

Gitlab



Ingo Ebel

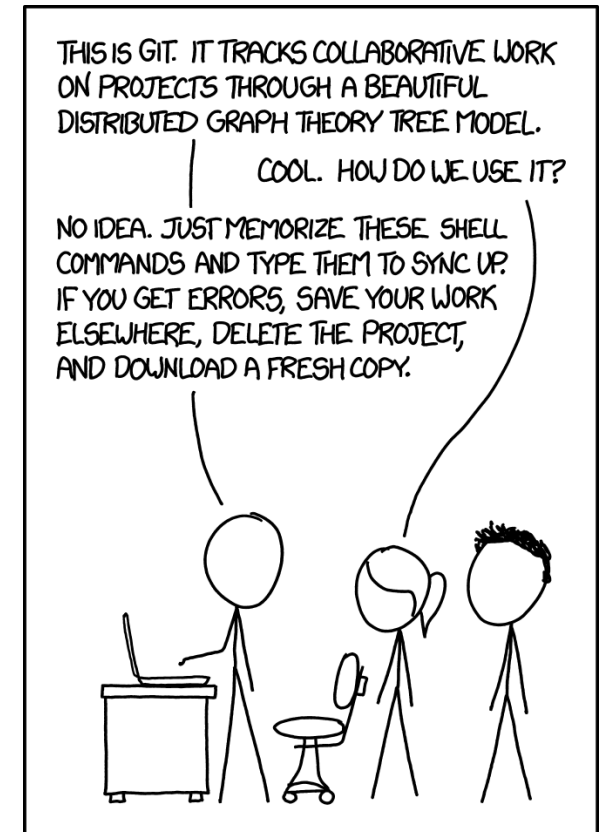
Who am I

- Ingo Ebel
- Working for DESY since 2018
- Part of the team bringing gitlab to desy
- also maintaining the SVN Service

