About Git

Get a copy of the course repository

The command `git clone` will create a local copy of a repository. I.e for the course repository you can use `git clone git@<remote repository>` if you have a ssh-key configured setup in your remote repository. If you prefer to use a access token or username & password (not recommended), use `git clone https://<remote repository>` *NOTE* that there is a difference in the protocol part of the url (https vs git), this determines which authentication method is used.

```
# I recommend you to create a folder in i.e you home folder, that is named repos. i.e on linux/mac
mkdir -p ~/repos

# Change directory to the folder where you want to store your project
cd ~/repos/

# To clone the course repository use
git clone <url>
```

If you use `git clone <url>` , it creates a folder named as the repository.

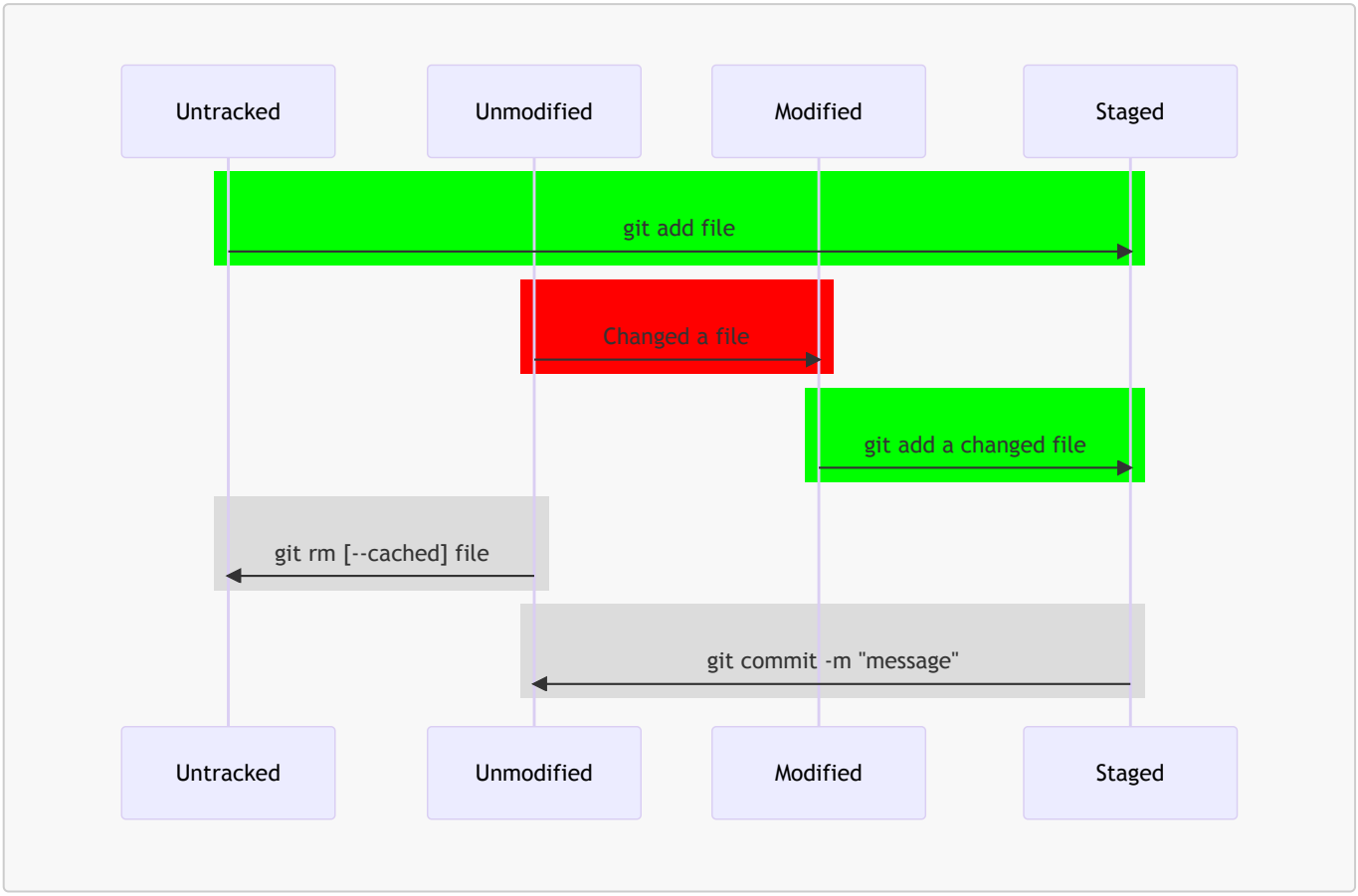
Configure Git

Before committing you should config your `user.name` and `user.email` for the repository or globally (sets a default for git).

```
# Set globally, i.e same for all
git config --global user.name "Firstname Lastname"
git config --global user.email "youremail@yh.nackademin.se"
```

Local Files in Git

Files can either be `Untracked` or `Tracked`. If a file is tracked it recognized by git, a tracked file can be `Unmodified`, `Modified` or `Staged`. You can see the status of all files with `git status`.



All files are `Untracked` until you add them to your local repository.

Adding a file

```
# Within your working directory create a file, on linux/mac
touch my_new_file

# Run `git status`
git status
```

The output:

```
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        my_new_file

nothing added to commit but untracked files present (use "git add" to
track)
```

To track the file:

```
# Add the file to git
git add my_new_file

# Run git status
git status
```

The output:

```
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   my_new_file
```

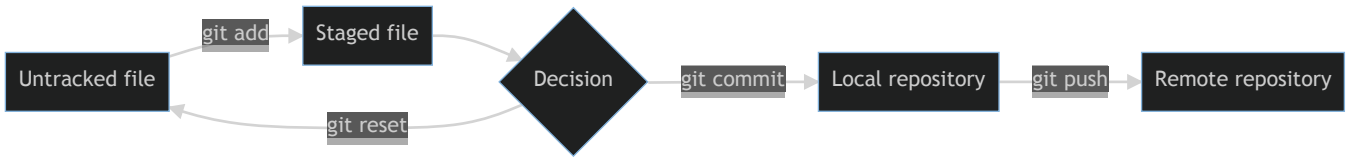
