ETH zürich

Version control with Git

Best Practices in Programming

Piotr Kupczyk, Swen Vermeul, John Hennig

July 5, 2023

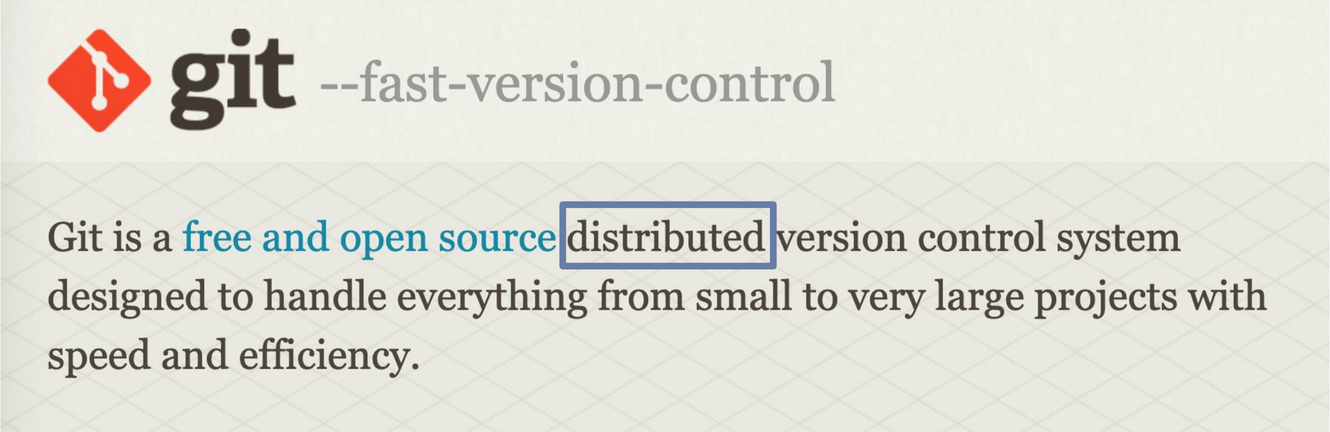
Informatikdienste

What is a Version Control System (VCS)?

*„… is a class of systems responsible for **managing changes** to **computer programs**, documents, large web sites, or other collections of information…”*

[Wikipedia: Version control]

What is Git?



git --fast-version-control

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

[git-scm.com]

Version Control System (VCS)

History of files and folders

History of files and folders

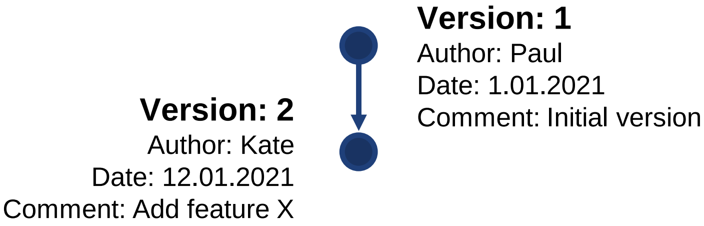
Tracks source code changes over time

(Who? What? When? Why?)

History of files and folders

● **Version: 1**
Author: Paul
Date: 1.01.2021
Comment: Initial version

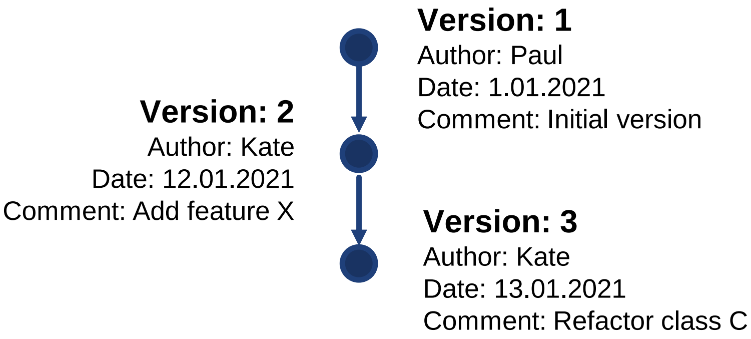
History of files and folders



Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

History of files and folders

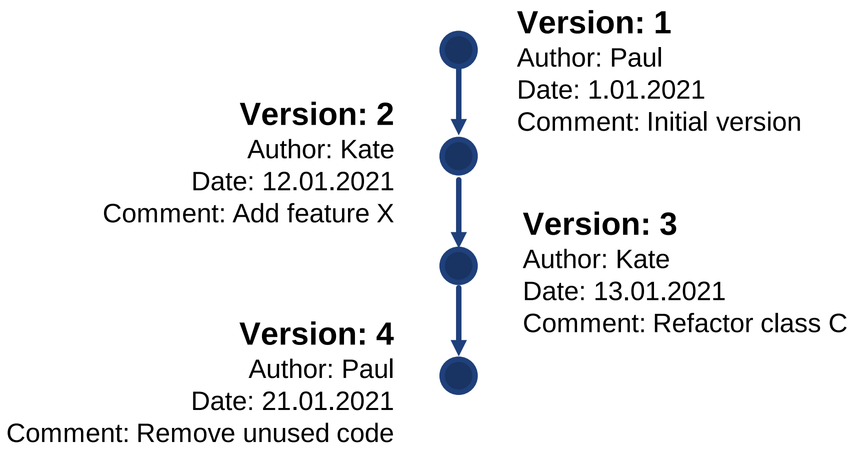


Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 3
Author: Kate
Date: 13.01.2021
Comment: Refactor class C

History of files and folders



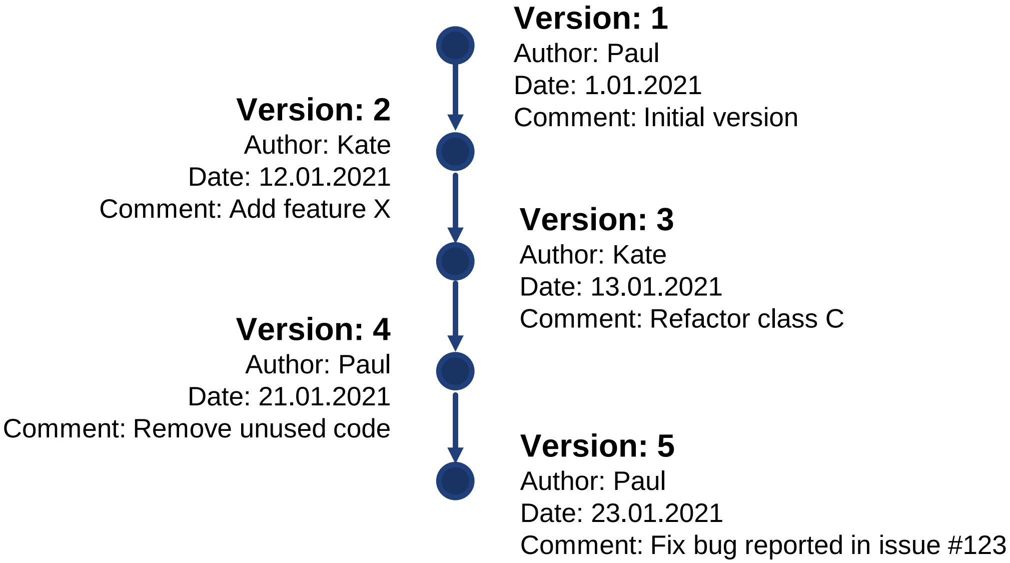
Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 3
Author: Kate
Date: 13.01.2021
Comment: Refactor class C

Version: 4
Author: Paul
Date: 21.01.2021
Comment: Remove unused code

History of files and folders



Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

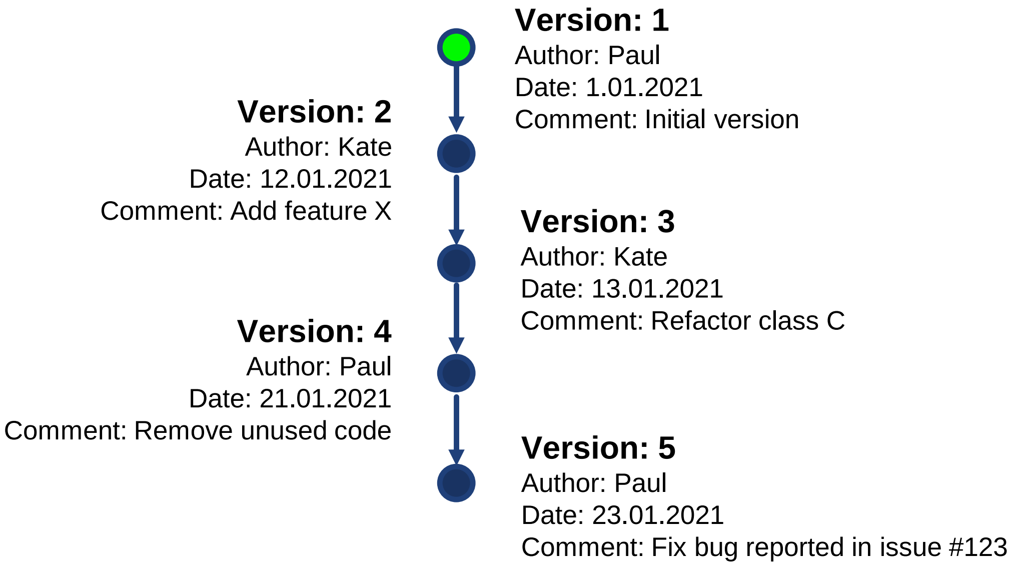
Version: 3
Author: Kate
Date: 13.01.2021
Comment: Refactor class C

Version: 4
Author: Paul
Date: 21.01.2021
Comment: Remove unused code

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Fix bug reported in issue #123

