

Picture 1. Basic GitLab Flow, Good Practices

Install and configure git if not already done: <https://git-scm.com/downloads>

1. If Git isn't configured, set username and email e.g.:
2. git config --global user.name "John Doe"
3. git config --global user.email "John.Doe@etteplan.com"

Set ssh-keys

1. You need access rights to GitLab: *etgitlab1.etteplan.com*
2. Master-branch is here: <https://etgitlab1.etteplan.com/SES_Lab/job-searcher-master>
3. Both master and production branches are protected, merge requests are used
4. Fist from top right corner - go to Preferences/SSH keys
5. Check possibles ssh-keys: ls ~/.ssh/\*.\*
6. If ssh-keys not generated generate them e.g.:
7. ssh-keygen -t ed25519 -C “GitLab Key Pair”
8. cat ~/.sssh/id\_ed25519.pub
9. and copy the ssh-key into the ssh-key text-box.
10. Now you should be able to Clone a repository with SSH and also to upload your own feature-branches

Committing changes to master-branch:

1. Create your own branch: **git checkout -b <your name>-feature-branch**
2. Make your additions, test those and commit the changes with informative commit messages.
3. Check the branches: **git branch**
4. With following command push your changes to master: **git push -u origin <your name>-feature-branch**
5. Then the merge request can be approved and done at GitLab website by the person having the access rights – meaning merging **<your name>-feature-branch to master** (which is default)
6. Both master-branch and production-branch are protected
7. After successful merge you can move back to local repo and delete <your name>-feature-branch
8. **git branch -d <your name>-feature-branch**
9. Check current branches at local repo: **git branch --all** (you will see all active branches)
10. To actually delete <your name>-feature-branch also from .git give command: **git pull --prune**
11. Now check again with command: **git branch –all**, you won’t see your feature-branch anymore.