

Inhaltsverzeichnis

Gitlab	3
<i>Dateien aus privatem Repository per Access Token herunterladen</i>	4

Gitlab

Gitlab als Docker container

```
mkdir -p ~/gitlab/dockerdata
```

[docker_run.sh](#)

```
GITLAB_HOME=~/gitlab/dockerdata
docker run --detach \
  --hostname myhostname \
  --publish 443:443 --publish 8090:80 --publish 2222:22 \
  --name gitlab \
  --restart always \
  --volume $GITLAB_HOME/config:/etc/gitlab \
  --volume $GITLAB_HOME/logs:/var/log/gitlab \
  --volume $GITLAB_HOME/data:/var/opt/gitlab \
  gitlab/gitlab-ee:latest
```

wenn Gitlab hinter einem Proxy und/oder einem relativen URL (Unterverzeichnis) laufen soll, muß man den obigen Shell- Aufruf noch um einen Parameter ergänzen ¹⁾:

```
--env 'GITLAB_RELATIVE_URL_ROOT=/gitlab' \
```

admin password anpassen: (<https://stackoverflow.com/a/55747403>)

```
docker ps -a

docker exec -it gitlab bash <-- with your CONTAINER_ID

$ gitlab-rails console -e production

user = User.where(id: 1).first
user.password = 'your secret'
user.password_confirmation = 'your secret'
user.save
exit
```

URL anpassen:

For GitLab to display correct repository clone links to your users, it needs to know the URL under which it is reached by your users, e.g. <http://gitlab.example.com>. Add or edit the following line in `/etc/gitlab/gitlab.rb`:

```
external_url "http://gitlab.example.com"
```

for the change to take effect, run:

```
sudo gitlab-ctl reconfigure
```

Backup ziehen:

```
docker exec -t <container name> gitlab-backup create
```

Backup wieder einspielen:

```
# Stop the processes that are connected to the database
docker exec -it <name of container> gitlab-ctl stop puma
docker exec -it <name of container> gitlab-ctl stop sidekiq

# Verify that the processes are all down before continuing
docker exec -it <name of container> gitlab-ctl status

# Run the restore - filename WITHOUT THE TRAILING TAR EXTENSION STUFF !!
docker exec -it <name of container> gitlab-backup restore
BACKUP=11493107454_2018_04_25_10.6.4-ce

# Restart the GitLab container
docker restart <name of container>

# Check GitLab
docker exec -it <name of container> gitlab-rake gitlab:check SANITIZE=true
```

Dateien aus privatem Repository per Access Token herunterladen

Man braucht nur ein Token <token> mit den Rechten `api`, `read_user` und `read_repository`

Dann braucht man entweder die numerische Projekt-ID <id> oder äquivalent der Namespace - und Projekt- Pfad, **aber URL encoded!**, aus / wird z.B. `%2F`.

Das gleiche gilt für den Datei- Pfad innerhalb des Repository <file_path>, auch er muß URL encoded sein.

Will man nicht auf die aktuelle Datei, sondern einen bestimmten Stand verweisen, braucht man als <ref> die Commit-ID oder, bei einem anderen Branch, den Branch- Namen.

Dann bastelt man man den URL nach folgendem Schema ²⁾

```
GET /projects/<id>/repository/files/<file_path>/raw[?ref=<ref>]
```

Dann braucht man noch einen URL- Downloader, der den HTTPS-Header mit befüllen kann

```
curl --header "PRIVATE-TOKEN: <token>"
"https://gitlab.com/api/v4/projects/<id>/repository/files/<file_path>/raw?re
```

```
f=<ref>"
```

Upps: Man kann das Token auch direkt an den Link hängen

```
curl  
"https://gitlab.com/api/v4/projects/<id>/repository/files/<file_path>/raw?re  
f=<ref>&private_token=<token>"
```

und dann geht das



1)

<https://github.com/sameersbn/docker-gitlab#deploy-to-a-subdirectory-relative-url-root>

2)

https://docs.gitlab.com/ee/api/repository_files.html#get-raw-file-from-repository

From:

<http://www.koehlers.de/wiki/> - **Steffen Köhlers Online- Bastelbuch**

Permanent link:

<http://www.koehlers.de/wiki/doku.php?id=pc:gitlab>

Last update: **2021/12/19 11:40**

