Process Discovery using Python (Practicum)

Introduction to Git

Mahsa Pourbafrani

Lab WS2020-2021

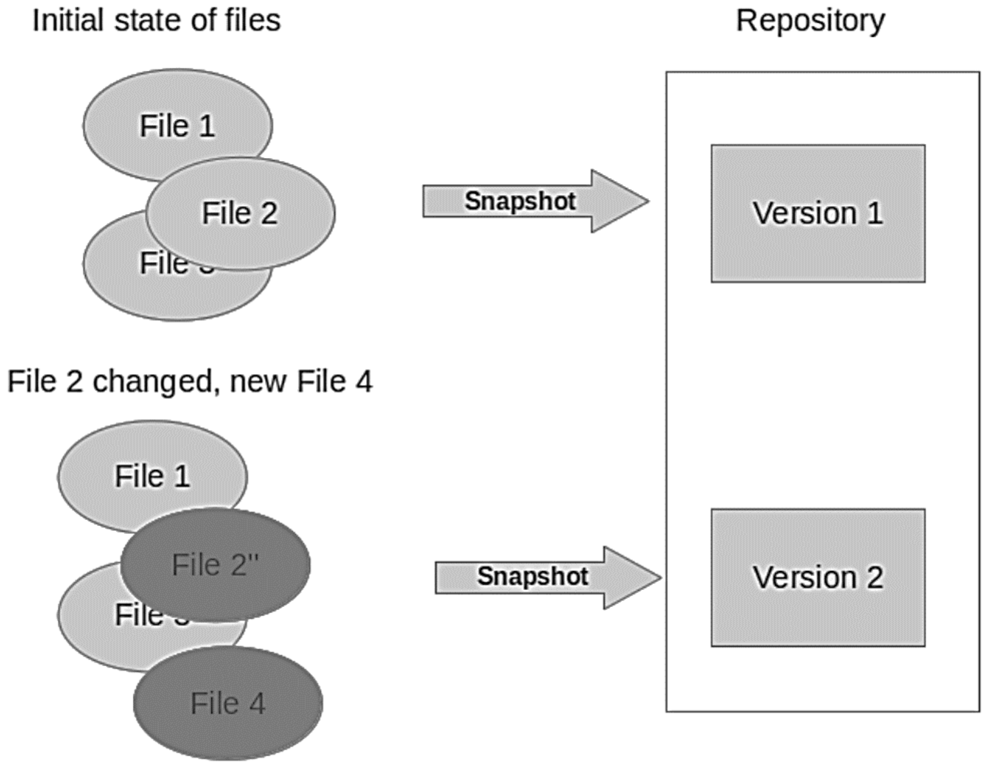
Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Version control system

- Keep track of changes happening to files/directories to get a specific version of the file over time.
- Characteristics of a VCS:
  - Data integrity
    - each small change shall be tracked, and never be lost
  - Possibility to get a specific version a file
  - Speed
  - Collaboration: the same files may be edited by several people

Version Control systems



Source: Google image

Introduction to git

- Primarily, used for source-code management in software development.
- Git was created by Linus Torvalds in 2005 for the development of the Linux kernel.
- Every Git directory on every computer is a full-fledged repository with a complete history and full version tracking abilities.
- Most of Git is written in C
- Compatible with Windows, mac OS, UNIX

Git repository

- A git repository contains the history of a collection of files starting from a certain directory.
- The process of copying an existing git repository via the git tooling is called ***cloning***.
  - After cloning a repository the user has the complete repository with its history on his local machine.
- git supports the creation of new repositories.
  - If you want to delete a git repository:
    - you can simply delete the folder which contains the repository.
  - If you clone a git repository:
    - by default, git assumes that you want to work in this repository as a user.
  - git also supports the creation of repositories targeting the usage on a server.