History of files and folders

Good performance



Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

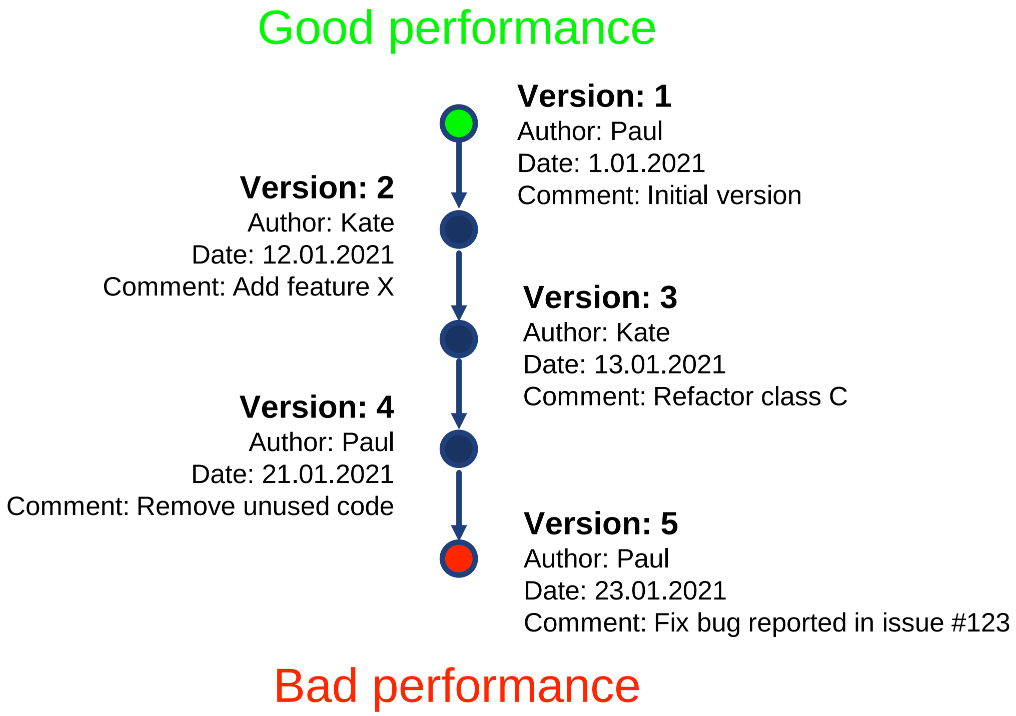
Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 3
Author: Kate
Date: 13.01.2021
Comment: Refactor class C

Version: 4
Author: Paul
Date: 21.01.2021
Comment: Remove unused code

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Fix bug reported in issue #123

History of files and folders



Good performance

Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

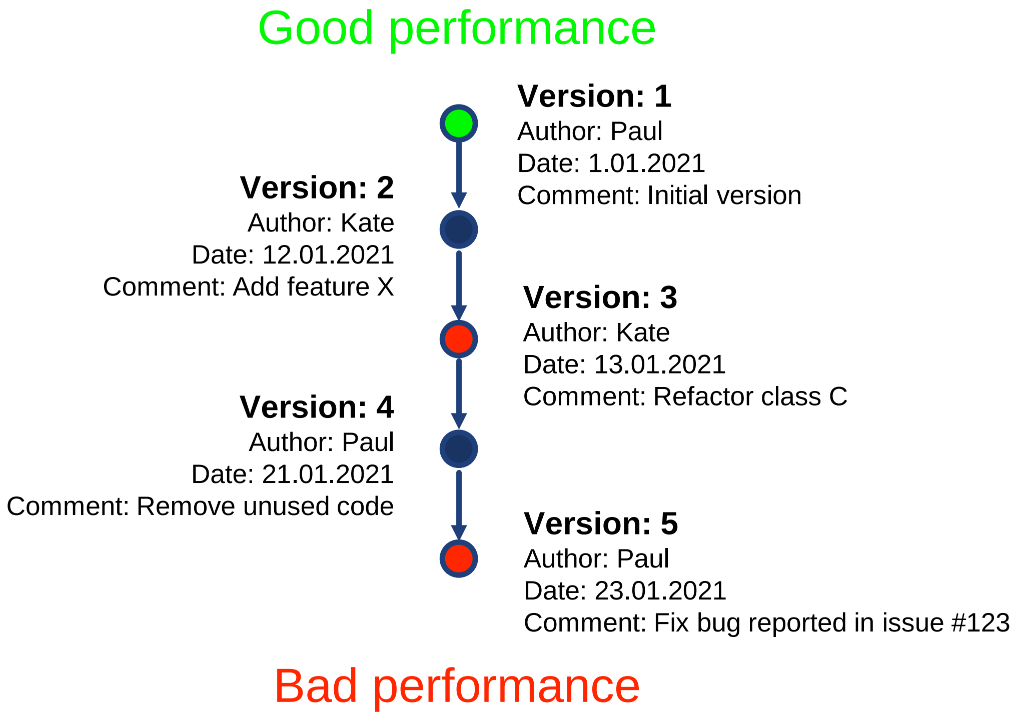
Version: 3
Author: Kate
Date: 13.01.2021
Comment: Refactor class C

Version: 4
Author: Paul
Date: 21.01.2021
Comment: Remove unused code

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Fix bug reported in issue #123

Bad performance

History of files and folders



Good performance

Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 3
Author: Kate
Date: 13.01.2021
Comment: Refactor class C

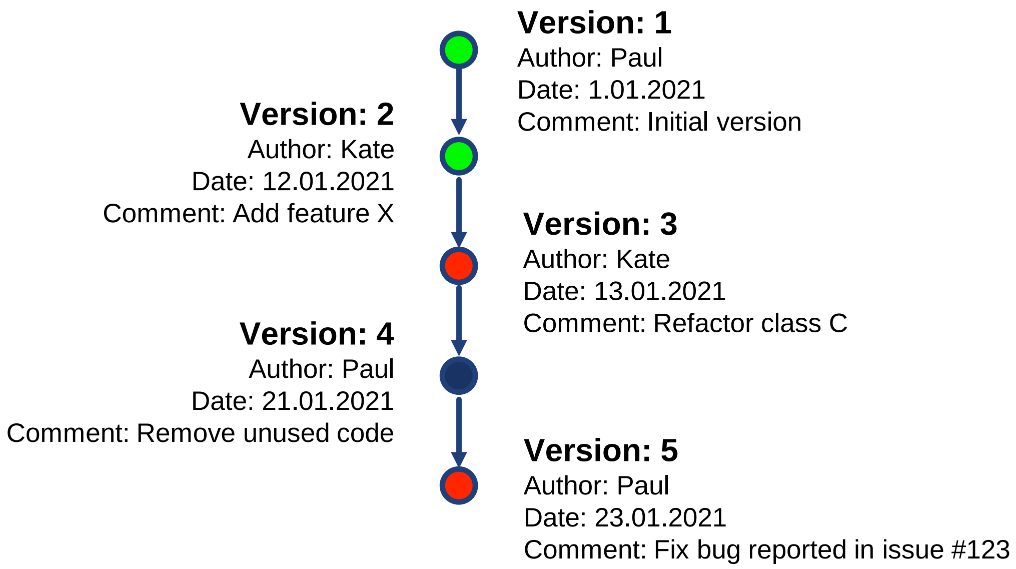
Version: 4
Author: Paul
Date: 21.01.2021
Comment: Remove unused code

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Fix bug reported in issue #123

Bad performance

History of files and folders

Good performance



Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 3
Author: Kate
Date: 13.01.2021
Comment: Refactor class C

Version: 4
Author: Paul
Date: 21.01.2021
Comment: Remove unused code

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Fix bug reported in issue #123

Bad performance

Branches

Branches

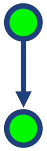
How to work independently of others?

No Branches

● **Version: 1**
Author: Paul
Date: 1.01.2021
Comment: Initial version

Stable version

No Branches

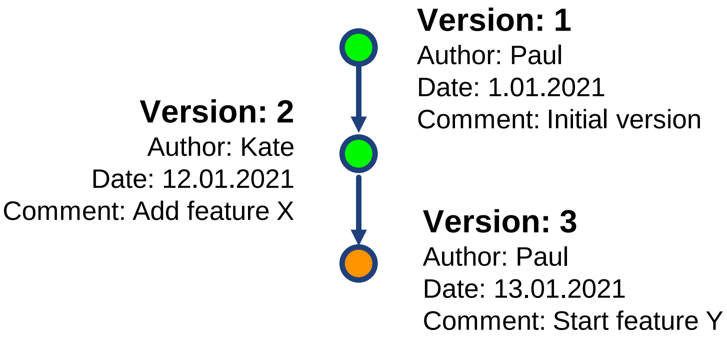


Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Stable version

No Branches



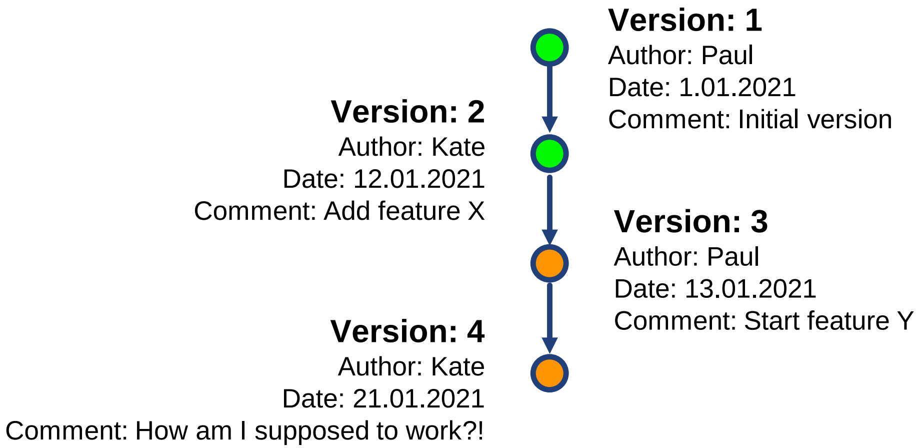
Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Unstable version

No Branches



Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

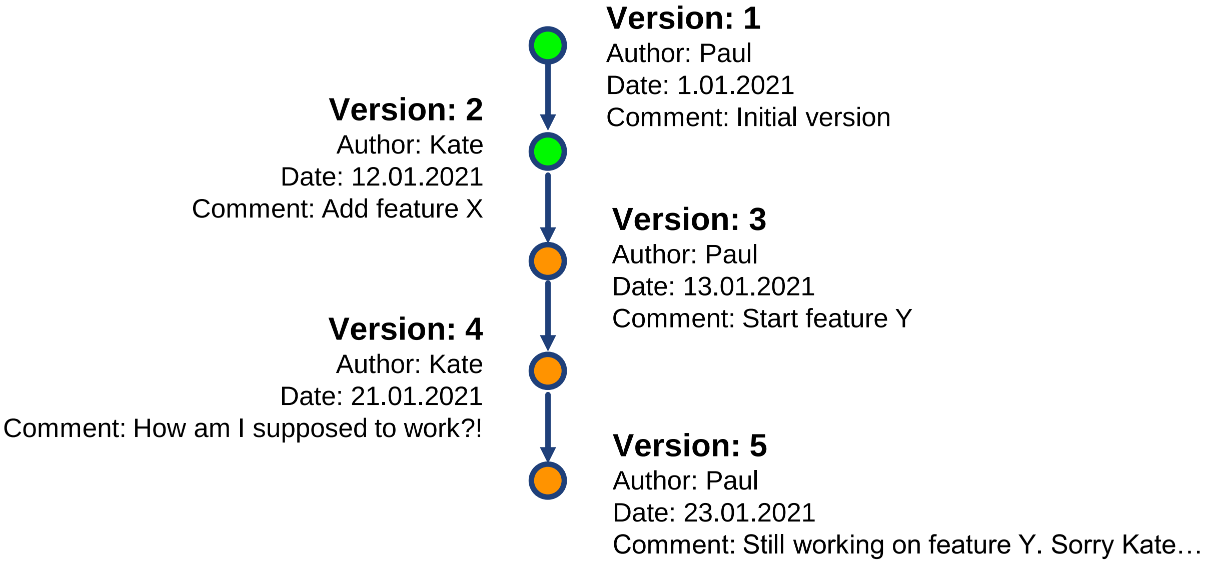
Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Version: 4
Author: Kate
Date: 21.01.2021
Comment: How am I supposed to work?!

