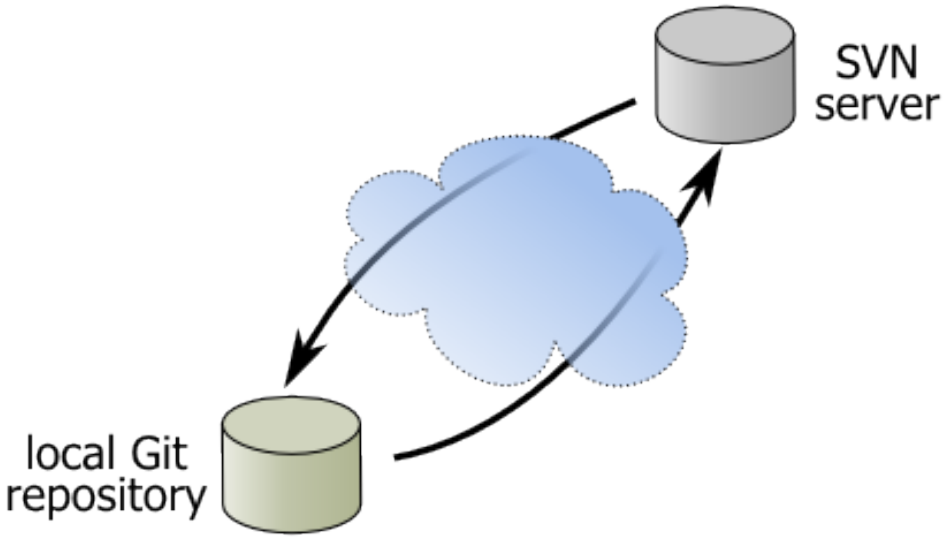
Why use git?

- Backtrack quickly when you make mistakes
- Develop code features separately
- Easily share your code and make it available
- Invaluable when multiple people are working on a project

How it works



- SVN server (e.g. github) main location of code
- Local copy stored on your machine
- git is used to upload/download changes (gracefully) to your local repository

Bitbucket vs. Github

Bitbucket

- Private repositiories
- Good for personal code
- Good for small private projects

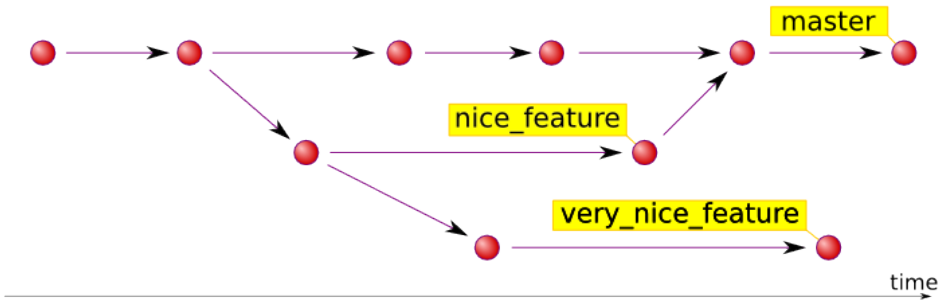
Github

- Public repositories
- Good for sharing your code
- Good for looking at other people's code

Bitbucket & Github features

- History of changes with ability to backtrack
- Differences in changes
- Easily have others review your code
- Overview of code commits

Using git



- Upload changes to server periodically
- New features can be developed separately in a branch. Does not affect main code (master)
- Once branches are finished, they can be merged with the master branch

Creating a new repository

- Use the new repository option on the Bitbucket or Github websites.
- Copy the empty repository to local machine using `git clone username@host:path/to/repository`. Specific command can be copied from repository web page on Bitbucket/Github.
- Sync these new files to the cloud (next slide)...

Syncing changes

- Add any new files: `git add .` or replace `.` with specific filenames.
- Commit changes to your local index
  `git commit -m "Message to yourself"` (messages to yourself are the first sign of madness!!
- Sync the last commit to the cloud: `git push`
- Bit of a keyboardful isn't it...

Be a lazygit

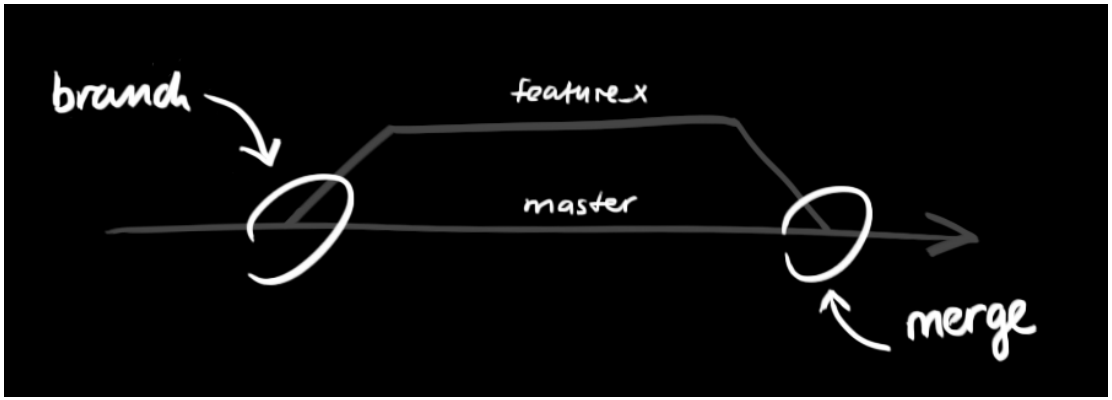
- Add this to the file `.bashrc` in your `/home/username` or `~` directory (create it if it doesn't exist).

```
function lazygit() {
    git add .
    git commit -a -m "$1"
    git push
}
```

- Now do everything on the previous slide by typing `lazygit "message"`

Branching

- Branching allows you to code up new features in isolation from your original project.
- Create a new branch: `git checkout -b name_of_branch`
- Push branch to server: `git push origin name_of_branch`



Merging Branches

- Once a feature has been created branches can be merged
- Switch back to master branch: `git checkout master`
- Merge branch to current branch:
  `git merge name_of_branch`
- Delete the branch: `git branch -d name_of_branch`

More Resources

- Setup SSH in Bitbucket: `https://confluence.atlassian.com/display/BITBUCKET/Set+up+SSH+for+Git`
- git - the simple guide: `rogerdudler.github.io/git-guide/`
- Bitbucket 101: `https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101`
- git in 5 mins: `http://classic.scottr.org/presentations/git-in-5-minutes/`