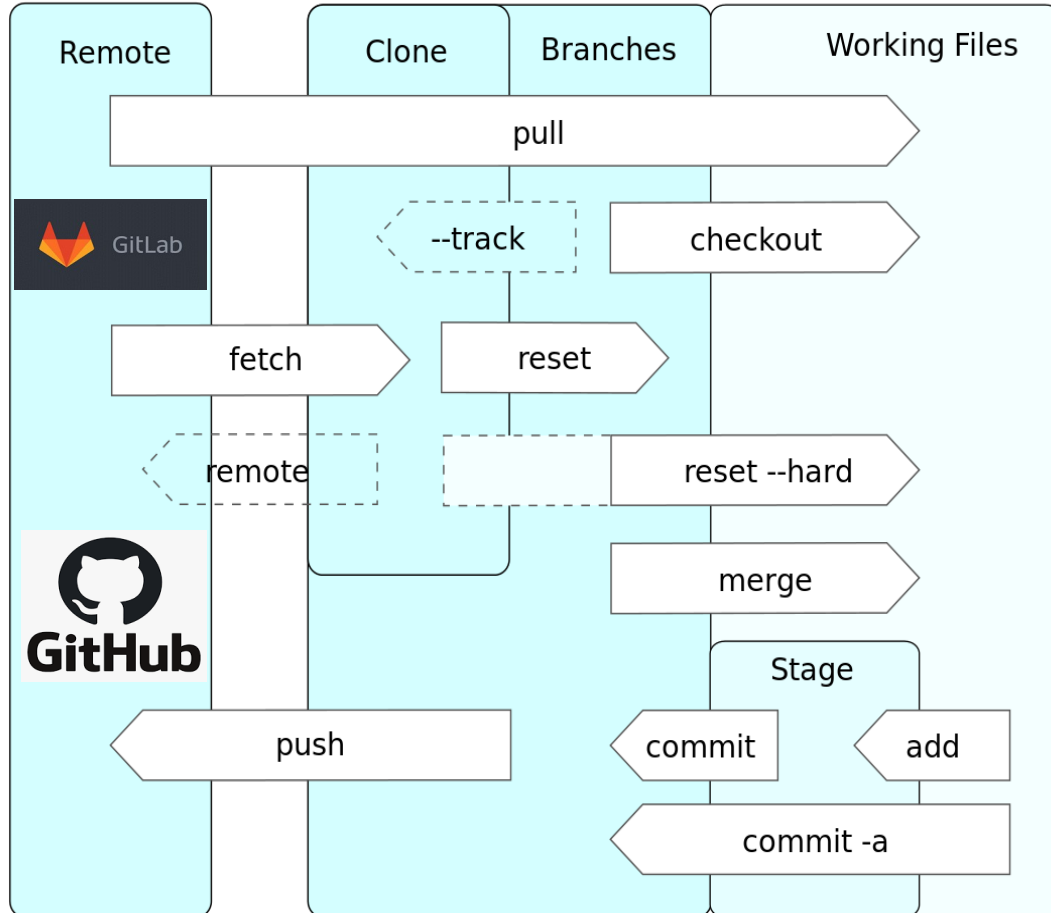


What is git?

- a free and open source distributed version control system
 - Every clone is really a full backup of all the data
 - Clients fully mirror the repository, including its full history
 - Possible to synchronize with several remote servers
 - In a hierarchical model (upstream project, forked to downstream adoption)
- was created by Linus Torvalds in 2005 for development of the Linux kernel
- <https://git-scm.com/book/en/v2>



Operations when working with git remote server



Source (image and text):
https://en.wikipedia.org/wiki/File:Git_operations.svg

git pull fetches remote changes into the local clone, and merges them into the current working files.

git checkout replaces the current working files with files from a branch.

git checkout --track creates a local branch from a remote branch, links them, and replaces the current working files with files from that branch.

git fetch downloads changes from a remote repository into the local clone

git reset makes the current branch point to some specific revision or branch.

git remote manages repositories whose branches the local clone tracks

git reset --hard makes the current branch point to some specific revision or branch, and replaces the current working files with the files from that branch.

git merge merges files from a given branch into the current branch.

git push uploads changes from local branches to the respective remote repositories.

git add puts current working files into the stage (aka index or cache)

git commit commits staged changes to a local branch

git commit -a commits all modified files to a local branch (shorthand for "git add" and "git commit")

What is gitlab?

- Webfrontend for git
 - Web IDE
 - Source Code Management
 - Code Review
 - Wiki
- DevOps platform
- Issues / Boards
- CI/CD
- Container Registry



Log in

- Via DESY Account
 - Just login and start your projects
- Via Helmholtz AAI
 - Nearly every University
 - and research institute
 - Or even google account and github
 - Users must request to be unblocked

https://gitlab.desy.de/users/sign_in

Suchen

DESY

Standard

DESY Username

Password

Remember me

Sign in

Sign in with

Remember me

Welcome to the GitLab at DESY

This is the DESY GitLab for

- Repository hosting
- continuous integration workflows
- container registry hosting

Public Repos

This is the Member Area. If you want the public Repositories please go to the [List of public Repos](#).

So many gitlabs

- gitlab.zeuthen.desy.de
 - Just for Users from DESY Zeuthen
- gitlab.desy.de
 - Main Gitlab, will replace Stash
- gitlab.cta-observatory.org
 - For CTA Projects
- HIFIS: gitlab.hzdr.de
 - For Helmholtz Projects



Add SSH Key

- Go to: User Settings → SSH Keys
- Add your key
- Then you can use the git clone via s:

User Settings > SSH Keys

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

To add an SSH key you need to [generate one](#) or use an [existing key](#).

Key

Paste your public SSH key, which is usually contained in the file '~/.ssh/id_ed25519.pub' or '~/.ssh/id_rsa.pub' and begins with 'ssh-ed25519' or 'ssh-rsa'. Do not paste your private SSH key, as that can compromise your identity.

Typically starts with "ssh-ed25519 ..." or "ssh-rsa ..."

Title

e.g. My MacBook key

Expires at

TT . MM . JJJJ

Give your individual key a title.

Add key

Your SSH keys (1)

ingo



Use Access Tokens

- Use tokens on virtual machines and CI Pipelines, not your username and password
- Personal Access Tokens
 - Go to your Profile → Access Tokens
- Project Access Tokens
 - In a project go to Settings → Access Tokens
- Project Deploy tokens
 - Settings → Repository
- Group Deploy Tokens
 - In a Group go to Settings Repository → Deploy Tokens
- Define scopes, expire dates etc.

