

Version control with Git

NEH Institute 2022

Why version management?

Ever made a change to a file and wanted to go back to a previous situation?

Ever accidentally deleted a file?

Right before a presentation?

Ever had to work with multiple people on the same files?

Why version management? (II)

Safety and accountability for data

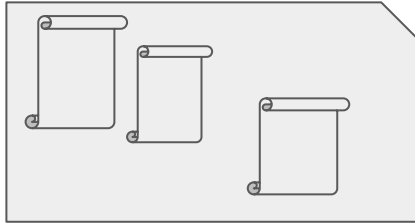
Collaboration

Safety and accountability

Version control allows one to recover from mistakes

Also allows for more experimentation and iteration

Workspace

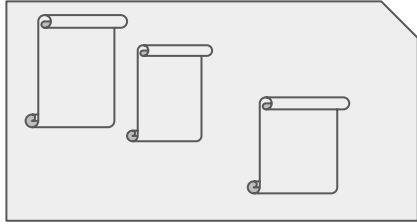


Workspace

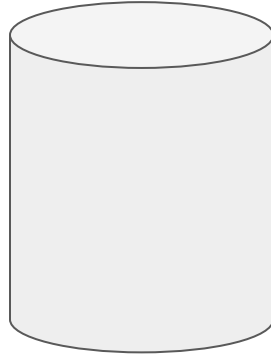
The workspace is a folder on your drive that contains all the files that you want to track and version control.

These are just ordinary files.
Editable with any editor.

Local repository



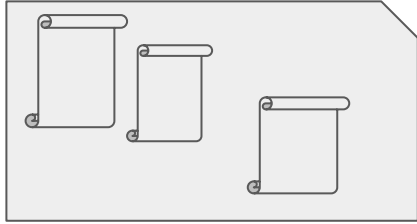
Workspace



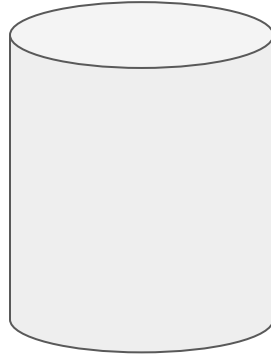
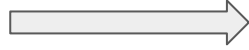
Local repository

Acts as a safe by
keeping copies of
your files at different
points in time

Commit



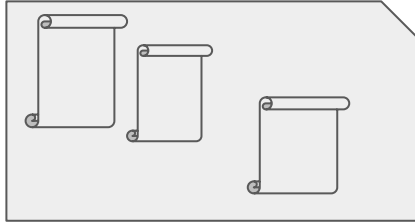
Workspace



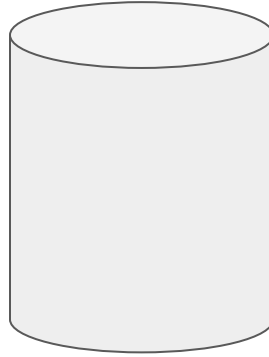
Local repository

Commit: makes a copy of your files and stores it in the local repository

Checkout



Workspace



Local repository

Checkout: restores your files from an earlier state in the repository to the workspace.

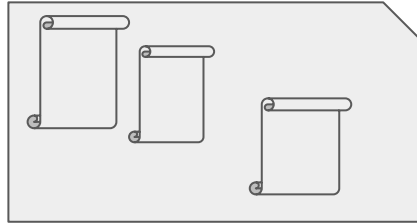
Git commands

Git init	Create a local git repository for a folder
Git status	Get an overview of which files have changed since the last commit
Git add	Add a file or folder to tracking
Git commit	Commit changes made to tracked files to the repository
Git log	Look at a log of commits
Git checkout	Move something from the repository into the workspace
Git rm	Remove a file or folder from tracking
Git mv	Rename a file that is under tracking

Local Workflow

1. Create a repository with **\$ git init**
2. Create one or more files with your favorite editor
3. Track the files with **\$ git add a.xml b.txt**
4. Check with **\$ git status**
5. Commit the files with **\$ git commit -m "Added files a.xml b.txt"**
6. Check your commits with **\$ git log**
7. Make changes to your files
8. Check with **\$ git status**
9. Commit your changes
\$ git commit -m "Added insightful comments to a.xml"
10. Repeat