Working tree

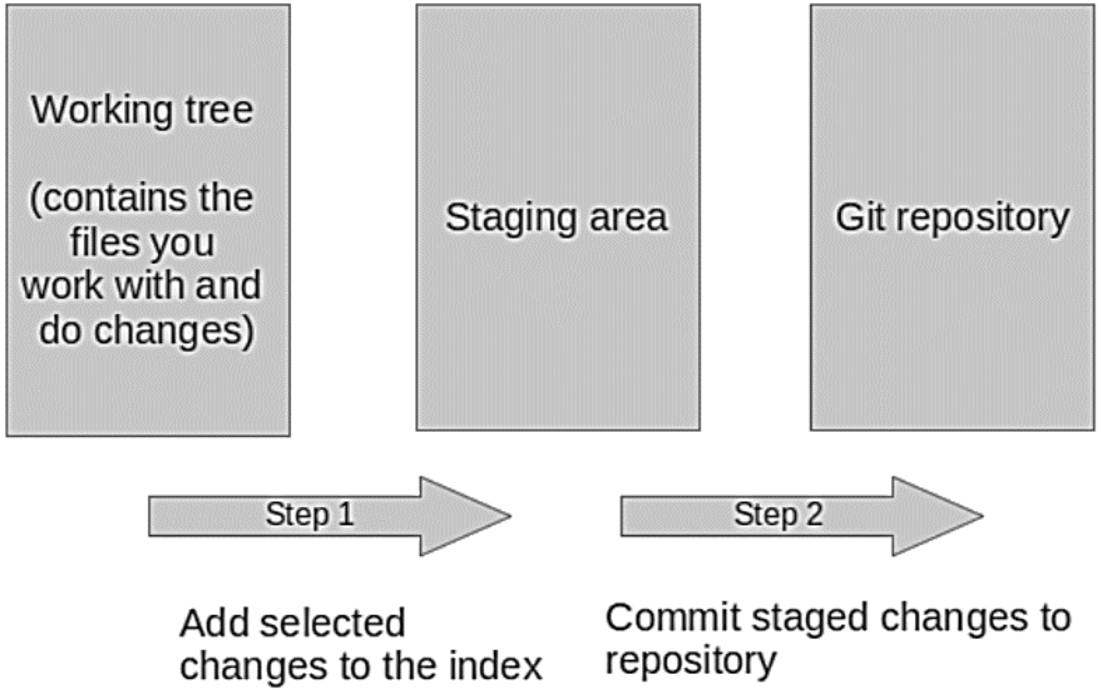
- A local repository provides at least one collection of files which originate from a certain version of the repository.
  - This collection of files is called the *working tree*.
- It corresponds to checkout of one version of the repository with potential changes done by the user.
- The user can change the files in the *working tree.*
  - by modifying existing files and by creating and removing files.

Working tree

- A file in the working tree of a git repository can have different states:
  - **untracked**: the file is not tracked by the git repository.
    - This means that the file never staged nor committed.
  - **tracked**: committed and not staged
  - **staged**: staged to be included in the next commit
  - **dirty /modified**: the file has changed but the change is not staged
  - After doing changes in the working tree, the user can add these changes to the git repository or revert these changes.

Adding to a git repository via staging and committing

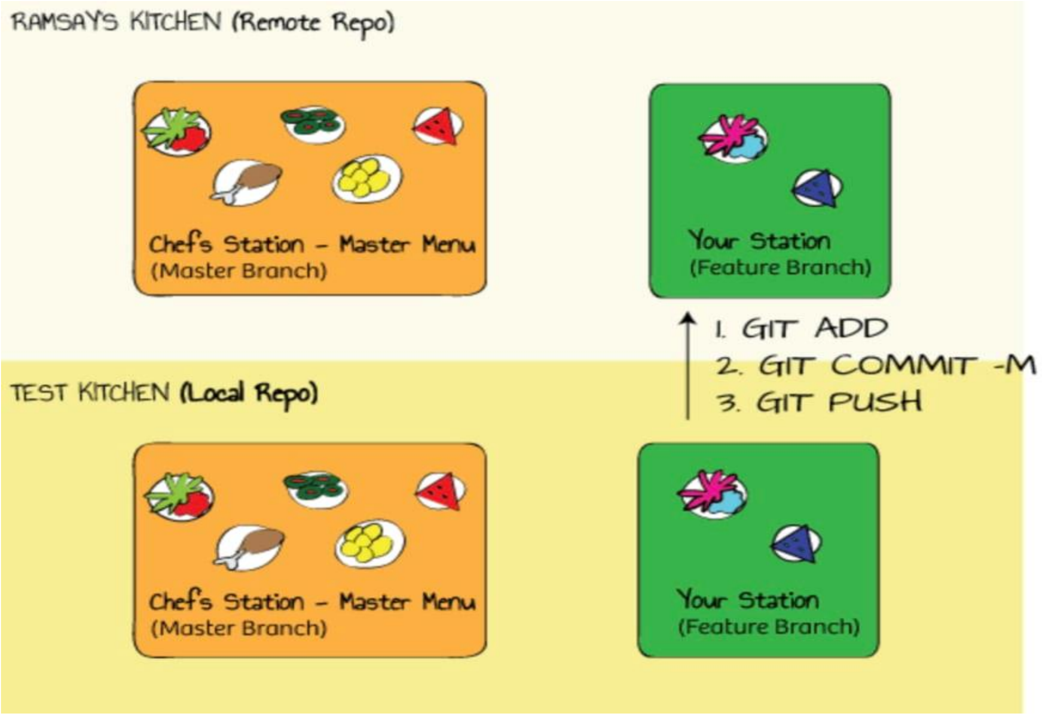
- After modifying your working tree:
  - Two steps to persist these changes in your local repository:
    - add the selected changes to the staging area (also known as index)
      - git add
    - commit the staged changes into the git repository
      - git commit