Or use Deploy Keys

- SSH Keys
- Read-only or read-write to a repository
- shareable between projects that are not related or don't even belong to the same group
- no expiration date
- See also
https://gitlab.desy.de/help/user/project/deploy_keys/index

The power of gitlab

Issue Boards

GitLab.org > GitLab > Issue Boards

The screenshot shows the GitLab Issue Boards interface. At the top, there's a navigation bar with 'Development W...' and a search bar. Below that, there are three boards:

- Open**: 33931 issues, 6205 subscribers. Issues include:
 - #285529: UI tip for why job Retry is not allowed for non-draft MR with merge trains enabled. Labels: Continuous Integration, SUS, FY22 Q1 - Incomplete, UI polish, UX, candidate, 13.10, devops, verify, feature, enhancement, frontend, group, continuous integration, pd, doing, section, ops, workflow, design.
 - #35135: Users cannot re-add merge request to a merge train when "pipeline must succeed" is enabled. Labels: Continuous Integration, Merge Trains, FY22 Q1, UI text, UX, approaching-SLO, bug, bug, transient, candidate, 13.10, customer, devops, verify, frontend, group, continuous integration, merge requests, missed-SLO, pd, doing, priority 1, section, ops, severity 2, workflow, design.
- workflow blocked**: 289 issues, 296 subscribers. Issues include:
 - #263240: Sync existing branch and commit data when a namespace is added to Jira Connect App. Labels: Integrations, Enterprise Edition, Integration, Jira, Jira, ConnectApp, Stretch, backend, devops, create, feature, feature, addition, frontend, group, ecosystem, priority 1, section, dev.
 - #249579: Create a new Jira issue from unresolved MR threads. Labels: Accepting merge requests, Category: Integrations, Integration, Jira, atlassian, customer, devops, create, group, ecosystem, priority 2, section, dev.
 - #321267: Automatically create change request ticket in ServiceNow. Labels: Accepting merge requests, Category: Integrations, Enterprise Edition, GitLab Premium, GitLab Ultimate.
- workflow ready for development**: 983 issues, 1539 subscribers. Issues include:
 - #223793: API to bulk delete artifacts. Labels: Accepting merge requests, CI artifacts, Category: Continuous Integration, GitLab Core, Pricing & Packaging, backend, candidate, 13.10, customer, devops, verify, direction, documentation, feature, group, continuous integration, missed-deliverable, missed:13.3, missed:13.4, quad-planning, ready, release post item, release post item, in review, section, ops, stage-efficiency.
 - #321267: Allow "pipeline must succeed" to merge if the pipeline was empty. Labels: Accepting merge requests, Category: Continuous Integration, GitLab Core, backend, candidate, 13.10, customer, devops, verify, direction, feature, group, continuous integration, merge requests, quad-planning, ready, section, ops.

- Issue tracking
- Project management
- Can be used as
 - Kanban Board
 - Scrum Board
- Multiple issue boards = premium feature :(

Source:

<https://gitlab.com/gitlab-org/gitlab/-/boards/1339417>

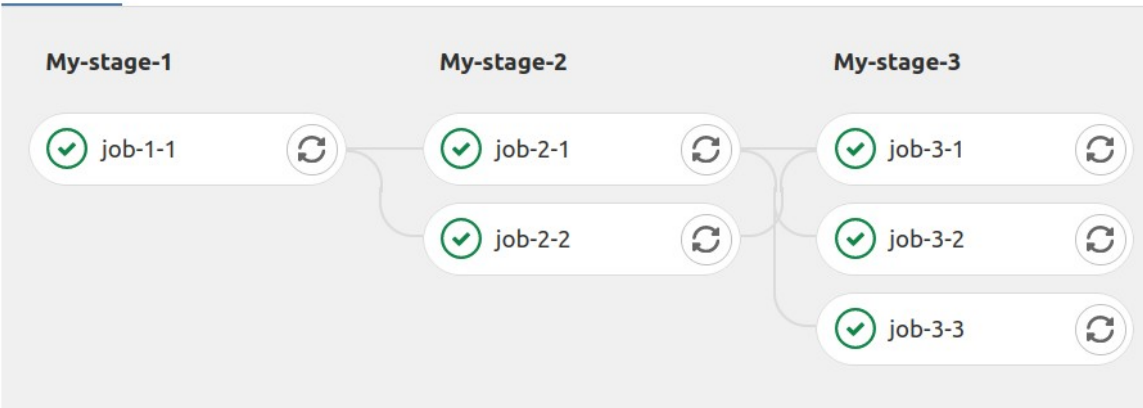
https://docs.gitlab.com/ee/user/project/issue_board.html

GitLab CI/CD Pipeline

GitLab CI/CD pipelines are configured using a YAML file called `.gitlab-ci.yml` within each project.

