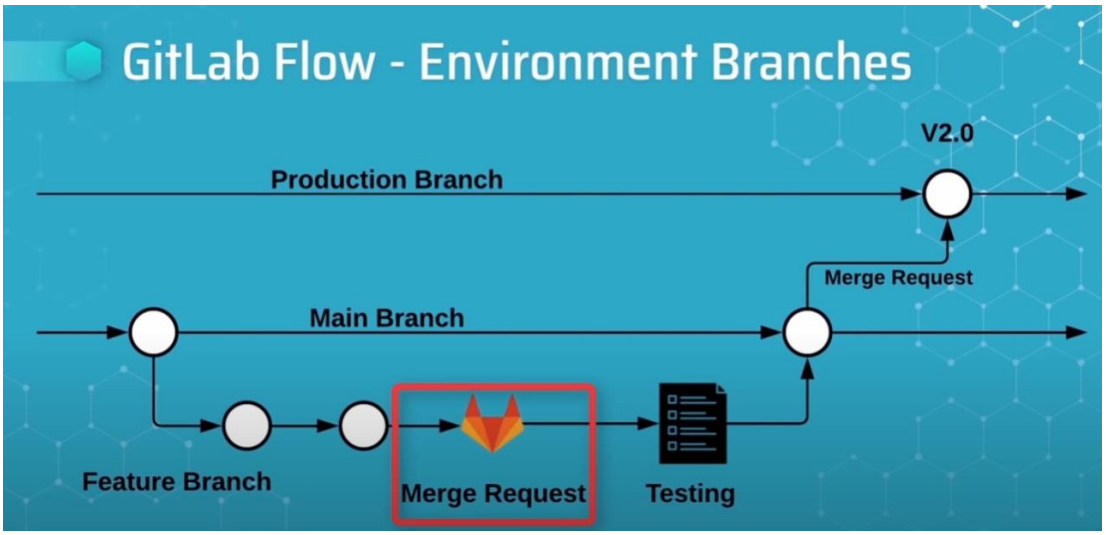

Picture 1. Basic GitLab Flow, Good Practices

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

4) You need access rights to GitLab: *etgitlab1.etteplan.com*
5) Master-branch is here: https://etgitlab1.etteplan.com/SES_Lab/job-searcher-master
6) Both master and production branches are protected, merge requests are used
7) Fist from top right corner - go to Preferences/SSH keys
8) Check possibles ssh-keys: ls ~/.ssh/*.*
9) If ssh-keys not generated generate them e.g.:
10) ssh-keygen -t ed25519 -C “GitLab Key Pair”
11) cat ~/.sssh/id_ed25519.pub
12) and copy the ssh-key into the ssh-key text-box.
13) Now you should be able to Clone a repository with SSH and also to upload your own feature-branches

Committing changes to master-branch:

14) Create your own branch: **git checkout -b <your name>-feature-branch**
15) Make your additions, test those and commit the changes with informative commit messages.
16) Check the branches: **git branch**
17) With following command push your changes to master: **git push -u origin <your name>-feature-branch**
18) 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)
19) Both master-branch and production-branch are protected
20) After successful merge you can move back to local repo and delete <your name>-feature-branch
21) **git branch -d <your name>-feature-branch**
22) Check current branches at local repo: **git branch --all** (you will see all active branches)
23) To actually delete <your name>-feature-branch also from .git give command: **git pull --prune**
24) Now check again with command: **git branch –all**, you won’t see your feature-branch anymore.

GitLab Flow short notes 20230207