Overview

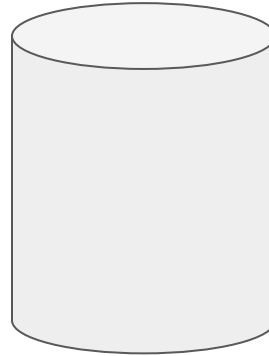


Workspace

Commit



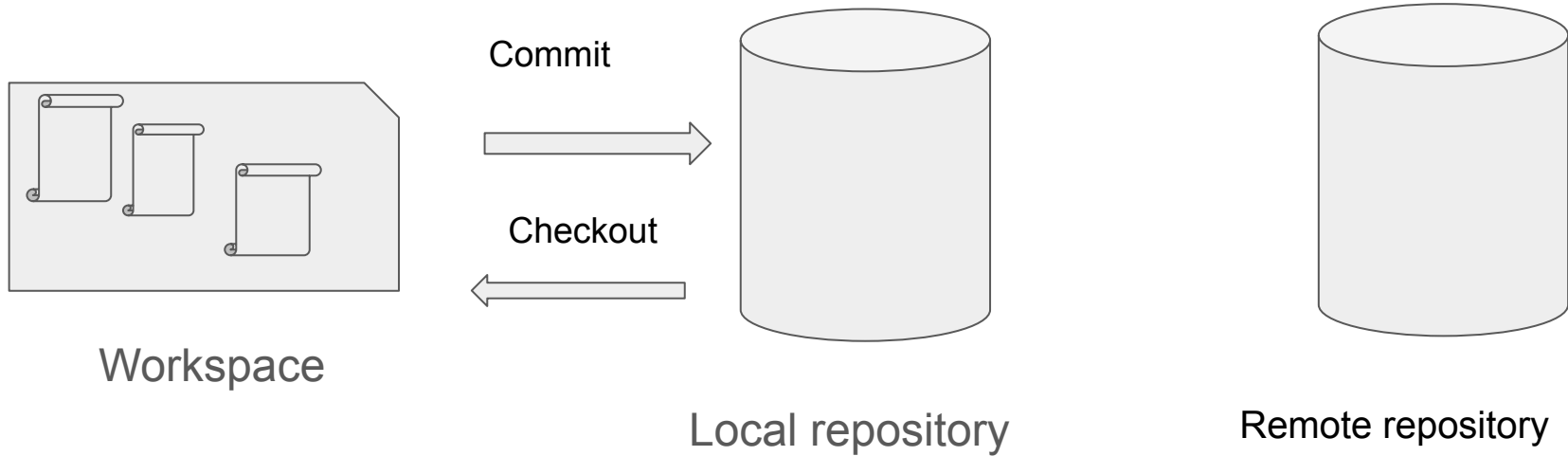
Checkout



Local repository

Pause

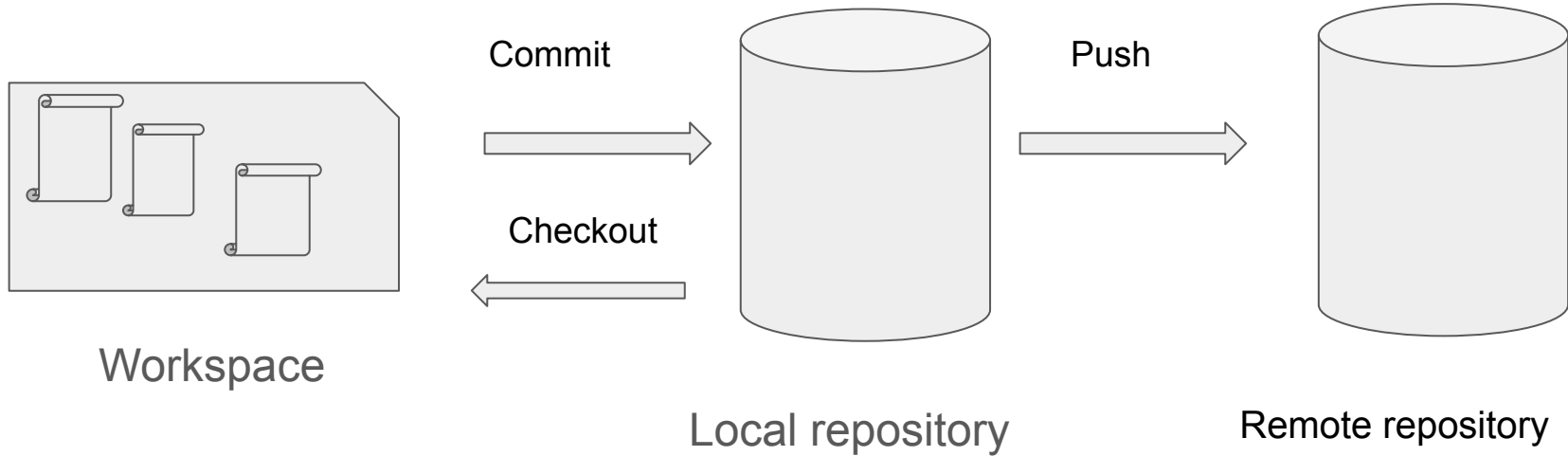
Introducing the remote repository



Remote repository is a safe, a backup for your local machine

