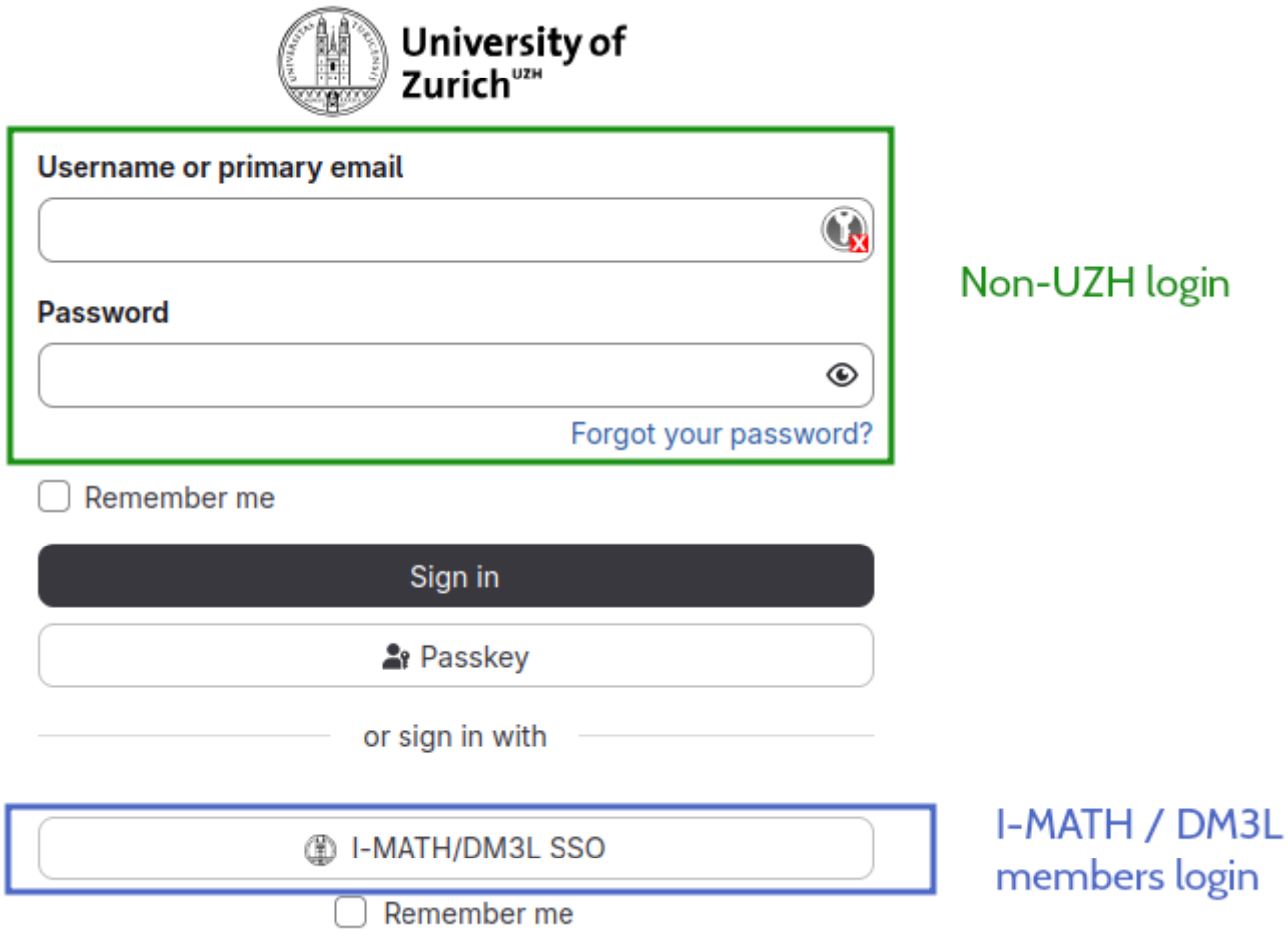
Git

Server Info	https://git.math.uzh.ch
Software	Gitlab
Git push/pull access	via SSH over `git@git.math.uzh.ch` *Do not forget to upload your public ssh key!*
Cheat sheet	GIT for SVN users
Beginners Guide	Einsteiger-Guide-Wegweiser-durch-Git-GitHub-und-GitLab

Prerequisites

- Git installed (already done on IMATH server).