At last you commit the staged file to your local repository

```
# Use git commit
git commit -m "A message"
```

Push files to remote in Git

So far all changes has been local, in a distributed project you will usually have a git server i.e GitLab or GitHub. When working multiple persons in a project it's good practice to use branches and add changes to the project through a `merge request` (GitLab) or `Pull Request` (GitHub).

Basic flow adding a file to git



Previously we added the `my_new_file` to the main branch in our local repository. Its not good practice to add empty files "dummy files", another problem is that we modified the main branch. Undoing changes after a push can be very difficult, so we will remove the `my_new_file` commit before pushing.

```
# To view historic commits use git log
git log

# From the log output copy the commit sha before your change that added
`my_new_file`
# We will use git reset --hard <commit sha> to reset the repository *USE
WITH CAUTION*
git reset --hard <commit sha from git log>
```

Now when the main branch is restored we can create a branch for our changes

```
# Make sure you are currently working on main, a asterisk mark current
branch
git branch

# Create a branch from current working directory
git branch your_username

# Change to the branch with git switch
git switch your_username

# Create a file, add it and commit it.
touch my_file # works on linux/mac
git add my_file
git commit -m "added a example file"
```

Now push to update the remote repository

```
# To push the changes to GitLab use git push
git push
```

The first time you push to a branch it will output:

```
fatal: The current branch your_username has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin your_username
```

It's required to set the upstream before you can successfully push, so just rerun the push once with:

```
git push --set-upstream origin your_username
```

Go to the GitLab or GitHub repository and verify that your change has been uploaded.