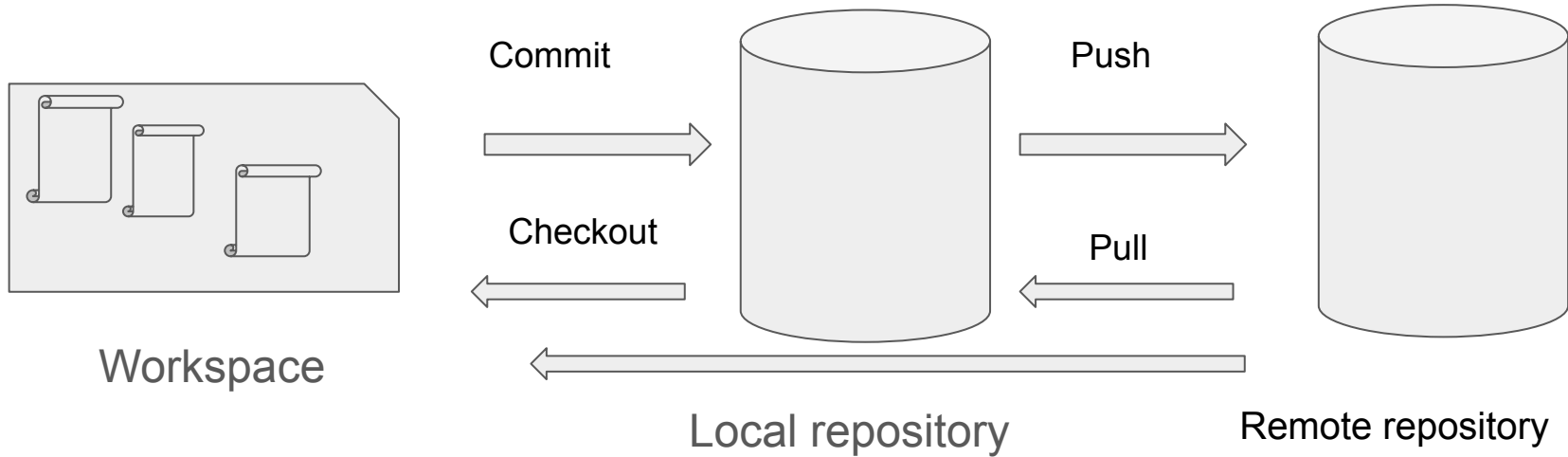
And a way to collaborate with others

Push commits from local to remote



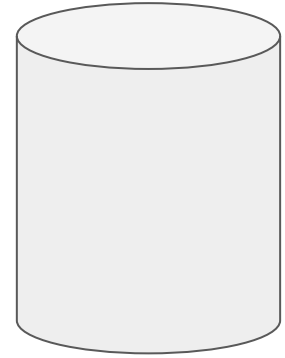
Push brings local commits to the remote repository. Make sure that you are in sync with the remote repository.