Git account



Members of UZH

- The GIT account is created automatically on your first access.
- If you already have an I-MATH/DM3L account, please sign in with `I-MATH/DM3L SSO` button - you will be redirected to https://login.math.uzh.ch/ unless you have already an active session.
- Otherwise, create first an I-MATH account on https://www.math.uzh.ch/account.

Non UZH persons

If you like to collaborate on a project which is hosted on https://git.math.uzh.ch, you have to request a Git account.

- Best is to ask a project member affiliated with I-MATH to request the account via https://math.uzh.ch/support.
- Once the request is accepted, you will receive an email for validation and first-time login.
- Sign in with the username/password form

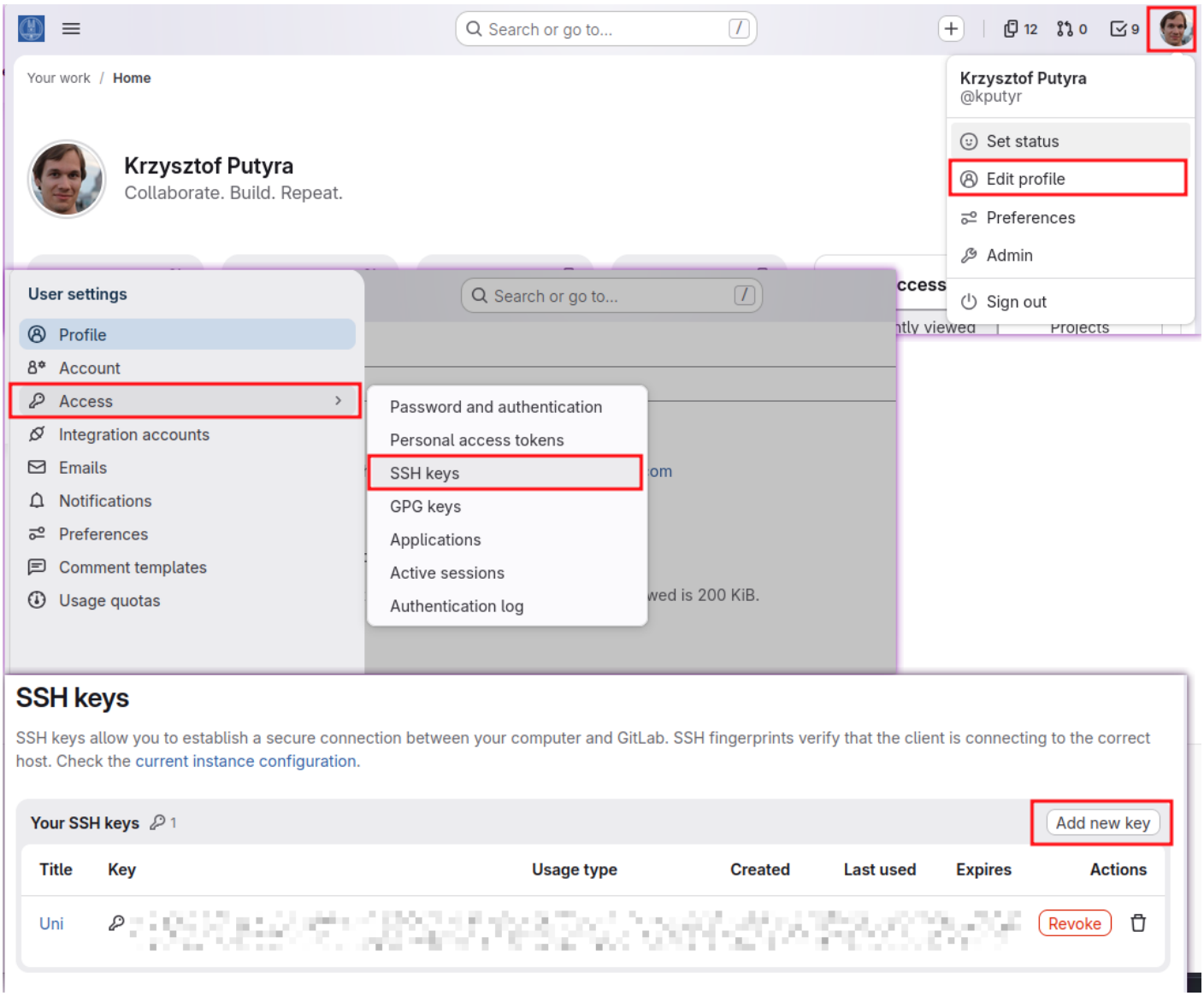
- After the first login you will be asked to change the initial password.