Unstable version

No Branches



Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

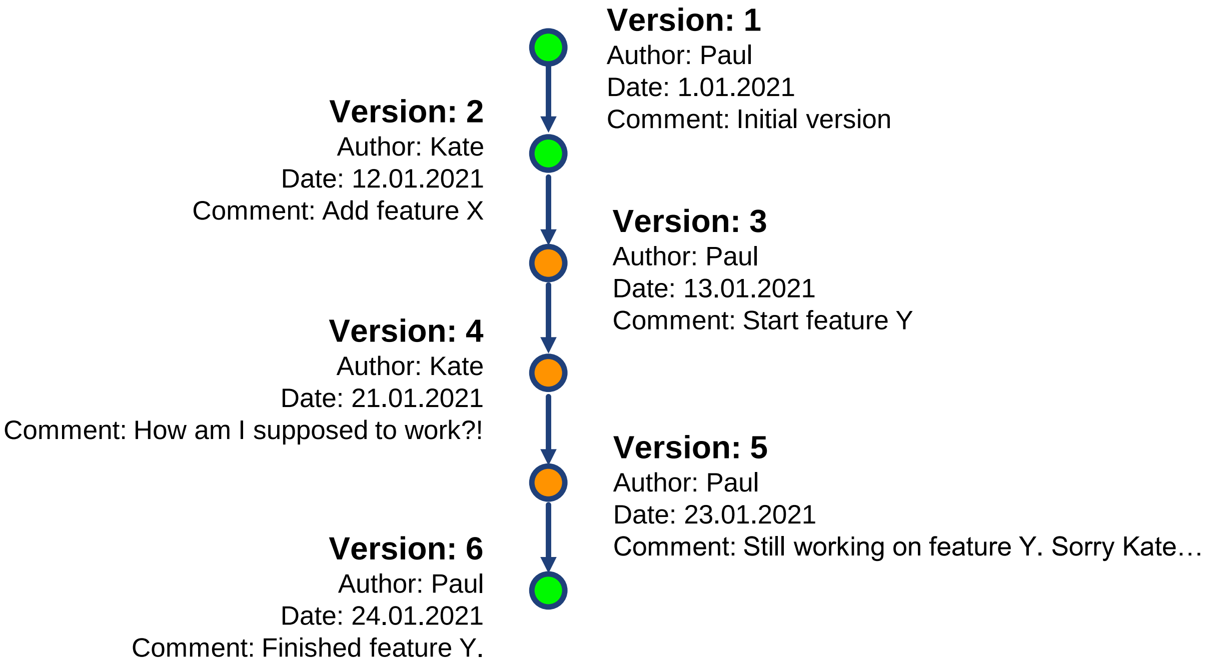
Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Version: 4
Author: Kate
Date: 21.01.2021
Comment: How am I supposed to work?!

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Still working on feature Y. Sorry Kate…

Unstable version

No Branches



Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Version: 4
Author: Kate
Date: 21.01.2021
Comment: How am I supposed to work?!

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Still working on feature Y. Sorry Kate…

Version: 6
Author: Paul
Date: 24.01.2021
Comment: Finished feature Y.

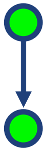
Stable version

With Branches

● **Version: 1**
Author: Paul
Date: 1.01.2021
Comment: Initial version

Stable version

With Branches

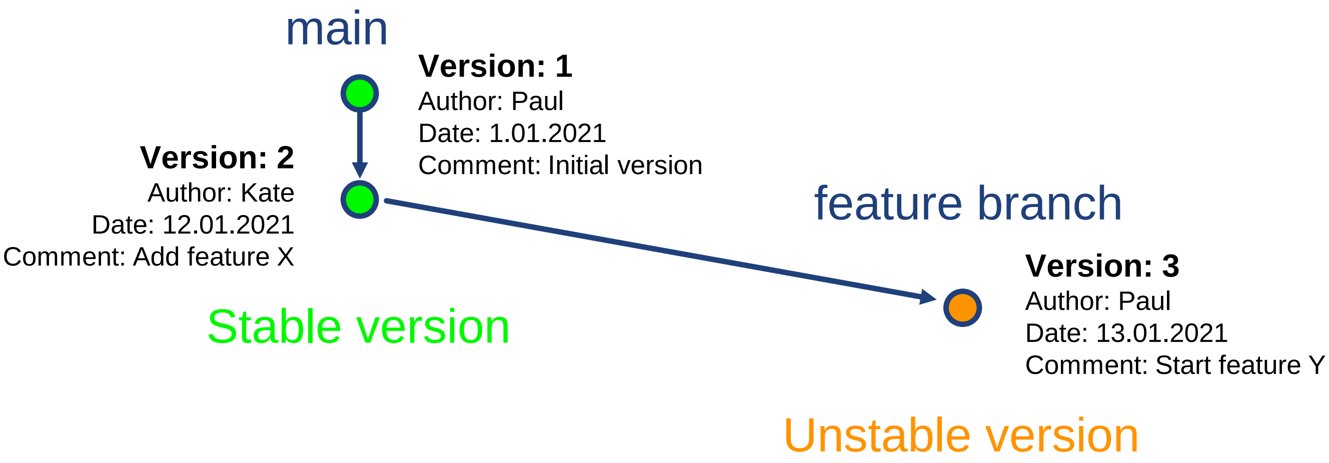


Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Stable version

With Branches



main

Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

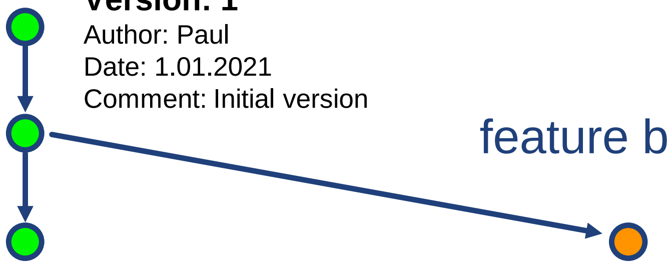
Stable version

feature branch

Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Unstable version

With Branches



main

Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 4
Author: Kate
Date: 21.01.2021
Comment: Wow, I can still work!

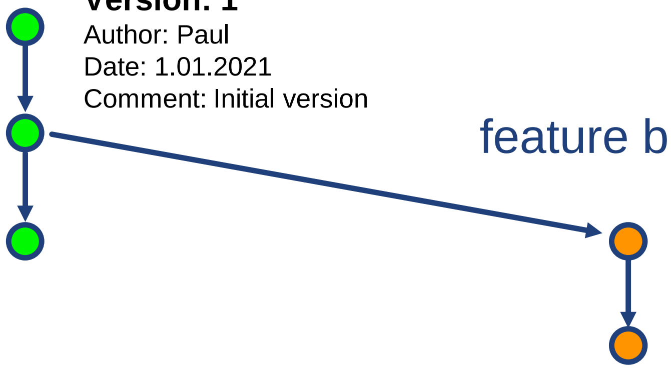
Stable version

feature branch

Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Unstable version

With Branches



main

Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 4
Author: Kate
Date: 21.01.2021
Comment: Wow, I can still work!

