Week 01: Introduction to Seminars

Welcome to the course!

Agenda

- Tutor introduction
- Checking Docker & WSL
- Course info
- Git Basics
- Gitflow
- Setup Gitlab & IDE

Who (am/are) (I/we)?

Let me (us) introduce myself (ourselves)

Course info: Basics

- Seminars are compulsory (max. 4 unexcused seminars)
- Source of truth: Syllabus and Gitlab
- Communication: Discord
  - Help support almost 24/7
  - We ❤ to help you

Web development and markup languages

We will focus on different technologies one must master to become a full-stack developer.

Frontend: JavaScript-based frontend library of choice (React.js & other React complementary libraries), with correct markup and styles (understanding HTML & CSS)
Backend: Creating a fully functioning REST API (with Node.js) with persistence (Prisma + PostgreSQL) & auth
CI/CD: Dockerizing the application, basics of deployment, creating pipelines for your projects

All of these areas could be separate courses.

Deadlines

For iterations: Three weeks after the iteration is published. Your tutor might give you more time depending on the circumstances (public holidays and so on)

For team projects: Three days before your presentation (more information later during the semester)

Course info: Evaluation

- Up to **40 points from iterations** (for completing assignments of the semestral project with the best effort and clean code). **Minimal number of points from iterations to get a passing grade is 20**.
- Up to **40 points for your team project** (for creating a complex solution, dividing work, and collaborating with others).
- Up to **20 points for exams** (the final ROPOT contains all the topics from the semester)

Grade	Points
A	100-94
B	93-88
C	87-82
D	81-76
E	75-70
F	69-0
Z	100-60
N	59-0

Let's check your development environment (student devices only)

- We will check whether your Docker works with this command. Alternatively, open Docker Desktop app and try to run the `hello-world` container from there.

```
docker run hello-world
```

- If you use Windows, we will check whether your WSL works. We also recommend you use a full-fledged Linux distribution, such as Ubuntu

```
# You can use a distro of your choice, but we recommend sticking with Ubuntu
# Please, if you haven't already, install it from home or somewhere else than the faculty WiFi
wsl.exe --install -d Ubuntu
```

Git: Setup

1. Install git (Depends on your OS)

   Using a package manager:

   ```
   apt install git     # ubuntu/debian
   brew install git    # mac
   winget install Git  # windows package manager, we advise using Git inside your WSL!
   ```

   Or you can use the Git website.

2. Setup keys

   ```
   ssh-keygen -o -a 100 -t ed25519 -f ~/.ssh/id_muni -C "xuser@fi.muni.cz"
   ```

Git: Setup

3. Add public key to Gitlab

- Go to gitlab.fi.muni.cz and sign in with `xlogin` (or `xUCO` if you're not from FI)
- Click on your profile avatar (top left corner)
- Click `Preferences`
- Find the `SSH Keys` category and click it
- Click `Add new key`
- Paste the content of the **public key** (`id_muni.pub`) into the `Key` text input
- Set the title to something meaningful, as the key cannot be edited, only deleted and added again
- **Remove expiration date**

Git: Setup

4. Update the ssh config (on Mac & Linux `/home/username/.ssh/config`, on Windows `C:\Users\username\.ssh\config`)

```
Host gitlab.fi.muni.cz
    User git
    IdentityFile /home/username/.ssh/id_muni # the path will vary depending on your OS
    IdentitiesOnly yes
```

5. Verify that your key works with `ssh`

```
ssh -T gitlab.fi.muni.cz
# After adding the certificate, should print:
# Welcome to Gitlab, @xlogin!
```

Git: Basics

Commit in Conventional commits

```
git config --global core.excludesFile "**/node_modules"
git config --global user.name xuser
git config --global user.email xuser@fi.muni.cz
```

```
git clone <url>                                          # Clones the repository
git status                                               # Show status of added, removed files
git checkout -b submit-00                                # Switch to new branch
git add -A                                               # Stage all files for the commit
git commit -m "feat: Some meaningful commit message" # Commit all staged files
git push --set-upstream origin submit-00                 # Push commited to remote branch
```