Pull from remote repository



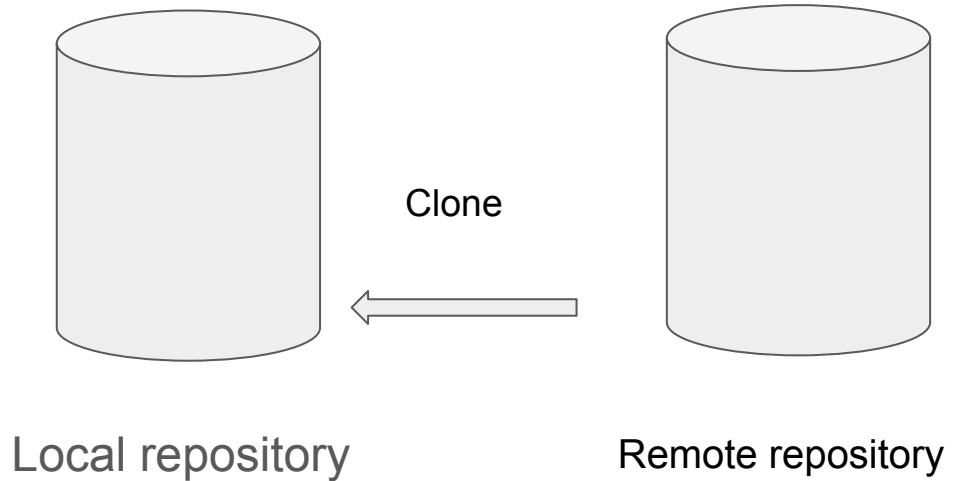
Pull does two things: syncs the local repository with the remote repository and changes the files in the local workspace. Make sure not to have any uncommitted changes before a doing a pull.

What if the remote repository is not started by you?



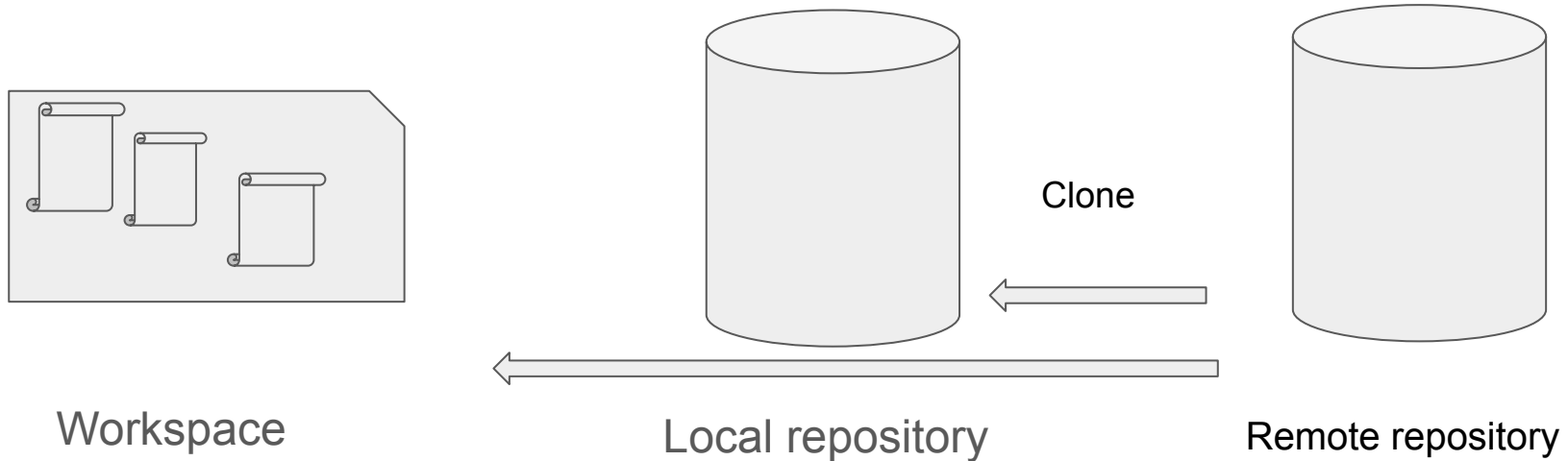
Remote repository

What if the remote repository is not started by you? (II)