Accessing with SSH keys

We recommend to provide your public SSH key - this streamlines working with Git repos and committing your changes. Do this on all devices which you like to use with the GIT server (and in your I-MATH ThinLinc session if you use it).

- GitLab documenation on setting up SSH access
- I-MATH wiki: creating SSH keys

Your public SSH key can be uploaded easily to GitLab on the web interface



Testing the setup

- With ssh command: `ssh -T git@git.math.uzh.ch`
  Do **not** replace `git` with your username, use **git@git.math.uzh.ch** as written!
- By cloning any Git repo.

Accessing with HTTPS

Authenticating git commands with username/password does not work when MFA is configuration. Instead, you have to use a personal access token - see the GitLab documentation on how to create one.

Working with Git

```
git COMMAND git@git.math.uzh.ch/<username>/<projectname>
```

Managing your Projects with Gitlab

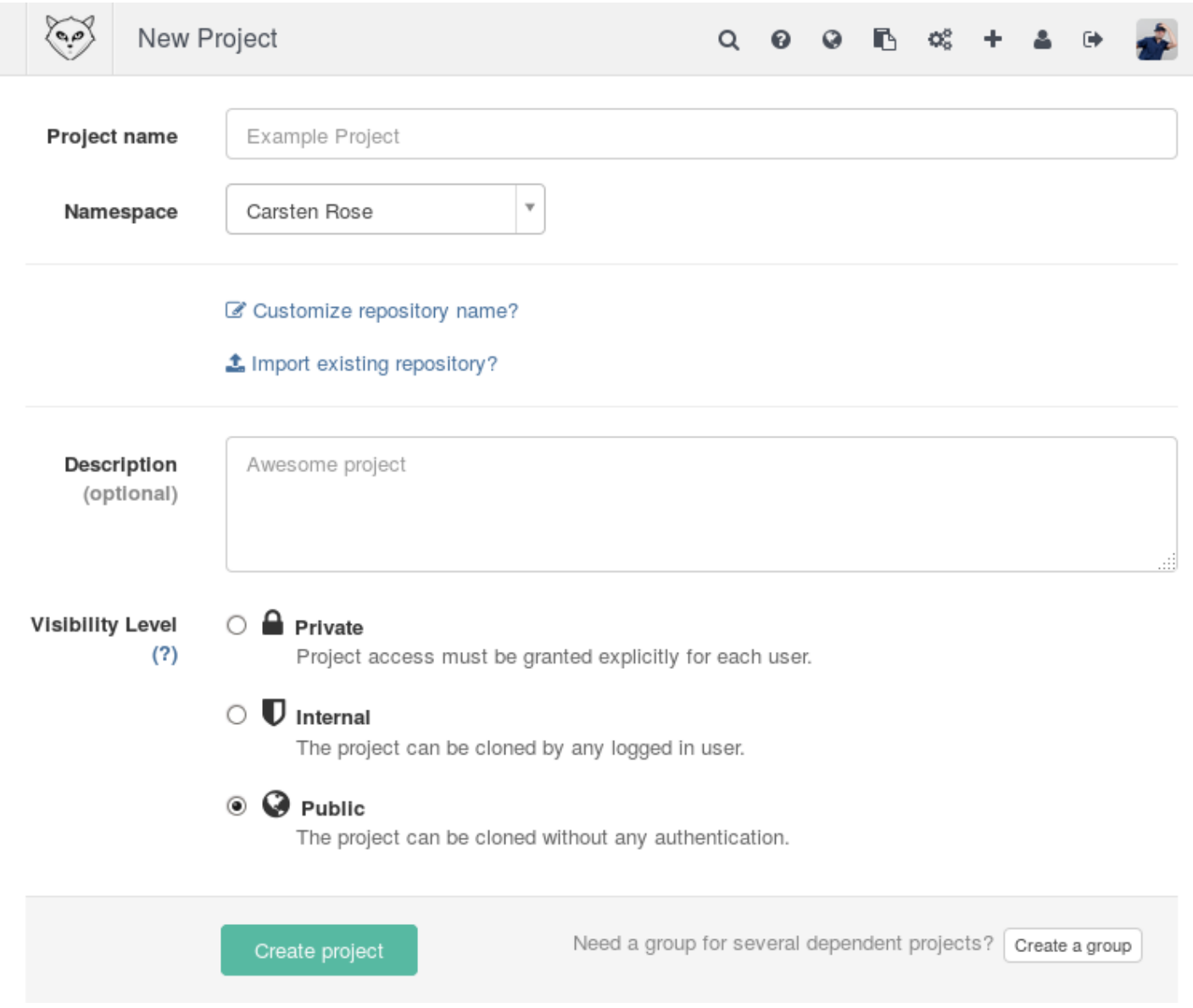
Gitlab has quite extensive managing capabilities.

- You can, among other things,
  - Crate/fork/rename/remove projects
  - Create/delete/join Teams
  - Assign members with roles to projects
  - Create/Assign issues
  - Have a wiki for a project
- The gitlab user documentation is found on `https://git.math.uzh.ch/help`.

GIT Best Practice

New project from scratch

- Create a new project via the webinterface.



- You'll see a summary with the next steps:

```