- What to execute in your CI/CD pipelines
- What to do on success / failure

Pipeline Needs Jobs 6 Tests 0



6 jobs for `master` in 32 seconds (queued for 5 seconds)

Source:


<https://gitlab.com/gitlab-org/gitlab/blob/master/.gitlab-ci.yml>

<https://docs.gitlab.com/ee/ci/yaml/>

```
.gitlab-ci.yml 724 Bytes
1  image: ubuntu:latest
2
3  stages:
4    - my-stage-1
5    - my-stage-2
6    - my-stage-3
7
8  job-1-1:
9    stage: my-stage-1
10   script:
11     - echo Hi from JOB $CI_JOB_NAME in STAGE $CI_JOB_STAGE
12
13  job-2-1:
```

GitLab CI/CD Variables

- Use builtin variables with Gitlab CI/CD
- And define own variables
 - Settings → CD / CD → Variables (projects and groups)

Type	↑ Key	Value	Protected	Masked	Environments	
Variable	MM_WEB_HOOK	*****	✓	✓	All (default)	

```
13 variables:
14   IMAGE: $CI_REGISTRY_IMAGE/chatops-env:latest
15   DOCKER_HOST: tcp://localhost:2375
16   DOCKER_TLS_CERTDIR: ""
17 script:
18   - docker login -u gitlab-ci-token -p $CI_JOB_TOKEN $CI_REGISTRY
19   - cd chatops && docker build --pull -t $IMAGE .
20   - docker push $IMAGE
21 only:
22   changes:
23     - Dockerfile
```

```
12 def post_mattermost(schedule):
13     url = os.environ["MM_WEB_HOOK"]
14     payload= {
15         "channel": "gitlab-test-project",
16         "username": "gitlab",
17         "text": schedule
18     }
```

Source:

<https://gitlab.com/gitlab-org/gitlab/blob/master/.gitlab-ci.yml>

<https://docs.gitlab.com/ee/ci/yaml/>

https://docs.gitlab.com/ee/ci/variables/predefined_variables.html#predefined-variables-reference

ChatOps



- Interact with CI/CD jobs through chat service
- ChatOps looks up the specified job name in `.gitlab-ci.yml`.
- A pipeline containing only that job is scheduled.

Settings → Integrations

✓	Mattermost notifications	Receive event notifications in Mattermost
✓	Mattermost slash commands	Perform common operations in Mattermost

```
29 hello:
30   stage: chatops
31   only: [chat]
32   script:
33     - if [ -z "$GITLAB_USER_NAME" ]; then echo "hello Mattermost"; else echo "Hello $GITLAB_USER_NAME";fi
```

Source: <https://docs.gitlab.com/ee/ci/chatops/index.html>

Using Chat Ops

The screenshot shows a chat interface with two messages from the GitlabCI bot and a notification from Ingo Ebel. The first message from the bot reports a push to the master branch. The second message reports a failed pipeline with a table of details.

GitlabCI BOT 16:22
Ingo Ebel pushed to branch master of rpm / test2 (Compare changes)