Git clone makes a copy of the remote repository locally

What if the remote repository is not started by you? (III)



Cloning does two things: it makes a copy of the remote repository locally and it starts the workspace with a checkout of the files.

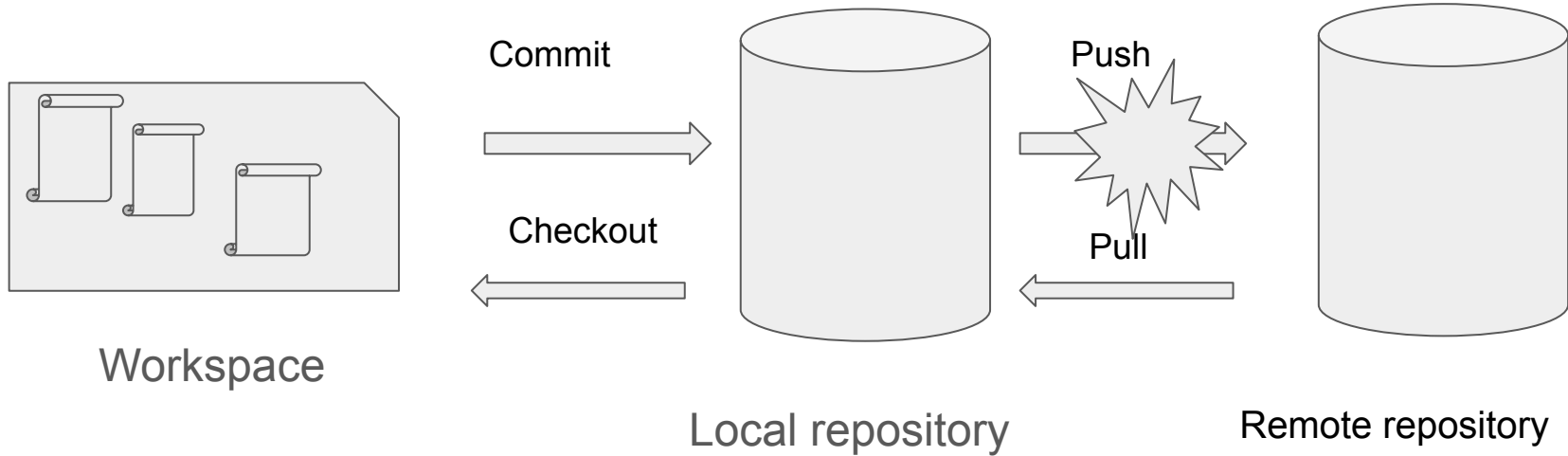
Git commands with remote

- Git clone Clone a remote repository to the local machine and checkout files
- Git push Push commits from local repository to remote repository
- Git pull Pull commits from remote repository to local repository
and workspace