Git global setup
----------------
git config --global user.name "First Lastname"
git config --global user.email "email@math.uzh.ch"

Create a new repository
-----------------------
mkdir <local projektdir>
cd <local projektdir>
git init
touch README.md
git add README.md
git commit -m "first commit"
```

```
git remote add origin git@git.math.uzh.ch:<project>.git
git push -u origin master

Push an existing Git repository
------------------------------
cd existing_git_repo
git remote add origin git@git.math.uzh.ch:<project>.git
git push -u origin master
```

GIT daily use

Local: Add new files to existing repo

```
git add <file>
```

Local: Work on your files

- Commit changes locally

```
git commit -a
```

- Or at IMATH you can use `git-commit`, which will in every file replace `$Id$` by User/timestamp:

```
git-commit
```

- Commit changes remote

```
git push
```

Local: clone an existing repo to start working on it

```
git clone git@git.math.uzh.ch:<project>.git
```

Purged Remote branches still shown locally

- See: http://stackoverflow.com/a/17320081/1167442

```
$ git remote update --prune
```

Purge a file completely from all commits & branches

- This is not how GIT should be used!
- ONLY, ONLY, ONLY do this if it is absolutly necessary!
- There might be side effects, you're on your own!
- Source: https://github.com/nachoparker/git-forget-blob

```
$ cd <root of your repo>
$ git-forget-file.sh <path to file>/<file>
```