Stable version

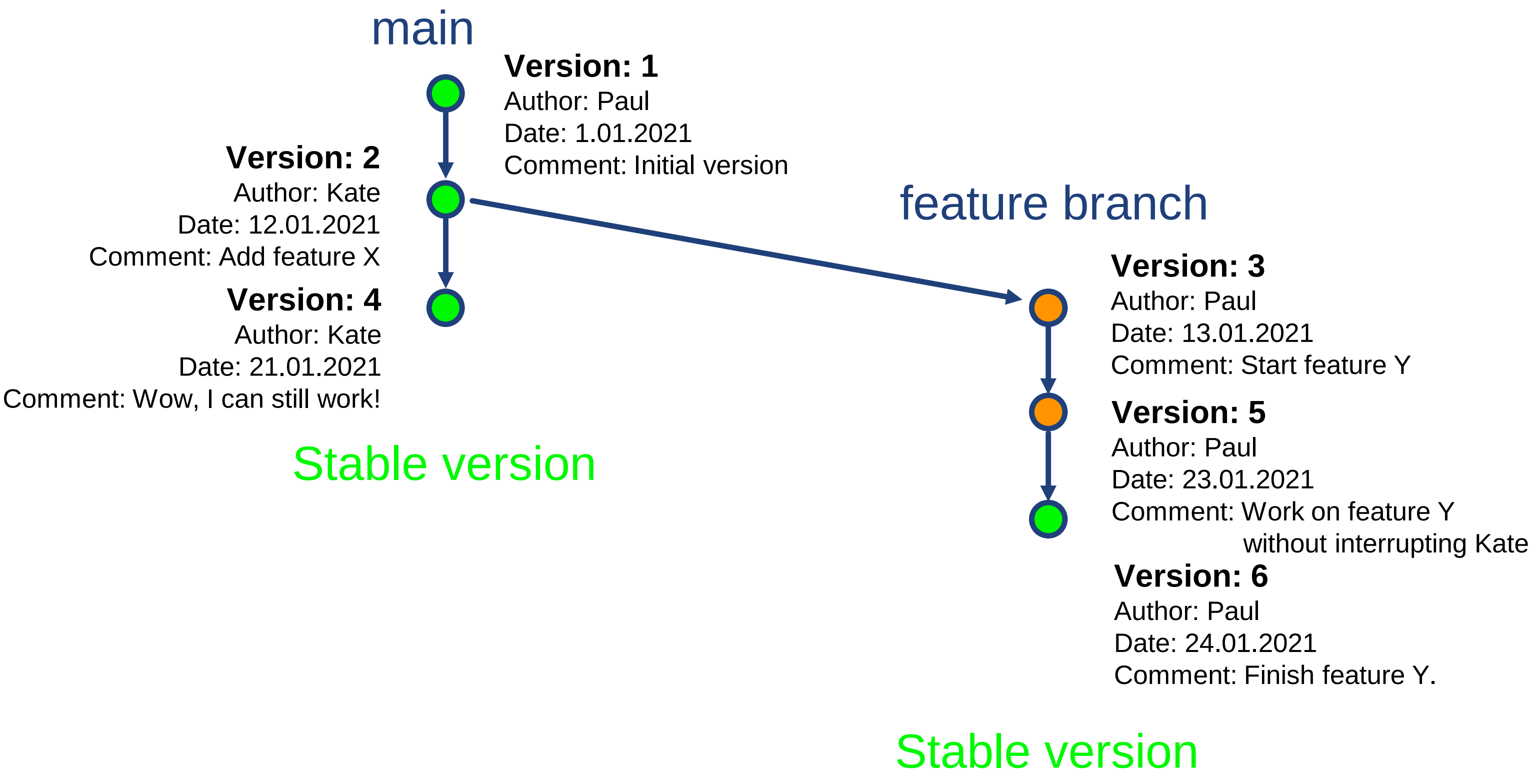
feature branch

Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Work on feature Y
without interrupting Kate

Unstable version

With Branches



main

Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 4
Author: Kate
Date: 21.01.2021
Comment: Wow, I can still work!

Stable version

feature branch

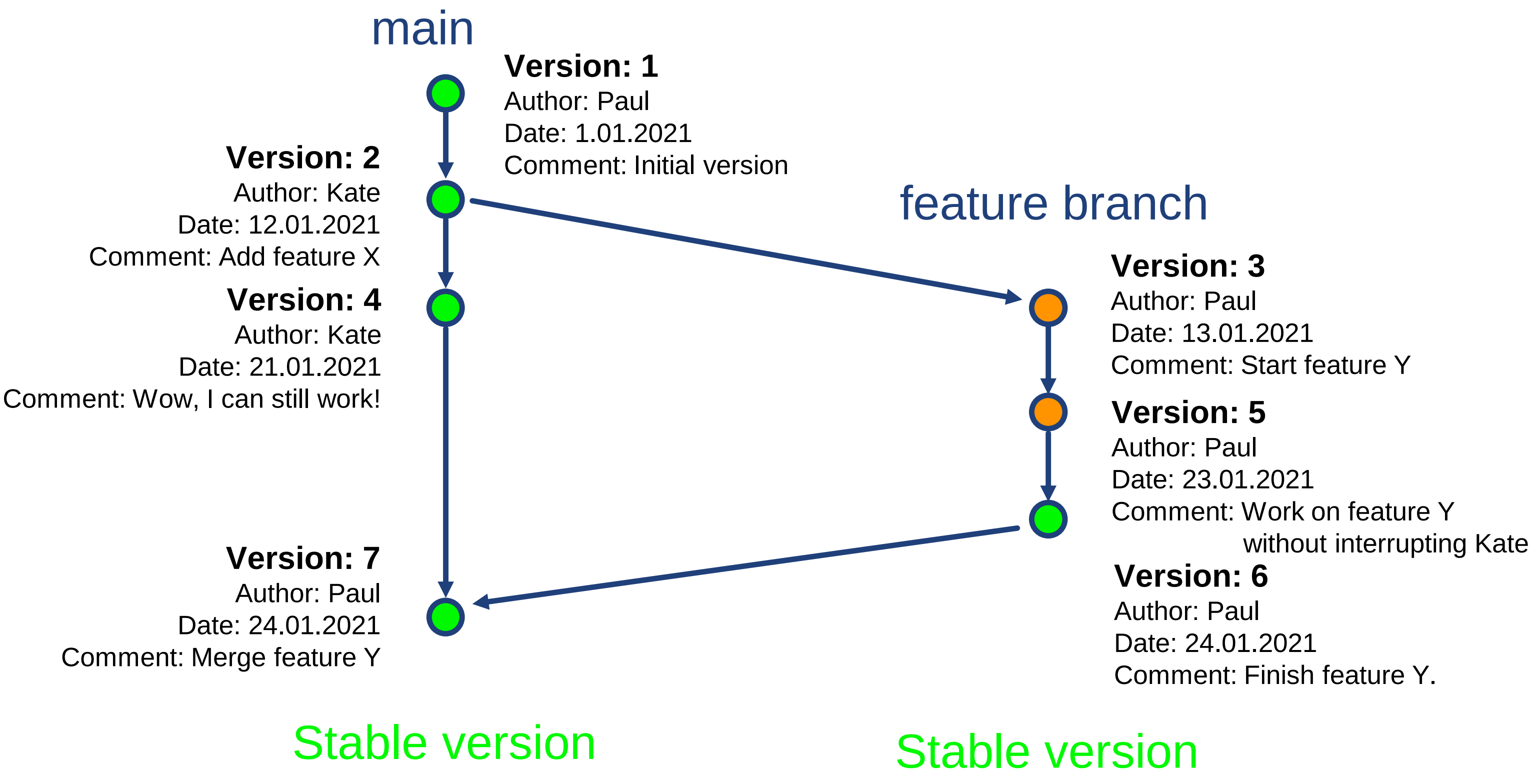
Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Work on feature Y
without interrupting Kate

Version: 6
Author: Paul
Date: 24.01.2021
Comment: Finish feature Y.

Stable version

With Branches



main

Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 4
Author: Kate
Date: 21.01.2021
Comment: Wow, I can still work!

Version: 7
Author: Paul
Date: 24.01.2021
Comment: Merge feature Y

Stable version

feature branch

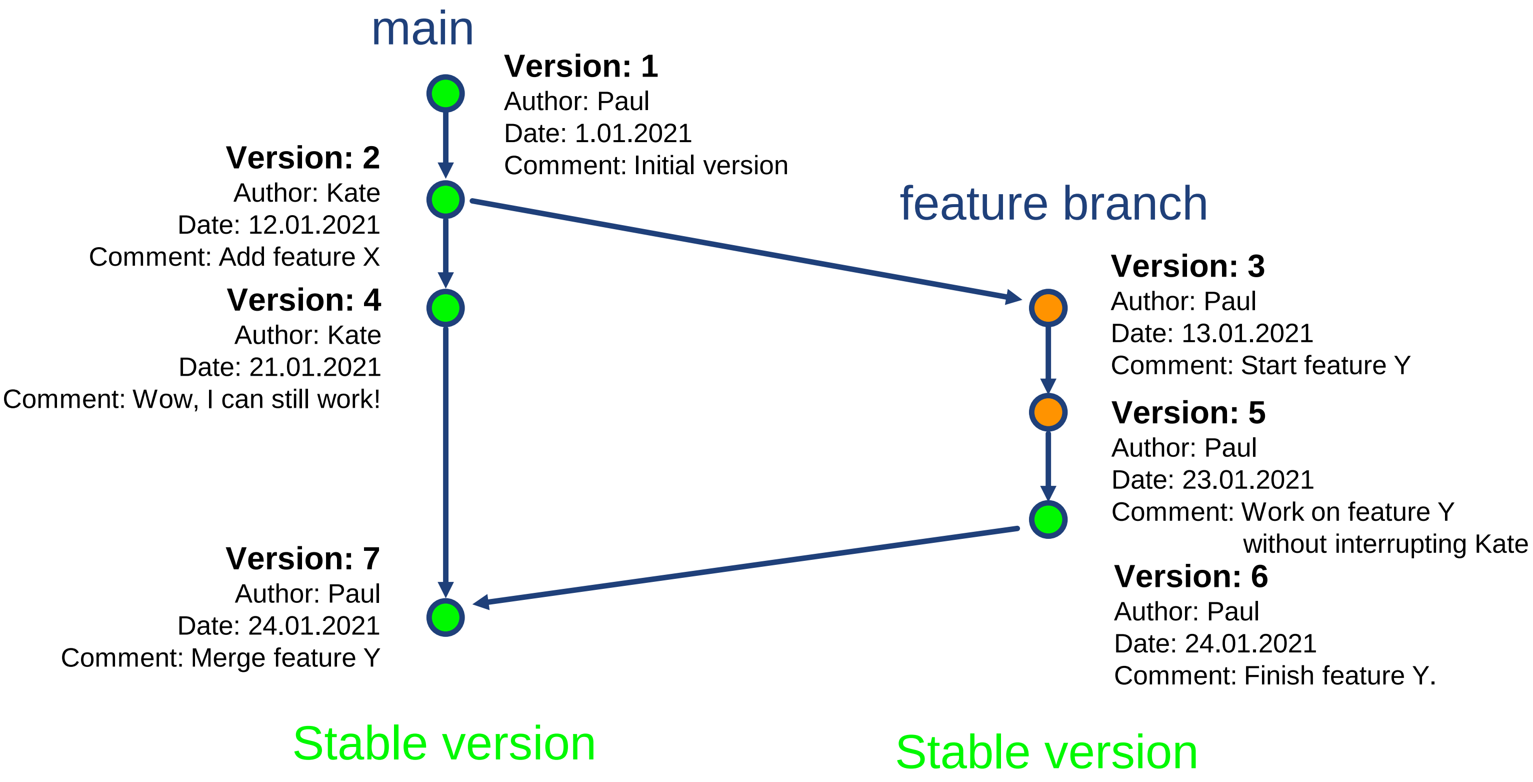
Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Work on feature Y
without interrupting Kate

Version: 6
Author: Paul
Date: 24.01.2021
Comment: Finish feature Y.

Stable version

With Branches



main

Version: 1
Author: Paul
Date: 1.01.2021
Comment: Initial version

Version: 2
Author: Kate
Date: 12.01.2021
Comment: Add feature X

Version: 4
Author: Kate
Date: 21.01.2021
Comment: Wow, I can still work!

