

Introduction to Git and gitlab.awi.de

Onboarding

- Understanding the importance of version control systems
 - Introduction to different version control systems
 - Understanding what Git is and its main features
 - Exploring basic Git commands and workflows
 - Getting to know the basics of Gitlab
-

Version Control Systems (VCS)

- Keep track of changes made to files over time
 - Revert to previous versions of files if needed
 - Collaborate with others on a project
 - Track who made what changes and when
-

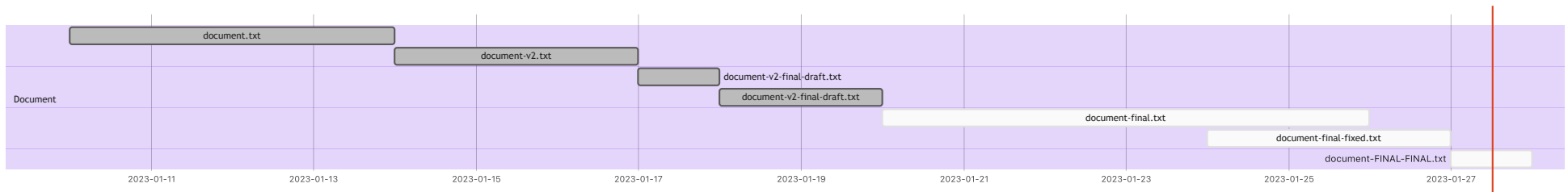
VCS Manual Backups

- Create a file
 - At some point name it v2
 - Have a final draft and send this to a colleague
 - Have another draft
 - Create the final version
 - Get some feedback and create the final fixed version
 - Well... maaaaybe fix a little more so you end up with the final final
-

VCS Manual Backups

```
my-project
├─ document_v1.txt
├─ document_v2.txt
├─ document_final_draft.txt
├─ document_final_draft2.txt
├─ document_final.txt
├─ document_final_fixed.txt
└─ document_FINAL_FINAL.txt
```

VCS Manual Backups (2)



Git

So, what is Git?

- Git allows you to keep track of changes
 - Capability to create branches
 - Allows multiple individuals to work independently
 - Facilitates collaboration by allowing multiple users to work on the same files and seamlessly merge their changes together.
-

Basic git commands

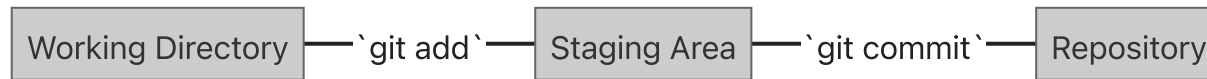
Working locally

- git init
- git add
- git commit
- git status

When working with remotes like Gitlab

- git clone
 - git pull
 - git push
-

The different areas



- **Working Directory** contains all of your files + the `.git` directory for the repository
 - **Staging Area** contains all modifications you added via `git add`
 - **Repository** is your local repository with all files and changes that have been committed
-

What is a commit

- Saving changes
 - Adding a message
 - Moving changes to the repository
-

E: Creating a local repository

- Move into the directory you want to create the repository in
 - Run the command `git init`
 - Create or modify files in this directory
 - Add them to the staging area via `git add`
 - When everything is in the staging area commit the content to the repository via `git commit`
-

Git History

```
git log
```

The git graph



```
git log --all --graph --oneline
```

Branching and Merging

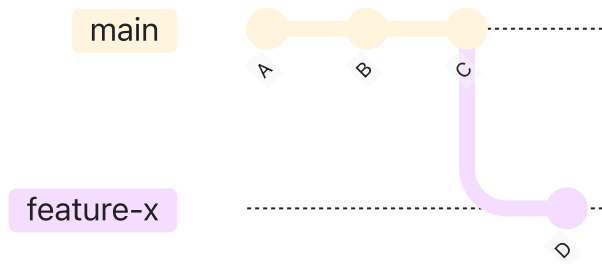
- Understanding the concept of branching
 - Creating and switching branches
 - Merging branches and resolving conflicts
 - Understanding merge strategies
-

What is branching?

Linear git graph



After creating a feature branch and a commit



E4: Branches

- `git switch -c new-feature` to create a new branch called `new-feature`
 - Make some changes to the project files
 - `git add .` to stage the changes
 - `git commit -m "Added new feature"` to commit the changes
-

E5: Merging

- `git switch main` to switch back to the main branch
 - `git merge new-feature` to merge the changes from the new-feature branch into the main branch
-

Merge conflicts

- For example, let's say you have a file called `file.txt` in the main branch, and you've made changes to the same file in a feature branch. The main branch contains the following lines:

```
Hello world
```

- and the feature branch contains:

```
Hello Git
```

- When merging, git will complain and modify the files:

```
<<<<<<< HEAD
Hello world
=====
Hello Git
>>>>>> feature-x
```

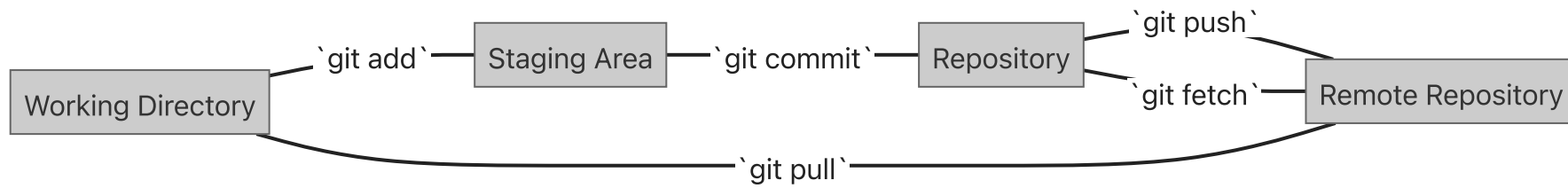
Git Repository Hosting Platforms

- Overview of popular platforms like GitHub, GitLab, Bitbucket and Gitea
 - Understanding the role of GitLab in Git
 - Features and functionalities of GitLab
-

GitLab Feature Overview

- Source Code Management (SCM)
 - Continuous Integration (CI)
 - Issue Tracking
 - Project Management
 - Container Registry
 - Access Control
-

Combining git and GitLab



- [Create a blank project on GitLab](#)
 - Upload content of our repository to GitLab
 - Make changes in the Web IDE
 - Pull the changes to our local repository
-

Next Steps and Additional Resources

- Oh shit, git!?! <https://ohshitgit.com/>
- Git documentation <https://git-scm.com/doc>
- GitHub documentation <https://docs.github.com/en>
- Atlassian Documentation <https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>
- Oh my git <https://ohmygit.org/>
- .gitignore generator <https://www.gitignore.io>