<https://www.youtube.com/watch?v=Ddd3dbl4-2w>

Version control system (VCS):

· Lets you retrieve any previous version.

· Lets the team work on multiple versions simultaneously, like a branching tree.

· Lets you “merge” changes between one branch of the tree to the other

Git – A distributed VCS:

· In git, every client can, if configured in the right way, also act as a server! git does not require a single central server, though many people use it that way; it works very well at that too!

Creating a git repository on Gitlab:

· Click the “New Project” (indicated by a ‘+’ sign) and select “New Repository” to create a new repository. Select “Blank project”

· Once selected, a form will allow you to select a number of options, including: the name of your repository, a longer description, whether it should be private, internal, or public. **Choose private.**

· The new empty repository is then created:

Cloning repository to local machine:

· Copy the URL of repository (there’s a handy copy button next to the URL)

· Follow git instructions in screenshot above – ‘**Create a new repository’** to add a README file using command line. Can also do this through GitHub desktop but easier to do straight from command line.

· When committing files important to supply informative commit messages – it means that others can figure out *why* you were making the changes, as well as what you were changing.

· Push commit – push file to repository

· Pull commit – pull file from repository

· Generally, *before* you commit, you should do a git pull to incorporate other people’s changes first.

· You should commit changes regularly, but it is generally considered poor practice to commit code that has syntax errors to the *master* (default) branch. Fix it before committing and pushing!

CREATING A BRANCH:

· However, sometimes you don’t want all your changes to be immediately visible to all your colleagues (or, at least, visible when they do a git pull).

· Git accommodates this by using branches

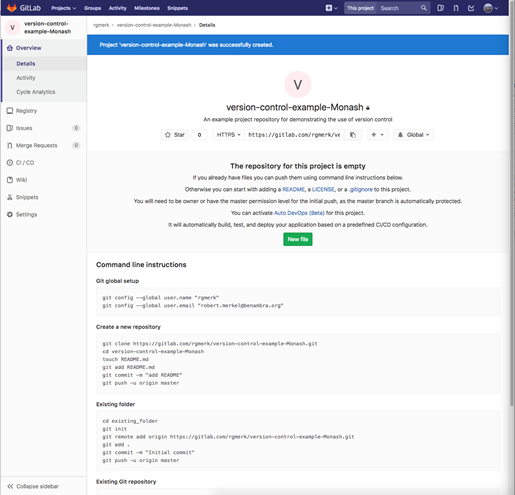
The .gitignore file

· To add all your changes files to the commit at once, you can do “git add <directory>”

· However, this can be a bit problematic if there are temporary autogenerated files that you don’t want to commit to the repository.

· The .gitignore file is a list of filename patterns which git will ignore.

· Typically, repository .gitignore files are checked in to the repository itself. So let’s add a simple one to the master branch rgmthat ignores all files ending in “.swp”:



· We can commit and push it upstream – and the .swp file isn’t added to the commit

SIMPLE MERGING:

· When multiple people are working on the source code at once, git needs to merge the resulting work together.

· Merging can happen within a *single* branch if multiple people are working on it at the same time, or you can have a merge between two branches.

USEFUL GIT COMMANDS TO USE IN COMMAND LINE:

· cd – to go to file location

· git pull origin master –allow-unrelated-histories

· git push

· git pull

· git status – check status of git file