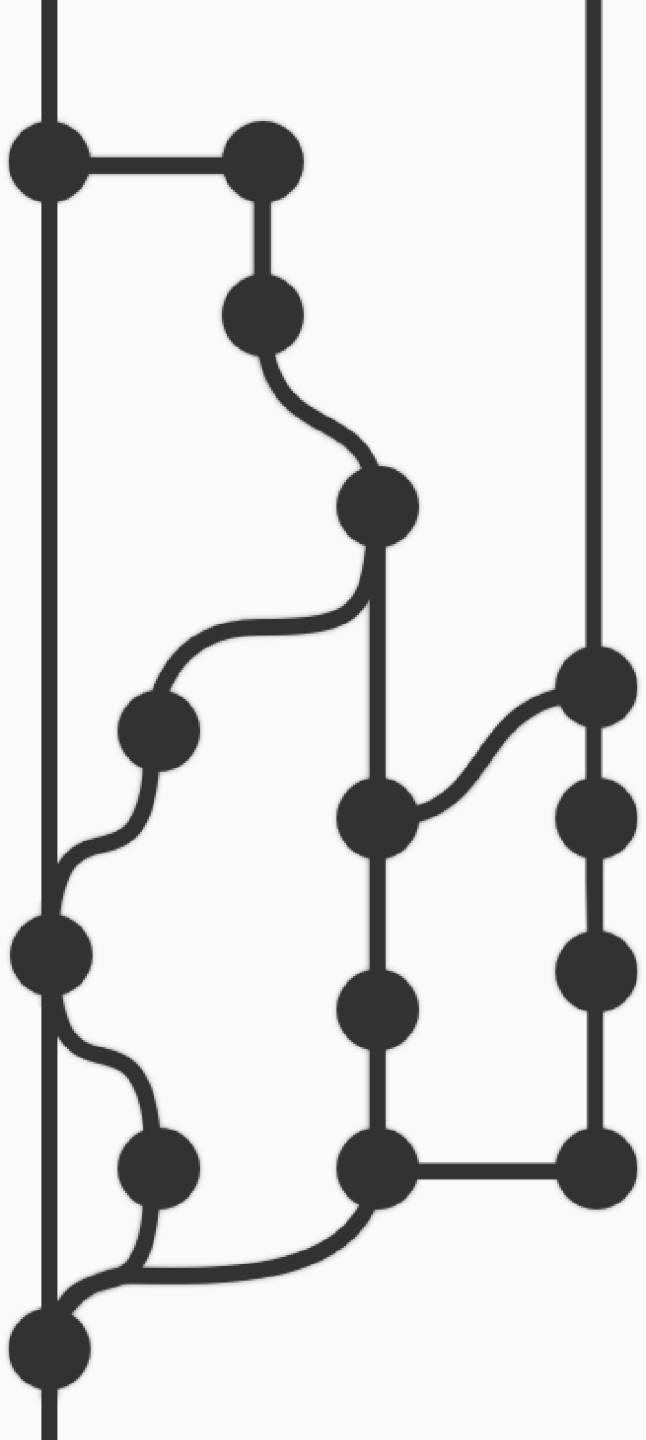
Cheatsheet

Git: Gitflow

- Starts from the main branch
- Feature branches contain new features, additions
- Master/Main is stable (tagged)

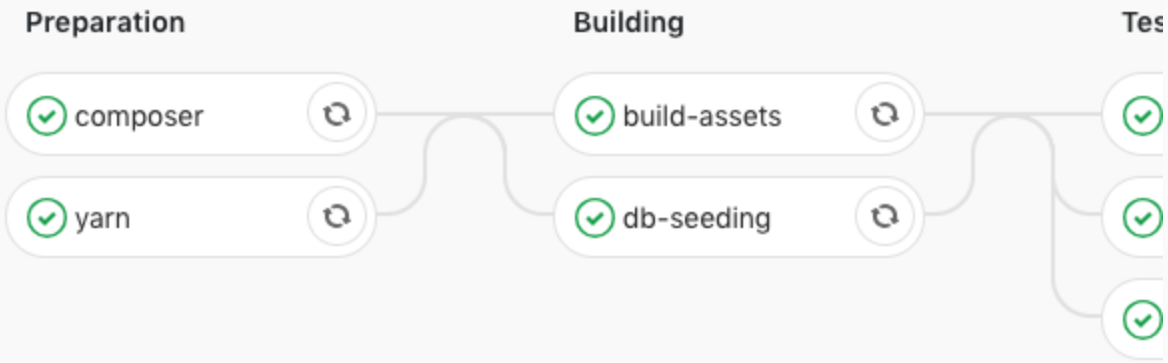
Note: You'll most likely work in trunk based development (it ignores the develop branch)