Version: 7
Author: Paul
Date: 24.01.2021
Comment: Merge feature Y

feature branch

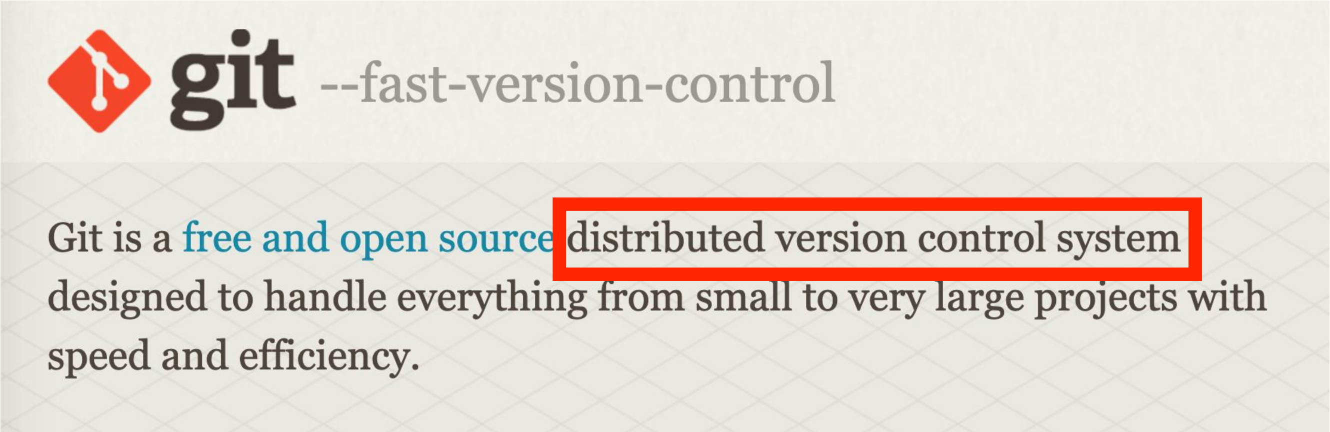
Version: 3
Author: Paul
Date: 13.01.2021
Comment: Start feature Y

Version: 5
Author: Paul
Date: 23.01.2021
Comment: Work on feature Y
without interrupting Kate

Version: 6
Author: Paul
Date: 24.01.2021
Comment: Finish feature Y.

Stable version

Stable version



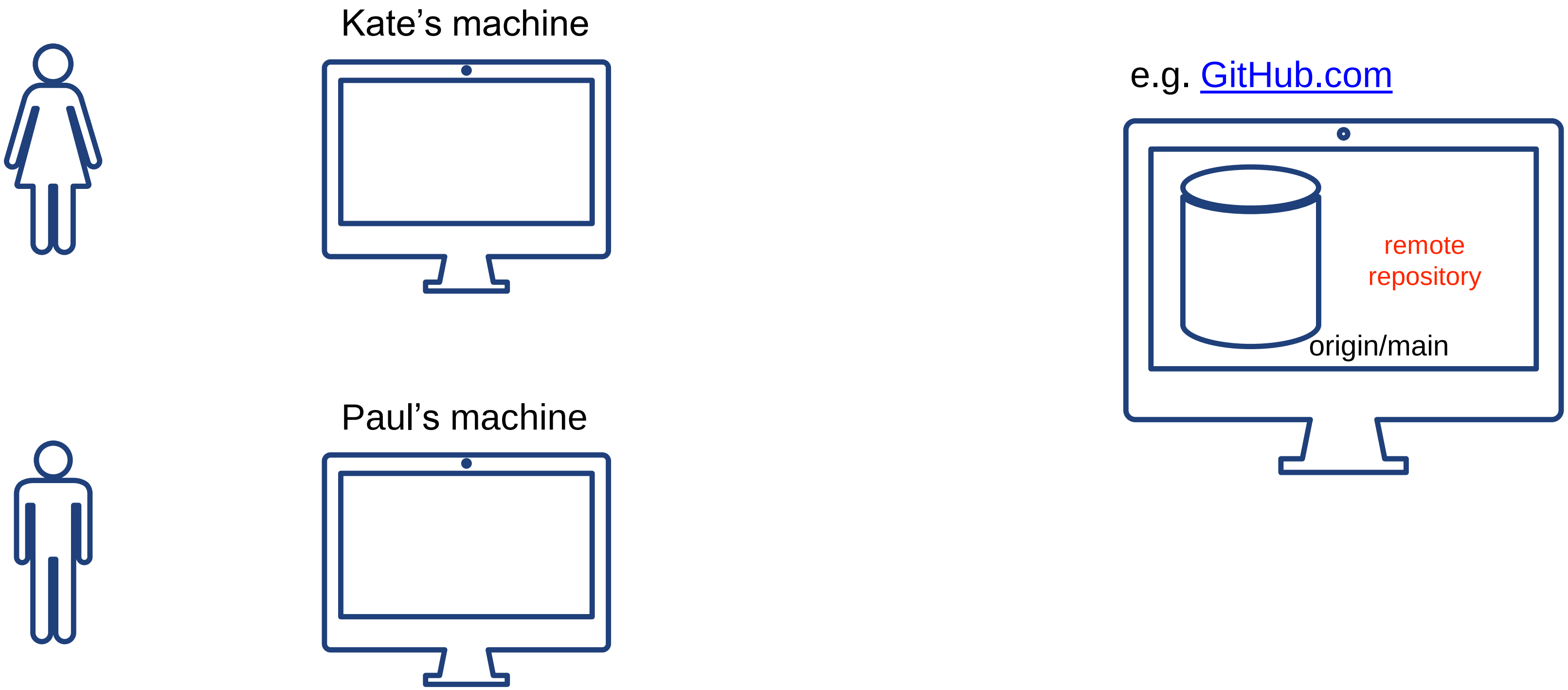
git --fast-version-control

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Collaboration

How to share changes with others?