Overwrite local branch with remote

- https://phoenixnap.com/kb/git-overwrite-local-branch-with-remote

Visually show commit history

- `gitg`
- Use the git.math.uzh.ch webinterface

Search the commit which breaks something

- Based on binary search.
- Give a good and a bad commit.
- The `git bisect ...` command always looks for the commit in the middle. Do a check of such a commit and decide to continue the search in the lower or upper half.
- https://git-scm.com/docs/git-bisect

Mac OS X

git: xcrun missing

- By using `git` the following error is shown:

```
xcrun: error: invalid active developer path (/Library/Developer/CommandLineTools),
missing xcrun at: /Library/Developer/CommandLineTools/usr/bin/xcrun
```

- Open a terminal and install `xcode`: a pop up have to acknowledged.

```
xcode-select --install
```

Umlaute

- MacOS X uses a different UTF8 encoding for filenames than MS-Windows and Linux.
- This leads to confusion. E.g.: after a fresh clone of a GIT repo (created on MS-Windows or linux) a `git status` will show untracked files if they contain umlaut in their filenames, even without any change has been done!
- https://www.git-tower.com/help/guides/faq-and-tips/faq/unicode-filenames/mac
- To fix this, one option is to change the handling of filenames in the local git configuration:

```
$ git config --global core.precomposeunicode false
```

- After that, clone the repository again.

GIT Client

- https://www.sourcetreeapp.com/?ref=catalins.tech

FAQ

Change Repo: http > ssh

```
# Status Quo
$ git remote -v show
origin□https://systemvcs.math.uzh.ch/it/typo3-docker.git (fetch)
origin□https://systemvcs.math.uzh.ch/it/typo3-docker.git (push)
```

```
# Remove Origin
$ git remote rm origin

# Set SSH Version
$ git remote add origin git@systemvcs.math.uzh.ch:it/typo3-docker.git



# Status Quo
git remote -v show
origin	git@systemvcs.math.uzh.ch:it/typo3-docker.git (fetch)
origin	git@systemvcs.math.uzh.ch:it/typo3-docker.git (push)

# Try a git pull:
$ git pull

# If you receive: There is no tracking information for the current branch. ... try:
$ git branch --set-upstream-to=origin/main main
```

Attention: if you change to SSH and

- You use SSH public key (stored on GIT server),
- You're logged in on a remote server by ssh and you're using a different user ID than the one where the public key on the git server belongs to,

than you should forward your SSH-Agent! E.g. **ssh root@host -A**

Pipeline failed notification

- Maintainer receives all pipeline errors per mail by default

✖ Your pipeline has failed.

Project	Benjamin Baer / pages-hexo
Branch	master
Commit	756dfc9d adding ci and gitignore
Commit Author	Benjamin Baer

Pipeline #701 triggered by Benjamin Baer

had 1 failed build.

