



Tutorial 1:

Set up your Environment

UZH VPN:

You need the UZH VPN to access the Server where your Webpage is deployed:

How to configure the UZH-VPN: [Step-by-Step-Instructions](#)

Already have VPN but need to [change Shared Secret](#)

Git-Lab: Workspace


You should have already received an invitation to this group by e-mail:

<https://gitlab.uzh.ch/ltwa-hs23>

Set up Your Workspace:

- > Fork the “Example Project” (on page 3)
- > Select your personal namespace in the Group
- > clone the forked project using the SSH version.
- > Terminal: git clone <insert URL>

ltwa-hs23 > Example project > Fork project

**Fork project**

A fork is a copy of a project. Forking a repository allows you to make changes without affecting the original project.


Project name

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

Project URL

Want to organize several projects? [Create a group](#)

Project description (optional)

Visibility level 

☒ **Private**
Project access must be granted to members of the group.

☐ **Internal**
The project can be accessed by members of the group.

☐ **Public**
The project can be accessed without any authentication.

Namespaces

ltwa-hs23/christianmatthias.fuerst

ltwa-hs23/zainab.affab

ltwa-hs23/yining.wang

ltwa-hs23/yasarkemal.demirelli

ltwa-hs23/yahui.li

ltwa-hs23/yahui.li

Project slug

This project is part of a group, access will be granted to members of the group. [Create a group](#)

If no SSH, set a new key in GitLab under ‘Preferences/SSH keys’

Here’s a Guid to generate a new key:

<https://spectralops.io/blog/guide-to-ssh-keys-in-gitlab/>

Deployment

- In order to make your website accessible to the world you need to deploy it to a webserver.
- A webserver is a computer that is constantly running.
- For each student enrolled in this course there is a server, that is accessible within the UZH network (from the wifi within UZH or through the UZH VPN).

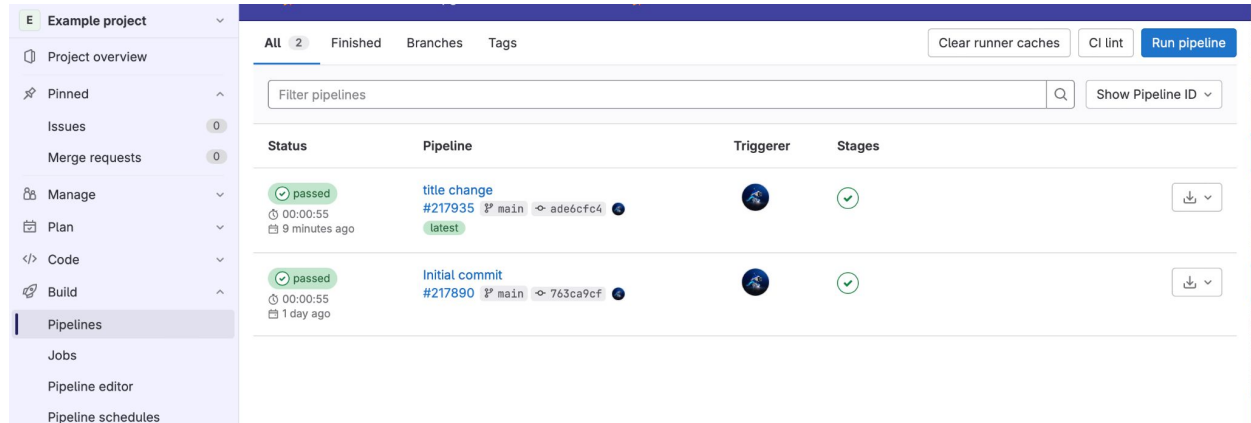
Deployment

- The **.gitlab-ci.yml** file in your repo tells GitLab to deploy your website to your designated server everytime you push to the master branch

Name	Last commit
static	Initial commit
templates	title change
 .gitignore	Initial commit
 .gitlab-ci.yml	Initial commit

Deployment

- You can run deploy manually: “Run pipeline” (can take a few minutes), after the first time you will receive an E-mail with the URL to your webpage and some additional informations to connect to your database. You can add the URL to the webpage to your READ ME for easy access.
- Changes pushed to your Repository get automatically deployed to the server.



The screenshot displays the GitHub Actions interface for a project named "Example project". The left sidebar contains a navigation menu with options: Project overview, Pinned, Issues (0), Merge requests (0), Manage, Plan, Code, Build, Pipelines (selected), Jobs, Pipeline editor, and Pipeline schedules. The main content area shows a list of pipelines under the "All" tab, with 2 pipelines listed. The table has columns for Status, Pipeline, Triggerer, and Stages. The first pipeline is titled "title change" (#217935) on the main branch, triggered by a commit (ade6cfc4), and is in a "passed" state. The second pipeline is titled "Initial commit" (#217890) on the main branch, triggered by a commit (763ca9cf), and is also in a "passed" state. Both pipelines show a duration of 00:00:55. At the top right of the pipeline list, there are buttons for "Clear runner caches", "CI lint", and "Run pipeline". A search bar for "Filter pipelines" and a "Show Pipeline ID" dropdown are also present.

Status	Pipeline	Triggerer	Stages
passed 00:00:55 9 minutes ago	title change #217935 main ade6cfc4 latest		
passed 00:00:55 1 day ago	Initial commit #217890 main 763ca9cf		