Collaboration



Kate’s machine

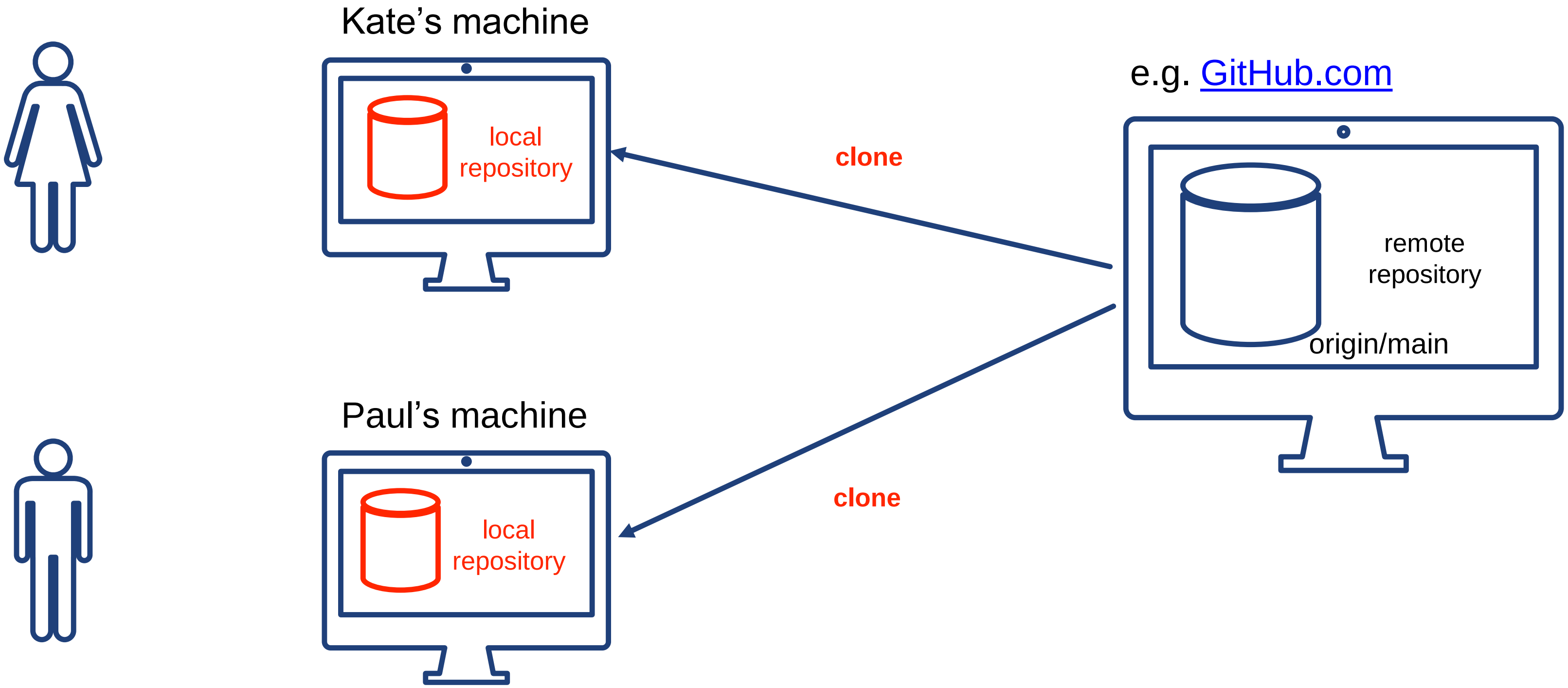
Paul’s machine

e.g. GitHub.com

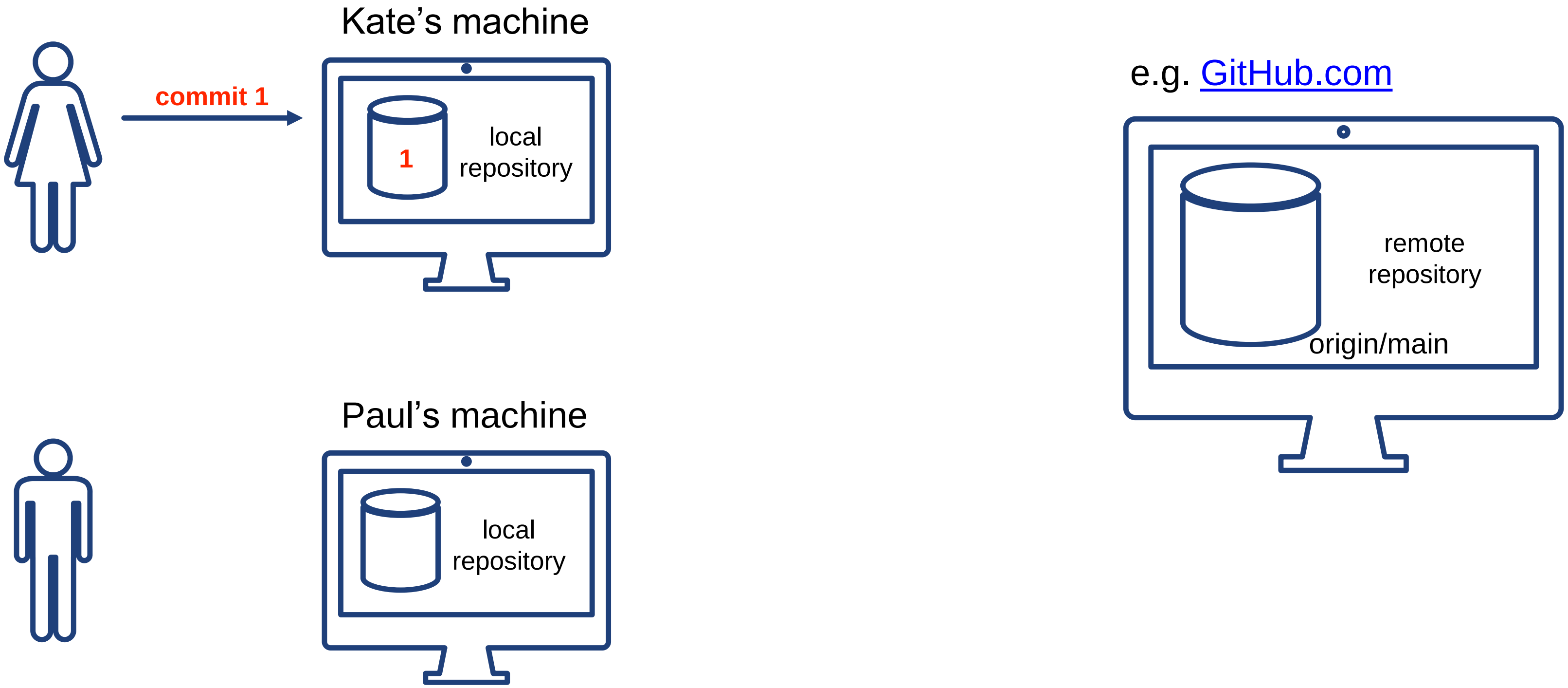
remote repository

origin/main

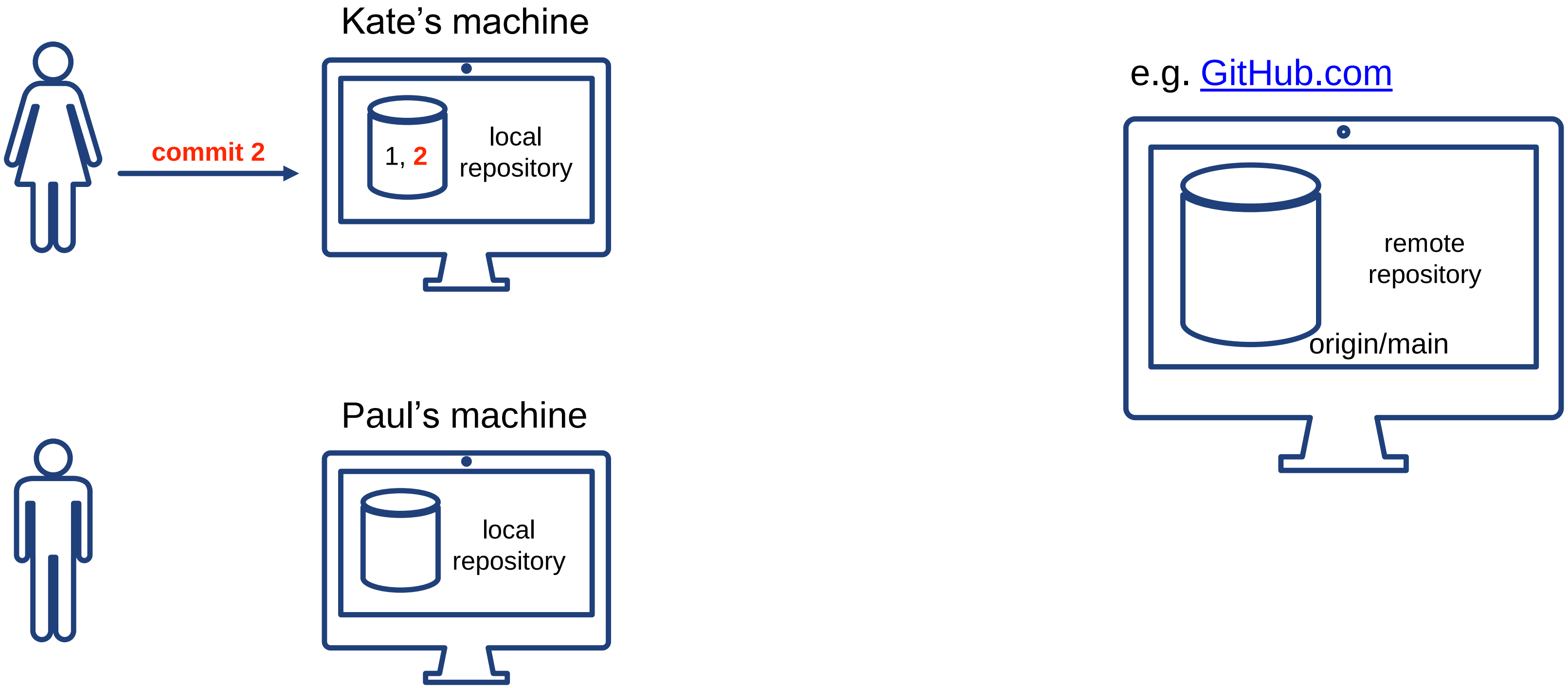
Collaboration



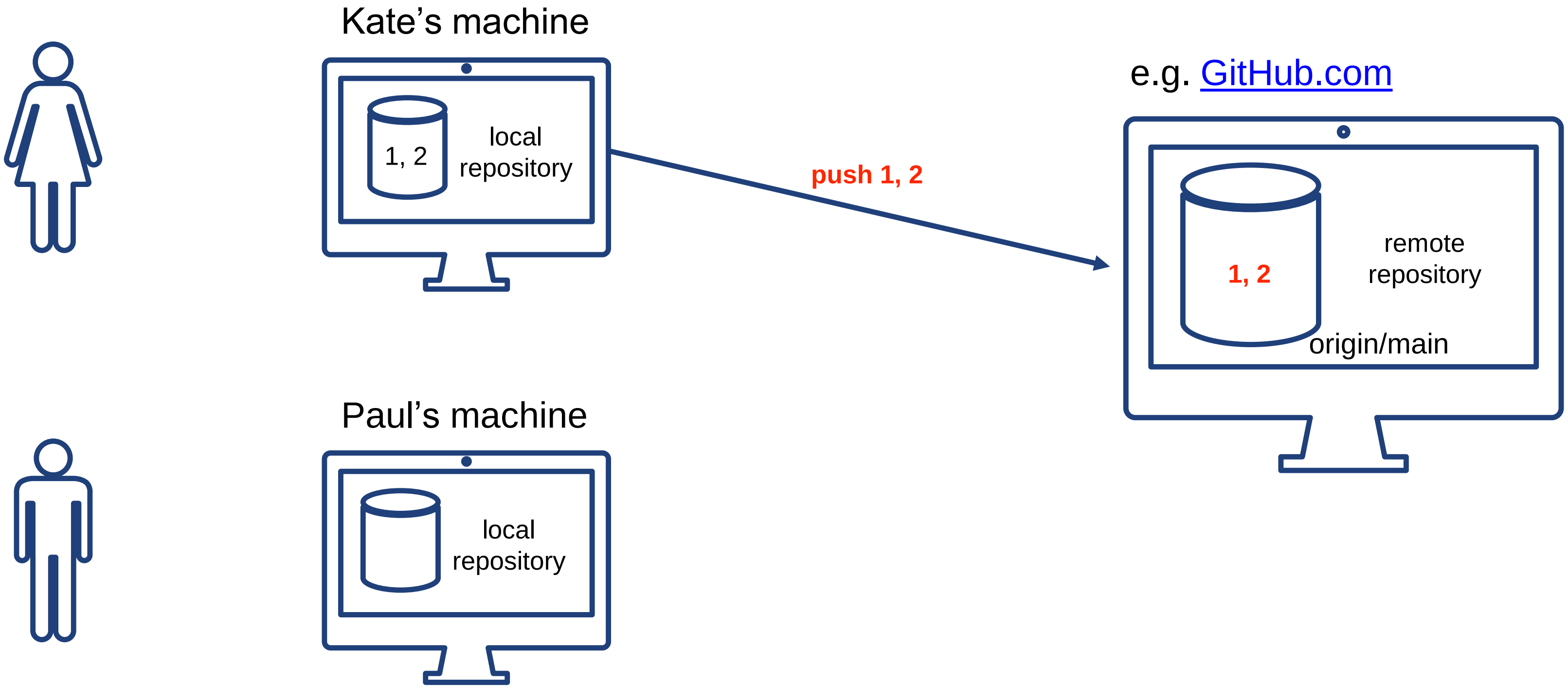
Collaboration



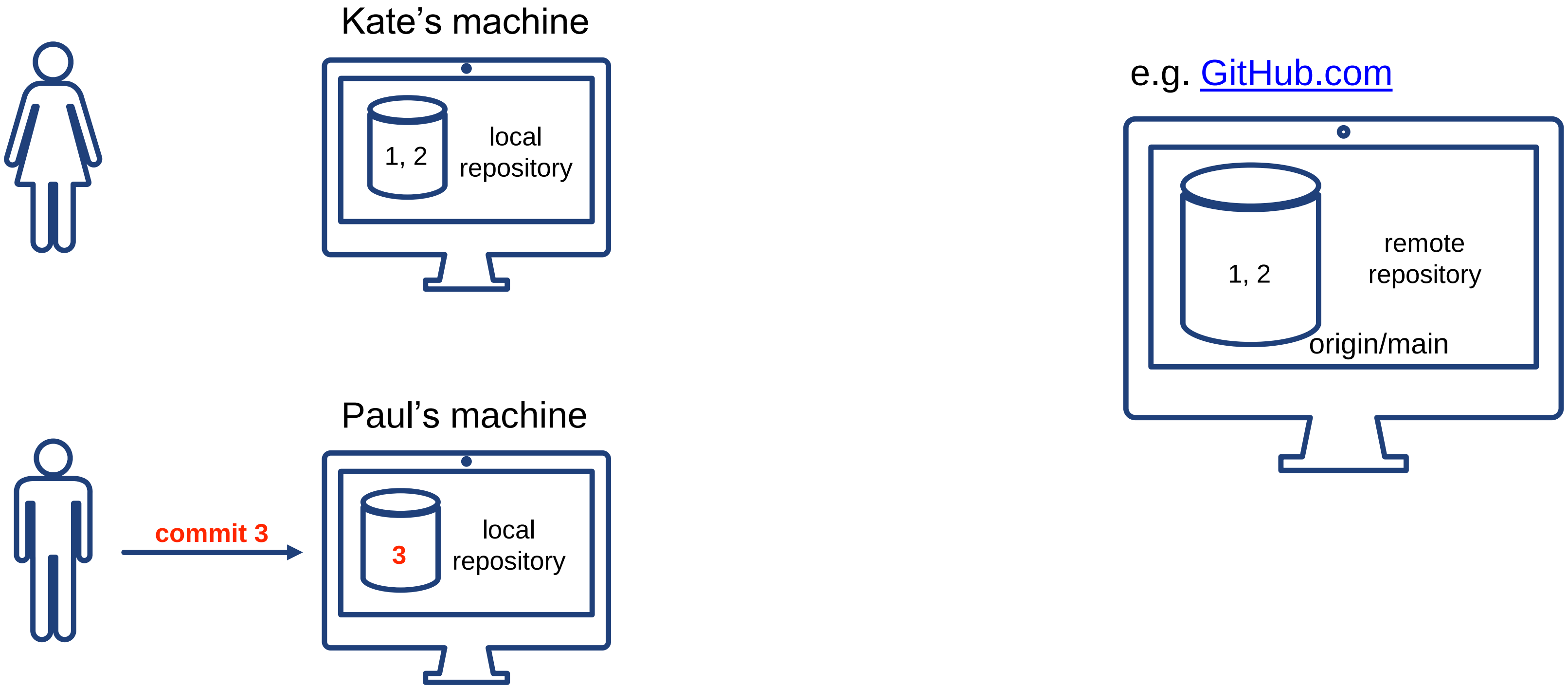
Collaboration



Collaboration



Collaboration



Kate’s machine