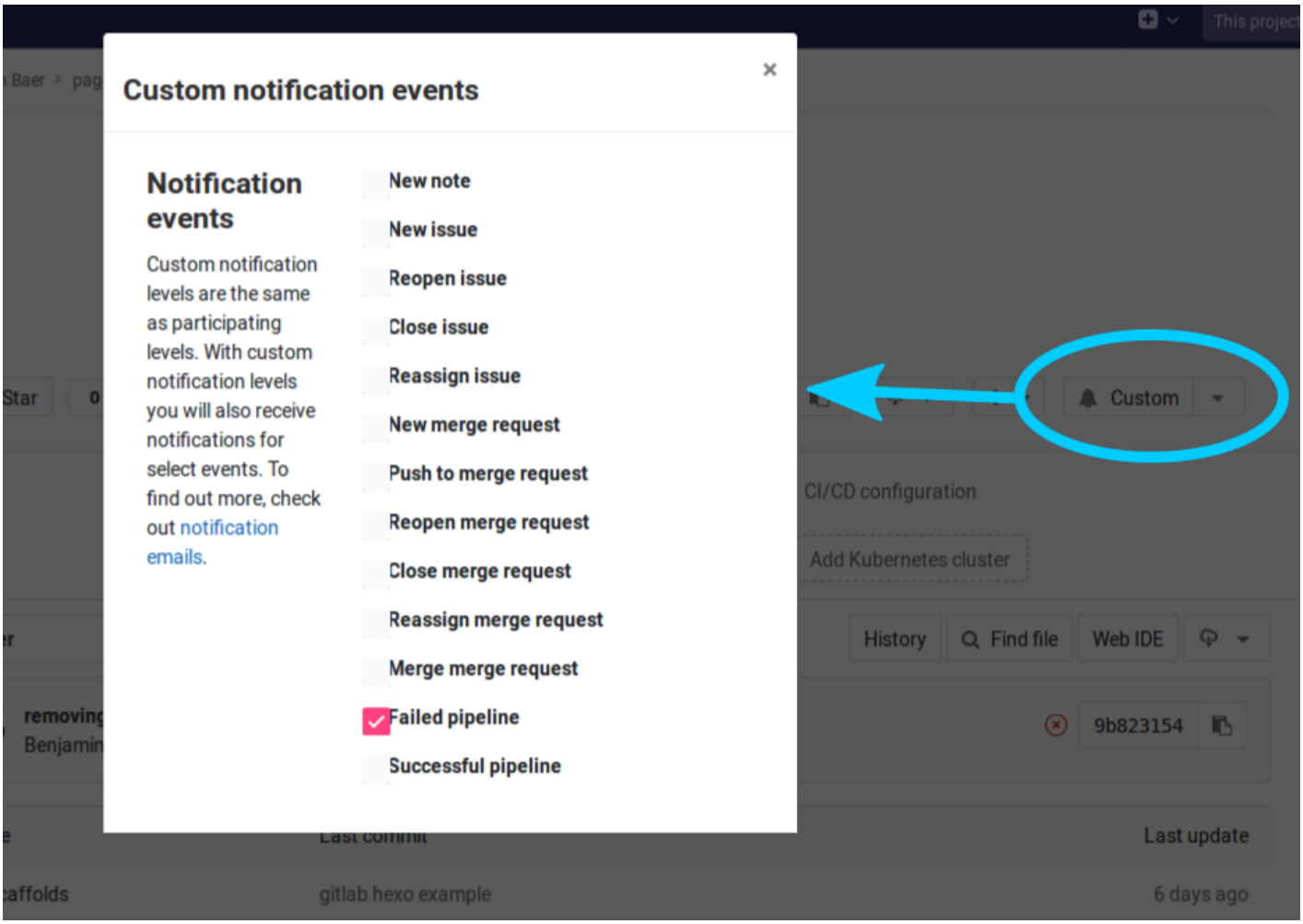
Logs may contain sensitive data. Please consider before forwarding this email.