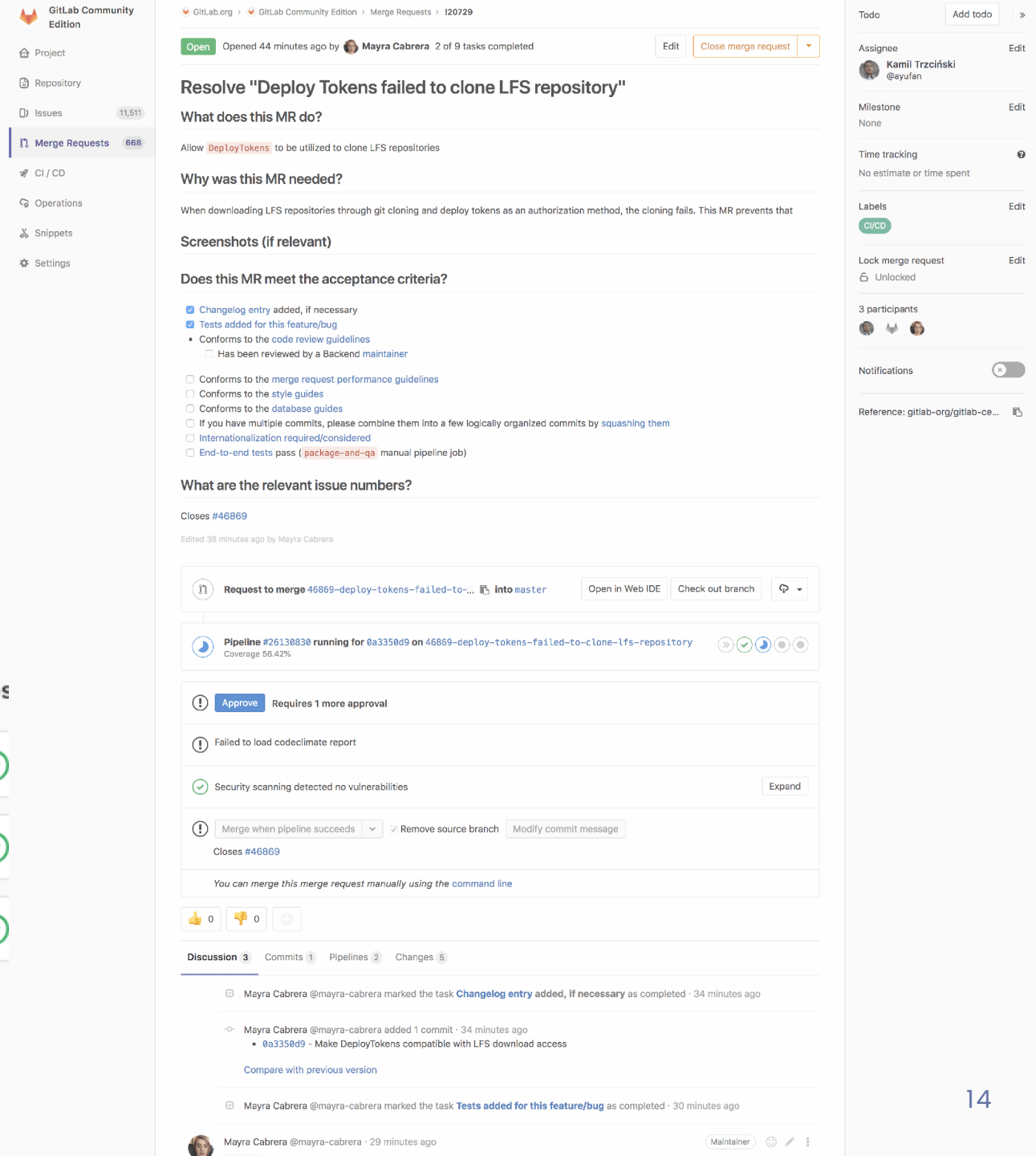
In-depth explanation



Git: Merge requests / Pull request

- MR/PR the way you check source code changes into a main branch
- Before pushing code, **check** common mistakes
- CI is not your enemy
- **We require you to discuss changes**





Let's create your repositories

This might be a lot of information to process. But fret not! We created comprehensive guides for you. You will learn how to:

- Fork the repository - **today!**
- Download the provided assignment template for each iteration - later during the course
- Submit the assignment - later during the course

Don't forget to

- Set your forked repository as **PRIVATE**
- Invite your tutor(s) and **give them at least `Developer` or better yet `Maintainer`** role
- When submitting the assignment, **set your code reviewer as `Asignee`**

Workspace setup

IDE: **Webstorm, VSCode, vim**
Git: **Gitkraken, Github desktop***

Visual Studio Code extensions will be announced for every seminar session (We use many in this course)
**Note: optional, but ignore if you like solving conficts on your own*

Kahoot

Let's get to know each other! Also, let's find out how deep your webdev knowledge goes!

For tutors: Kahoot session