1, 2

local repository

Paul’s machine

commit 3

3

local repository

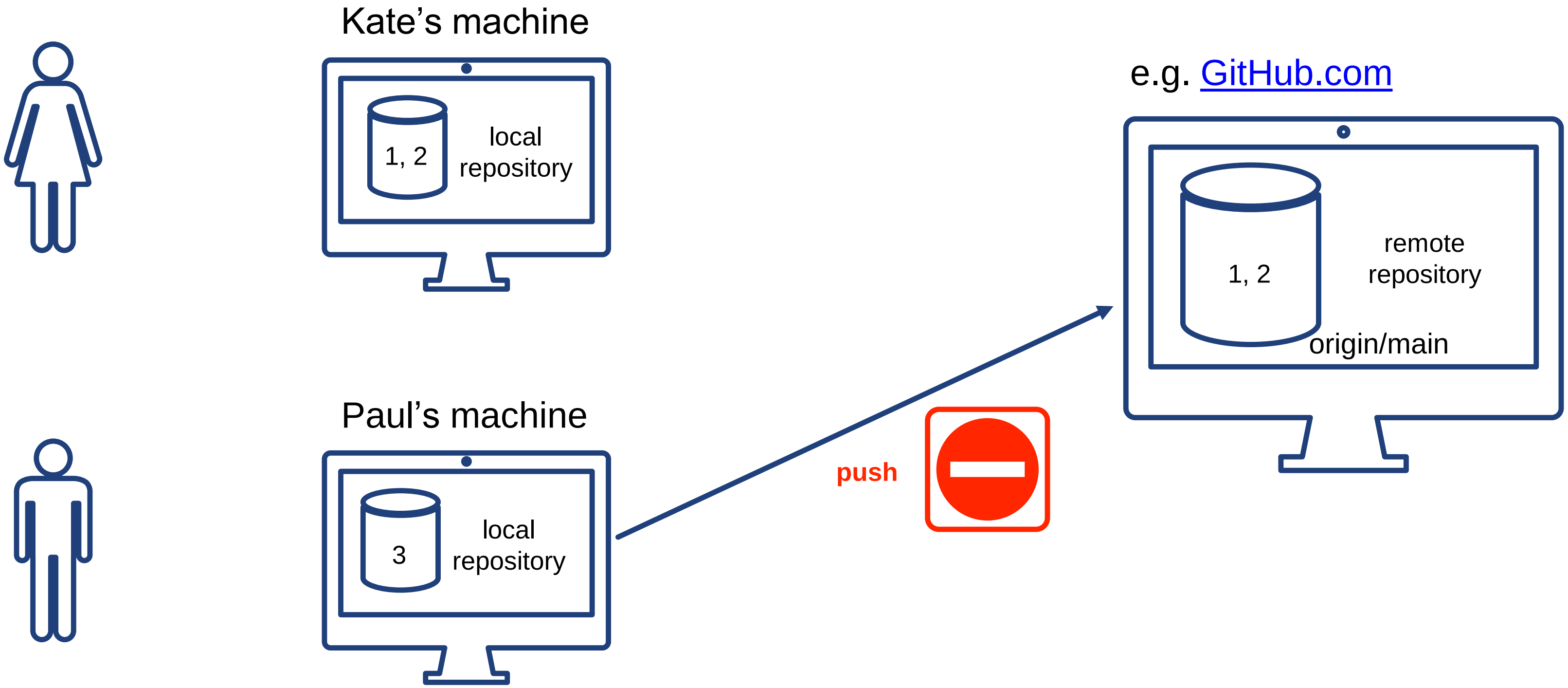
e.g. GitHub.com

1, 2

remote repository

origin/main

Collaboration



Kate’s machine

1, 2

local repository

e.g. GitHub.com

1, 2

remote repository

origin/main

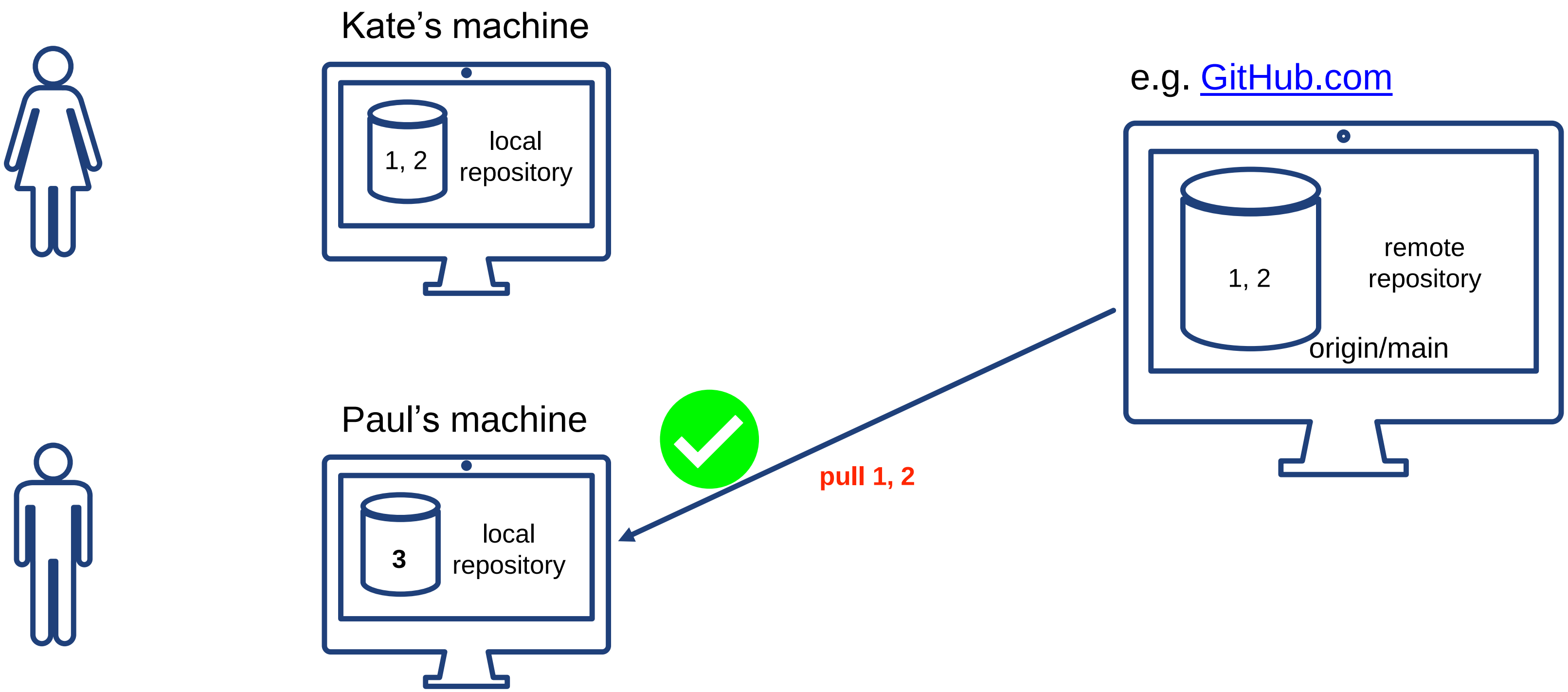
Paul’s machine

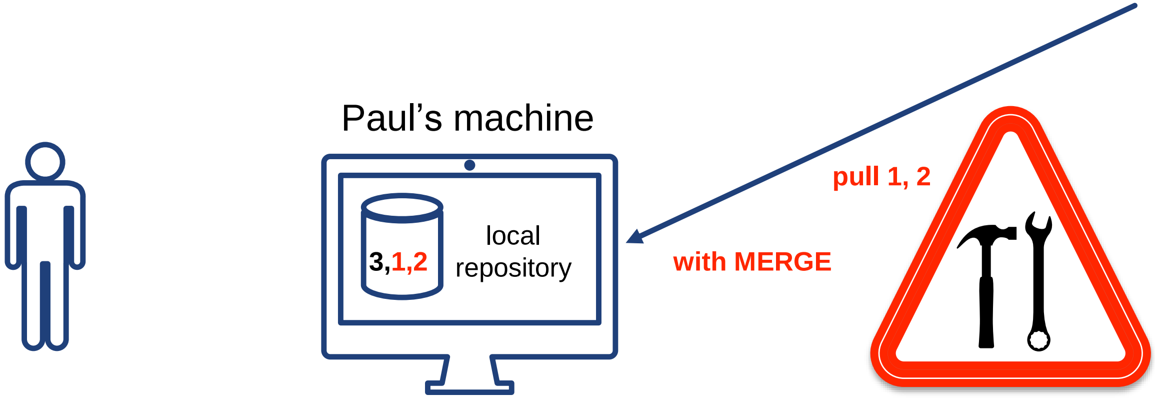
3

local repository