✖ test pages

```
npm ERR!   code: 'EACCES',
npm ERR!   syscall: 'access',
npm ERR!   path: '/usr/seppl.5.1/drwho/pack-a/nodejs-6.10.1-ro/amd
64-linux-ubuntu16.04/lib/node_modules' }
npm ERR!
npm ERR! Please try running this command again as root/Administrat
or.

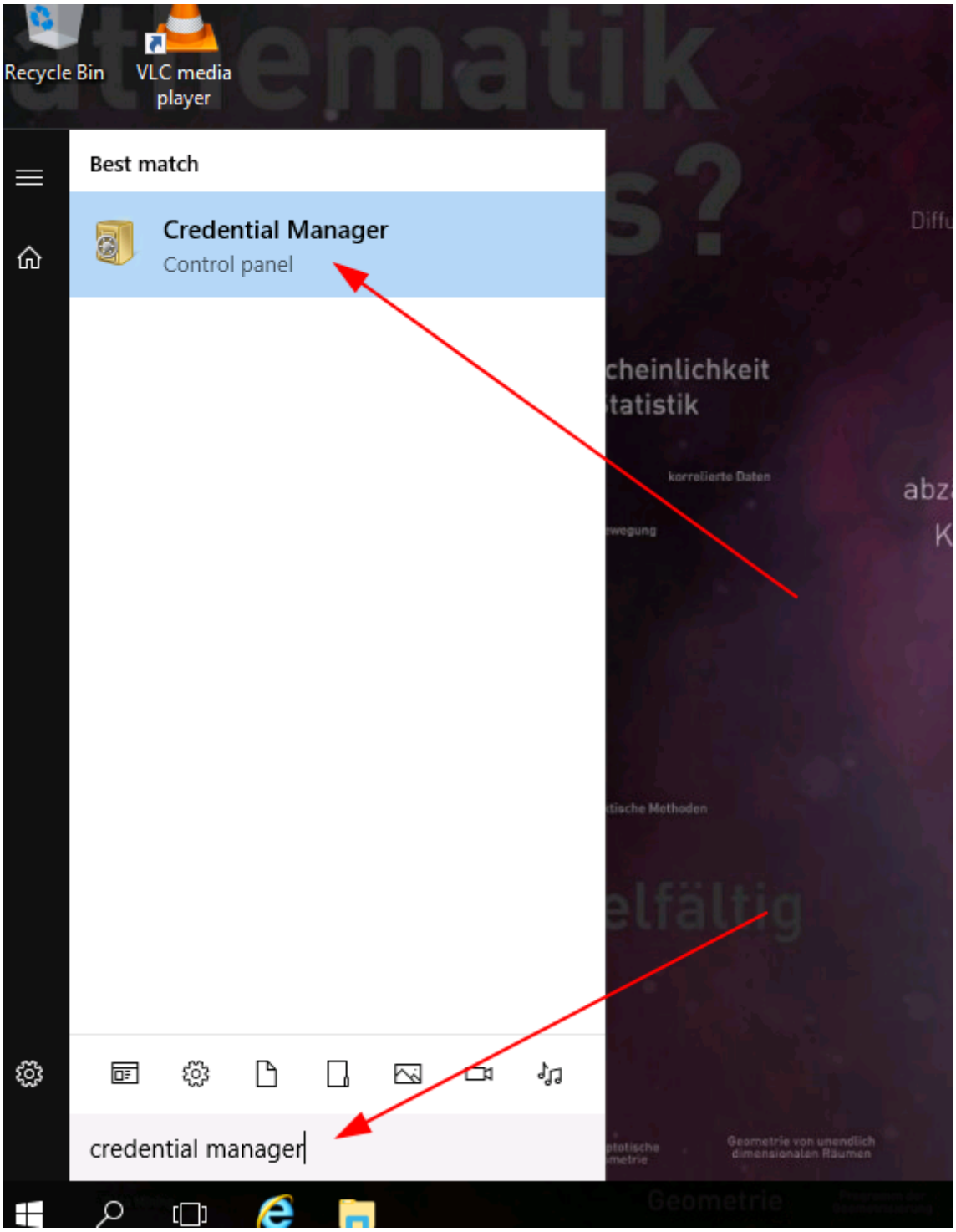
npm ERR! Please include the following file with any support reques
t:
npm ERR!     /home/gitlab-runner/builds/7e3e234d/0/bbaer/pages-hex
o/npm-debug.log
ERROR: Job failed: exit status 1
```

- Others can subscribe to it on the main site of a project:

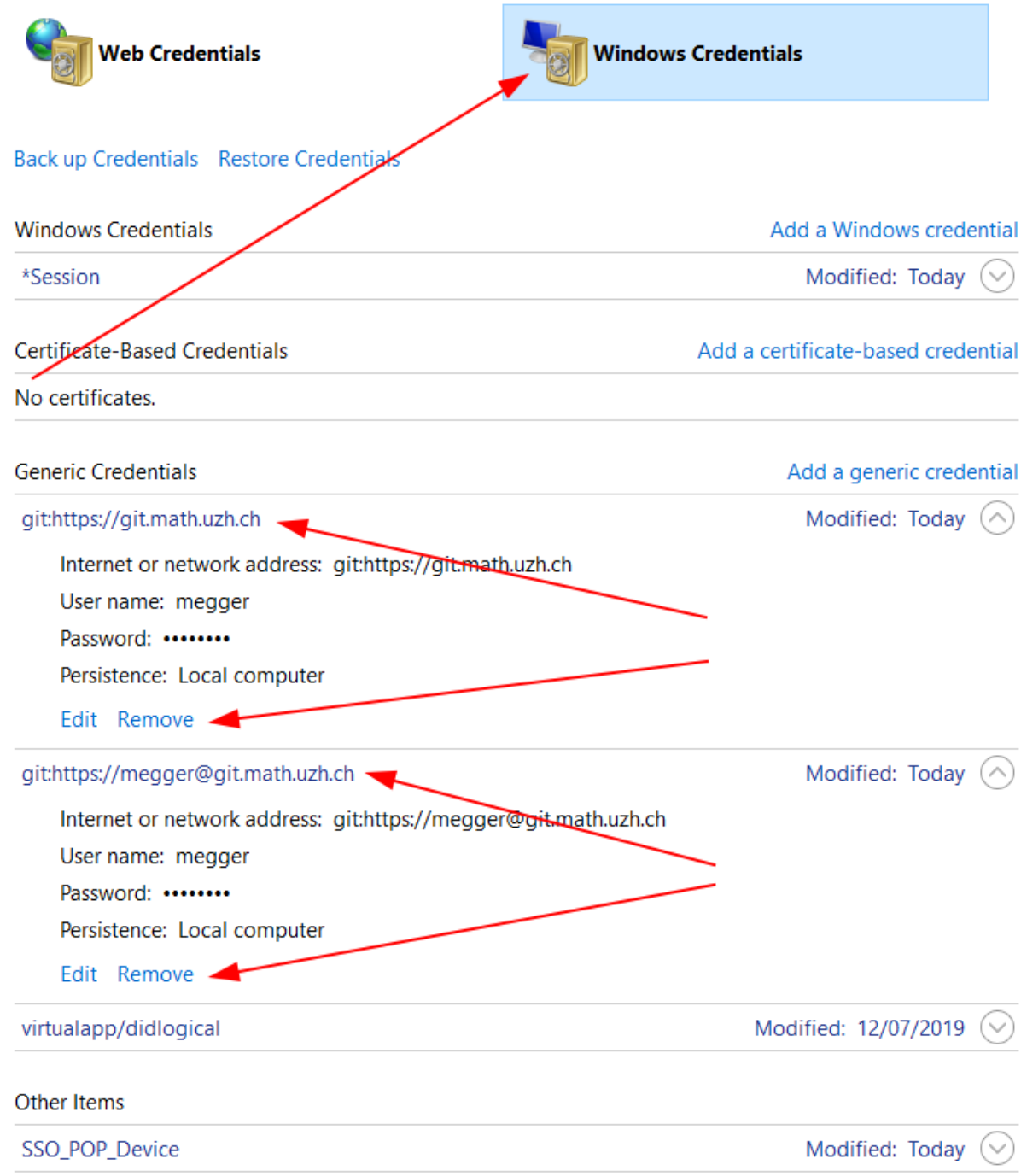


Windows Change Saved Git Password

- Open "Credential Manager" (German: "Anmeldeinformationsverwaltung")



- Choose "Windows Credentials" (German: "Windows-Anmeldeinformationen") and remove all saved credentials for Git.



- Use git as usual, you will be asked to enter your password

Revision #7
Created 2026-06-30 10:50:46 CEST by admino
Updated 2026-09-08 22:38:54 CEST by Krzysztof Putyra