bd12a8a0: Update .gitlab-ci.yml - Ingo Ebel

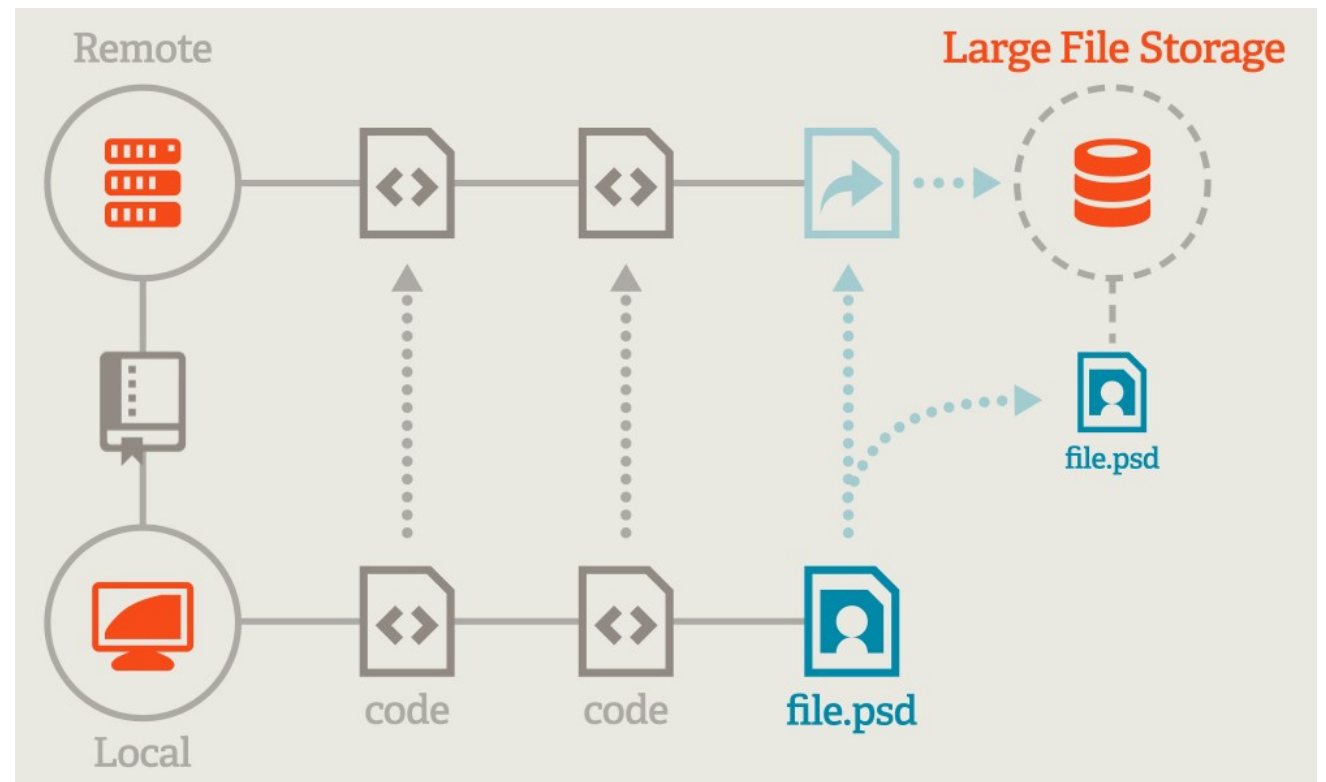
GitlabCI BOT 16:22

Ingo Ebel (ebeli)
Pipeline #599 has failed in 00:19

Branch	Commit
master	Update .gitlab-ci.yml
Failed stage	Failed job
build	build
test2	

Using git-lfs

- Git extension for versioning large files
 - No binaries in git repos for performance reasons
- Supported by our gitlab installation
- Needs support by your git client
 - In a git-repo: git lfs install
 - git lfs track "*.psd"



Limitations / Missing Things

- Can't reach internal desy services for gitlab/gitlab-ci (e.g. NIMS, repos or resources on local machines)
 - Use own gitlab runner
 - Use the gitlab.zeuthen.desy.de CI
- Working on user documentation
- No gitlab pages at the moment
- Open Source / Community Version
 - Some enterprise features are missing

Migration

Migration

- From svn
 - Install git-svn: apt-get install git-svn
 - git svn clone https://svnsrv.desy.de/desy/logstash
 - cd logstash
 - git remote add origin git@gitlab.desy.de:ingo.ebel/logstash.git
 - git push -u origin --all
- From stash/bitbucket
- From Jira
- From github
-
- <https://gitlab.desy.de/help/user/project/import/index.md>

The future

Next Steps

- Finalizing the backup strategy
- Finalizing the operational concept (Betriebskonzept)
- Operational in Q2 2021



Contact us

- Write to UCO or UCO-ZN
- Go to <https://chat.desy.de/desy/channels/gitlab>

Links

- CI Examples: <https://gitlab.desy.de/documentation/examples>
- User Documentation / FAQs:
<https://gitlab.desy.de/documentation/wiki/-/wikis/home>
- <https://gitlab.desy.de/public>
- <https://gitlab.desy.de/help/>
- <https://git-scm.com/book/en/v2>

Yeah!

Q&A