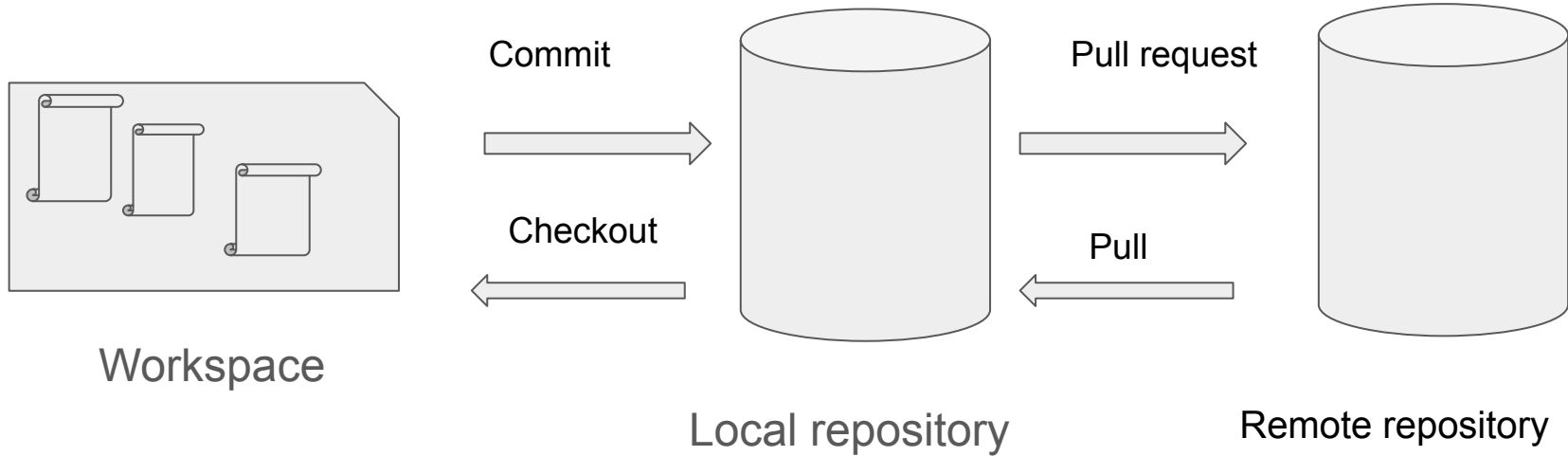
Remote Workflow

1. Clone a repository with **\$ git clone <https address>**
2. Make changes to your files
3. Check with **\$ git status**
4. Commit your changes
\$ git commit -m "Added insightful comments to a.xml"
5. Sync your local repository with possible remote changes with **\$ git pull**
6. Push your changes to remote with **\$ git push**
7. Some time passes...
8. Sync your local repository with possible remote changes with **\$ git pull**
9. Make changes
10. Repeat

What to do if push to remote repository is not allowed?



What to do if push to remote repository is not allowed? (II)

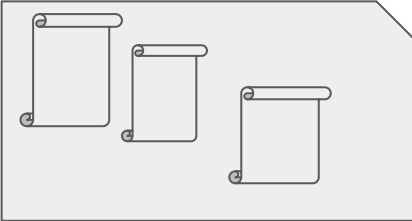


Tips and tricks

What if you are in the process of making changes and there is an update on the remote repository?

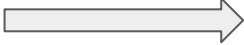
Use Git stash. Git stash sets aside the changes made to the workspace, which allows you to pull the remote changes before continuing to edit.

Git stash

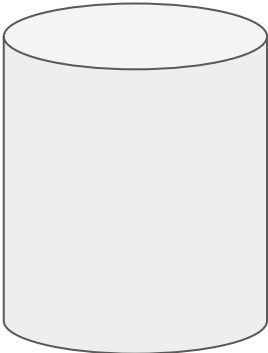


Workspace

Commit



Checkout

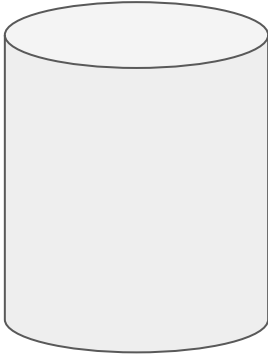


Local repository

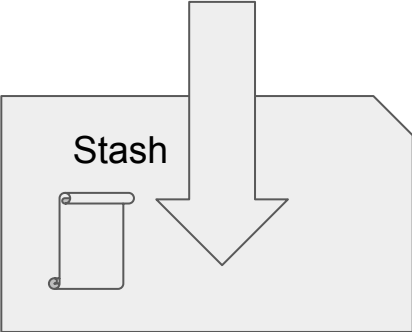
Push / Pull request



Pull



Remote repository



\$ git stash
\$ git pull

Git stash (II)