Working tree
(contains the files you work with and do changes)
Staging area
Git repository
Step 1
Step 2
Add selected changes to the index
Commit staged changes to repository

Source: Google image

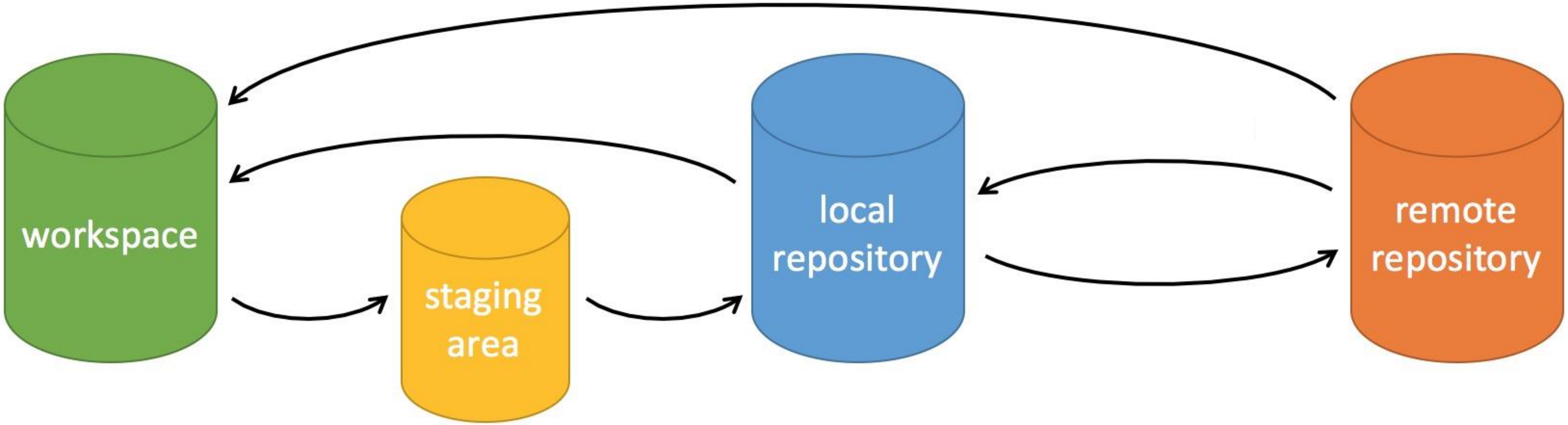
Branch in git

- After updating your remote branch:
  - Pull request for the responsible person (e.g., owner) to merge your changes to the main repository.



