den Organisationsteams zuordnen. (Optional - oben muss der Name des Claims angegeben werden)
{"Developer": {"MyGiteaOrganization": ["MyGiteaTeam1", "MyGiteaTeam2"]}}
Benutzer aus synchronisierten Teams entfernen, wenn der Benutzer nicht zur entsprechenden Gruppe gehört.
Benutzersynchronisation aktivieren
Diese Authentifikationsquelle ist aktiviert
Authentifizierungsquelle aktualisieren
Authentifikationsquelle löschen
Tipps
GMail Settings:
Host: smtp.gmail.com, Port: 587, Enable TLS Encryption: true

> groß- und kleinbuchstaben beachten, alles so nennen wie in anleitung (auch scope, gruppen, claims, slug bei provider usw)

---

Revision #14
Created 2025-06-26 09:17:39 UTC
Updated 2026-02-05 10:33:08 UTC