Pull latest file updates from remote in git

To get updates, i.e the last lessons code examples or todays lesson you need to fetch the changes from the remote repository. But before you try to get new changes you should always save your current work. Run `git status` to make sure you have no `Untracked` or `Modified` files. `git status` will also show which branch you are currently on. If all files are in order it should output:

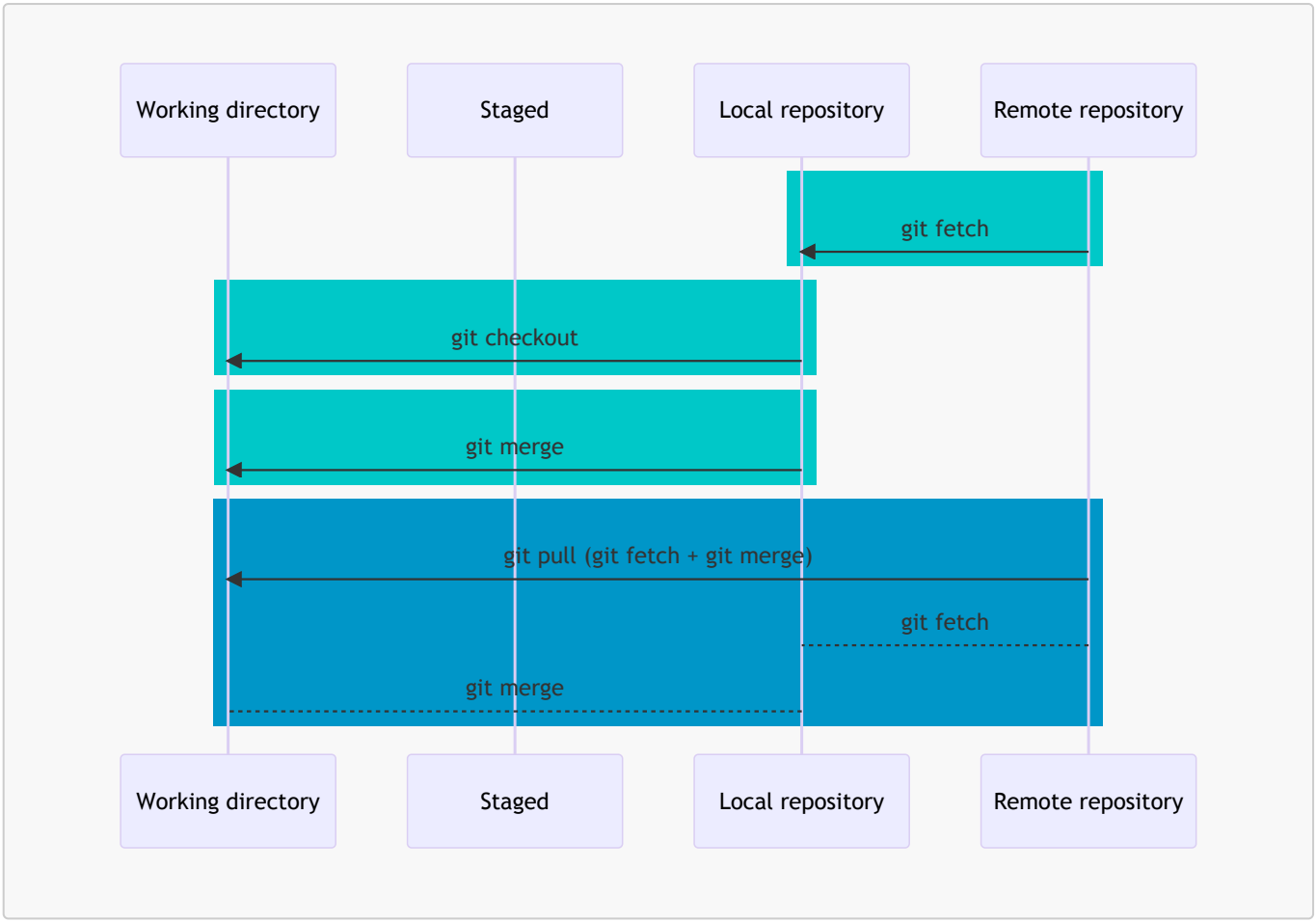
```
On branch your_username
Your branch is up to date with 'origin/your_username'.

nothing to commit, working tree clean
```

Update your branch

```
# To pull the latest changes you can either switch back to main then do a
git pull
git switch main
git pull
git switch <your branch>
git merge main

# Alternatively to update current branch with latest main in one command
git pull origin main
```


Working directory
Staged
Local repository
Remote repository
git fetch
git checkout
git merge
git pull (git fetch + git merge)
git fetch
git merge
Working directory
Staged
Local repository
Remote repository