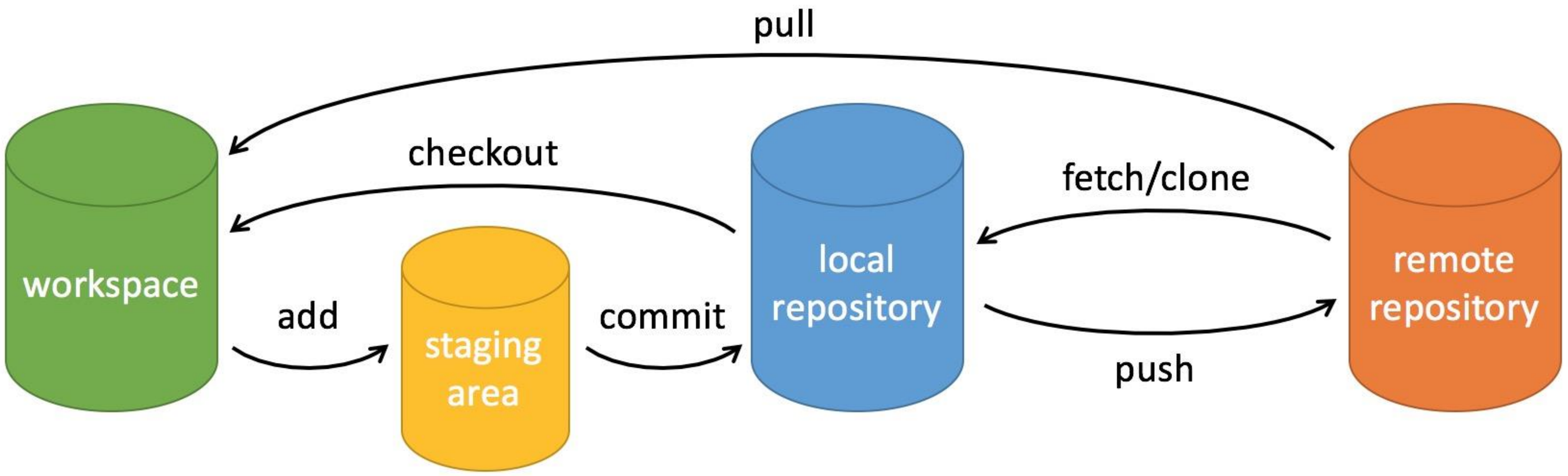
Source: Google image

Git main process at one glance



Source: Google image

Git main process at one glance

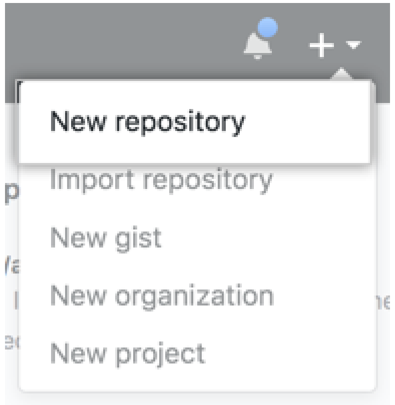


Start with git

- You can create a git repository without any remote counterpart
  - Create a folder
  - Open command prompt
  - Go to the direction of your folder
  - “**git init**”
    - From that moment, the specific folder is a repository

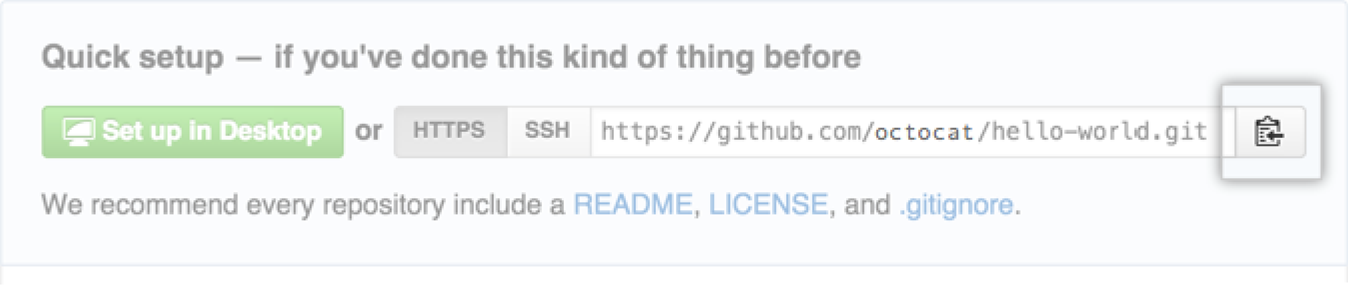
First step, creating a git repository to share with others

- Create a new repository on GitHub.
- To avoid errors, do not initialize the new repository with README, license, or gitignore files.
- You can add these files after your project has been pushed to GitHub.



First step, creating a git repository to share with others

- At the top of your GitHub repository's Quick Setup page, click to copy the remote repository URL.



Quick setup — if you've done this kind of thing before

https://github.com/octocat/hello-world.git

We recommend every repository include a README, LICENSE, and .gitignore.

- In the Command prompt, add the URL from the remote repository where your local repository will be pushed.
  - git remote add origin remote repository URL
- # Sets the new remote
  - git remote –v
- # Verifies the new remote URL

First step, creating a git repository to share with others

- Push the changes in your local repository to GitHub.
  - git push origin master
    - # Pushes the changes in your local repository up to the remote repository you specified as the origin

Get the latest version from the shared git repository

- git clone https://github.com/USERNAME/REPOSITORY.git
  - # Clones a repository to your computer
- git status

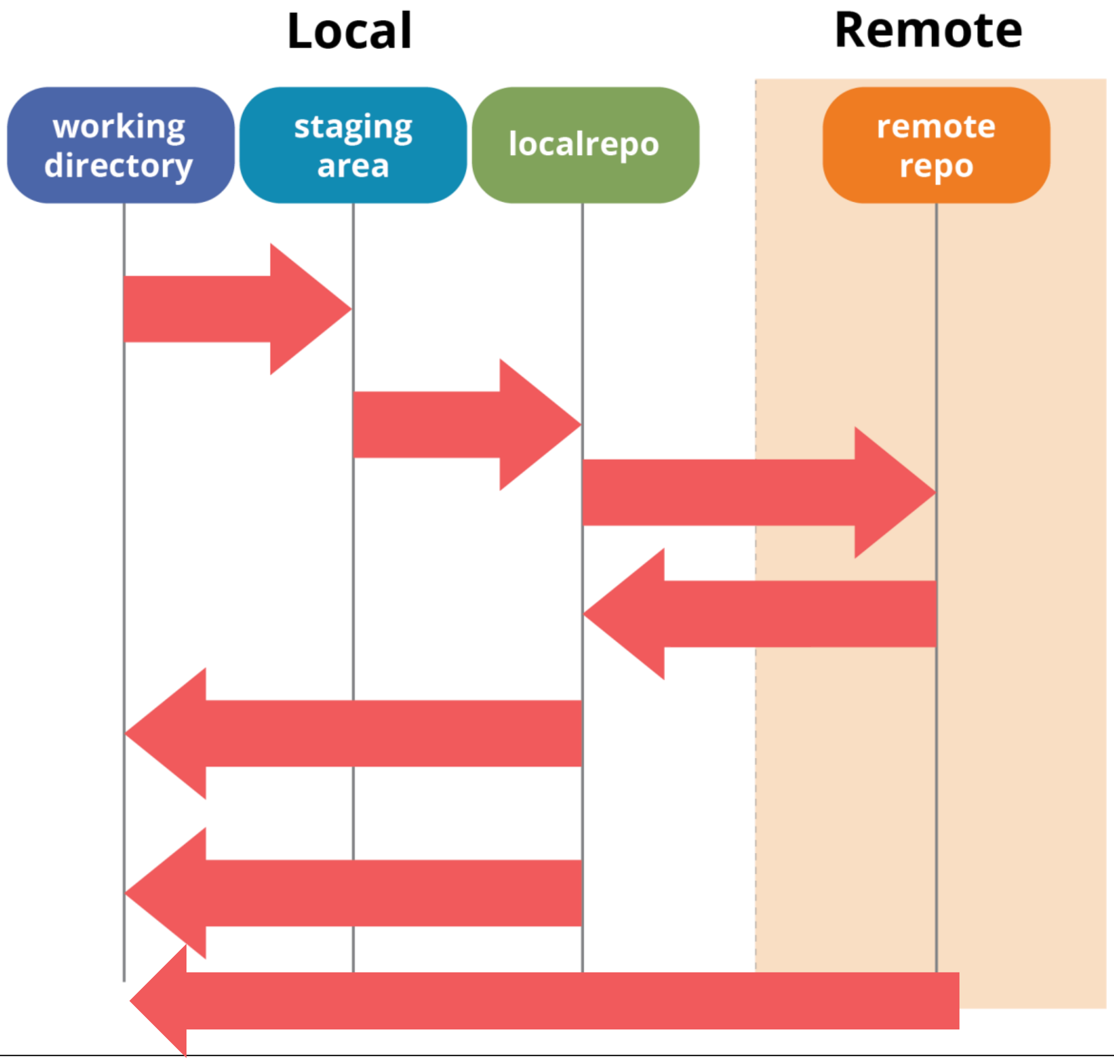
Commit to the git repository

- Git pull
- Git add.
- Git commit –m
- Git push

PULL vs PUSH

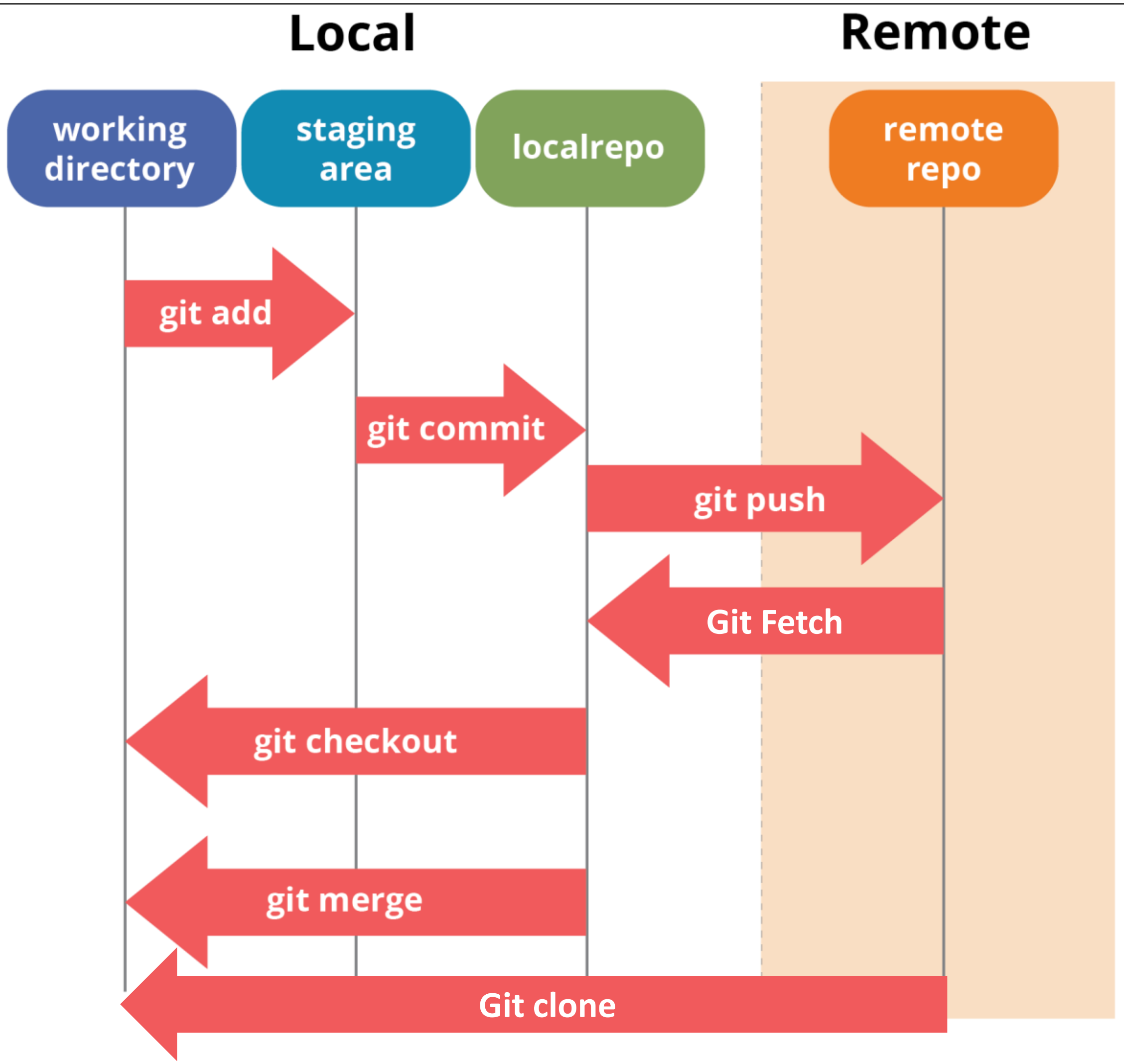
- PULL operation: I take remote commits to my local repository
- PUSH operation: I take local commits and I send them to the remote repository.
- It’s always better to pull before pushing.
- If there are commits in the remote repository that are not in the local repository, the push operation will fail and are reminded to pull.
- When the pull happens, if there are new commits in the remote repository, the “commit” is changed accordingly.

Git important commands



Source: Google image

Git important commands



Important dates

20.11.2020	14:30-16:00	Meeting3	High-quality coding, Unit test,	Mahsa
22.11.2020	23:59	Deadline	Project backlog, user stories, and requirement analysis(10 points) upload via Moodle	