push

Collaboration





Paul’s machine

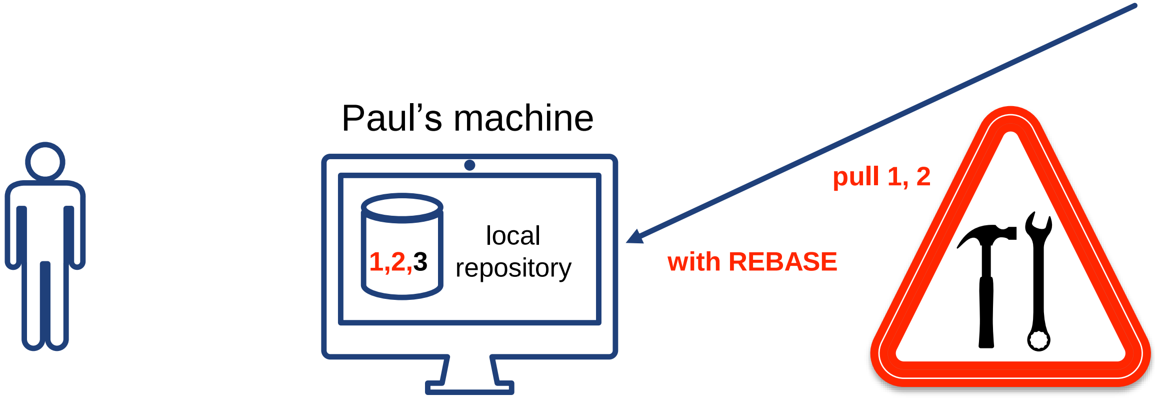
3,1,2

local repository

pull 1, 2

with MERGE

resolve possible conflicts



Paul’s machine

1,2,3

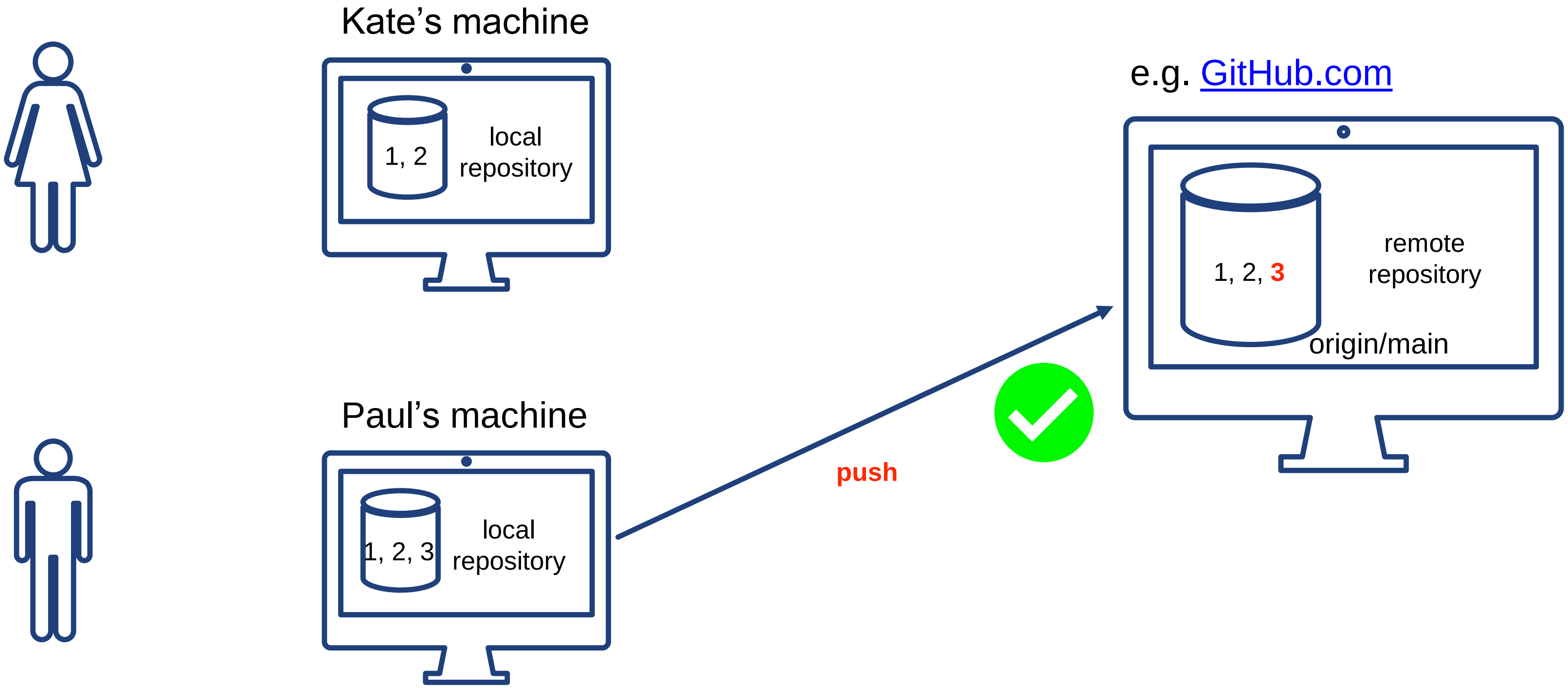
local repository

pull 1, 2

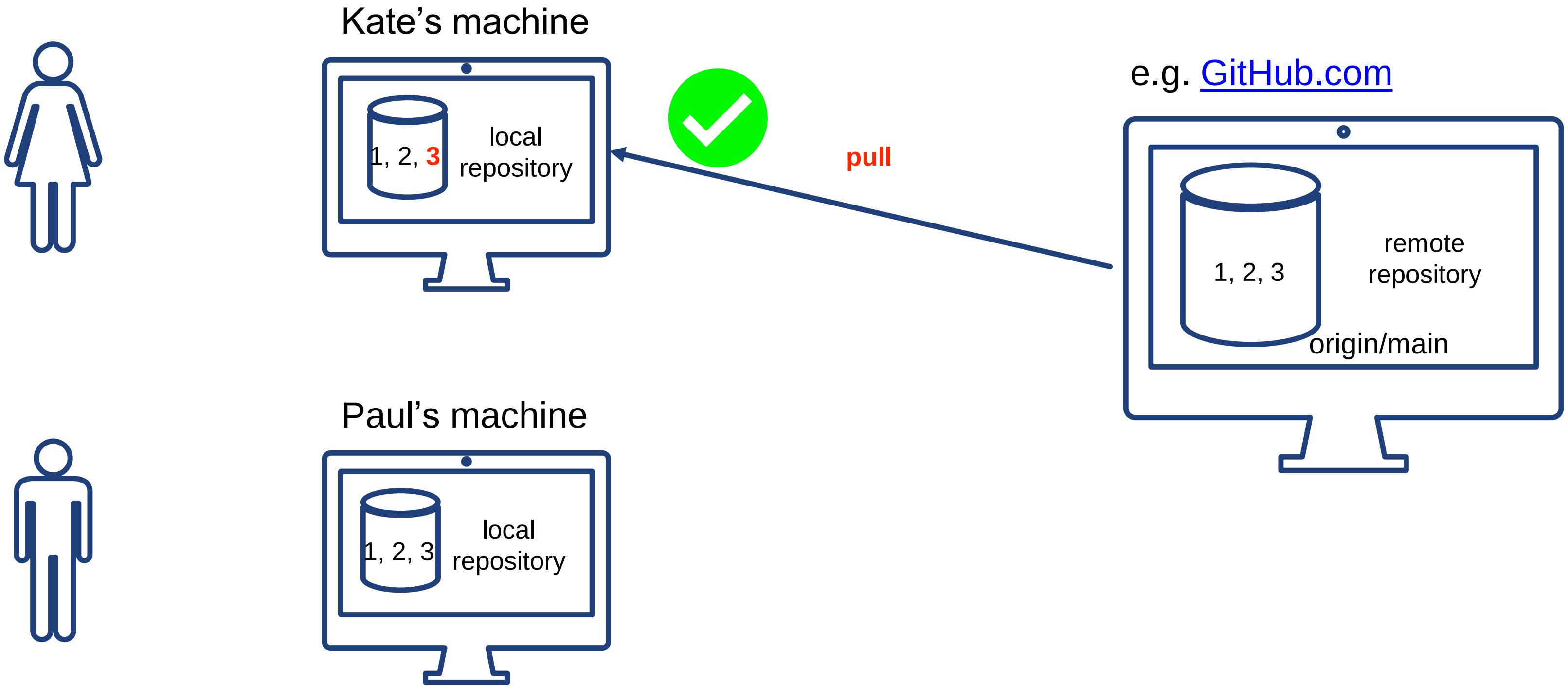
with REBASE

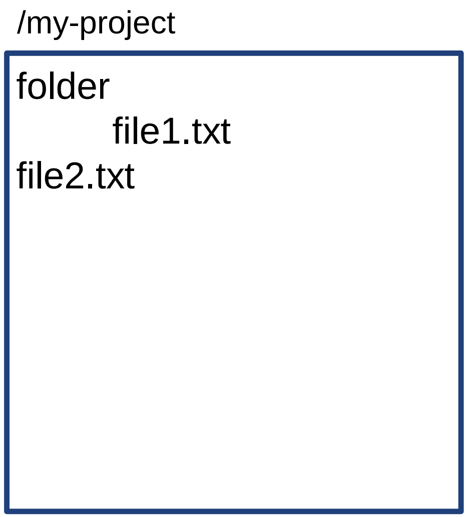
resolve possible conflicts

Collaboration



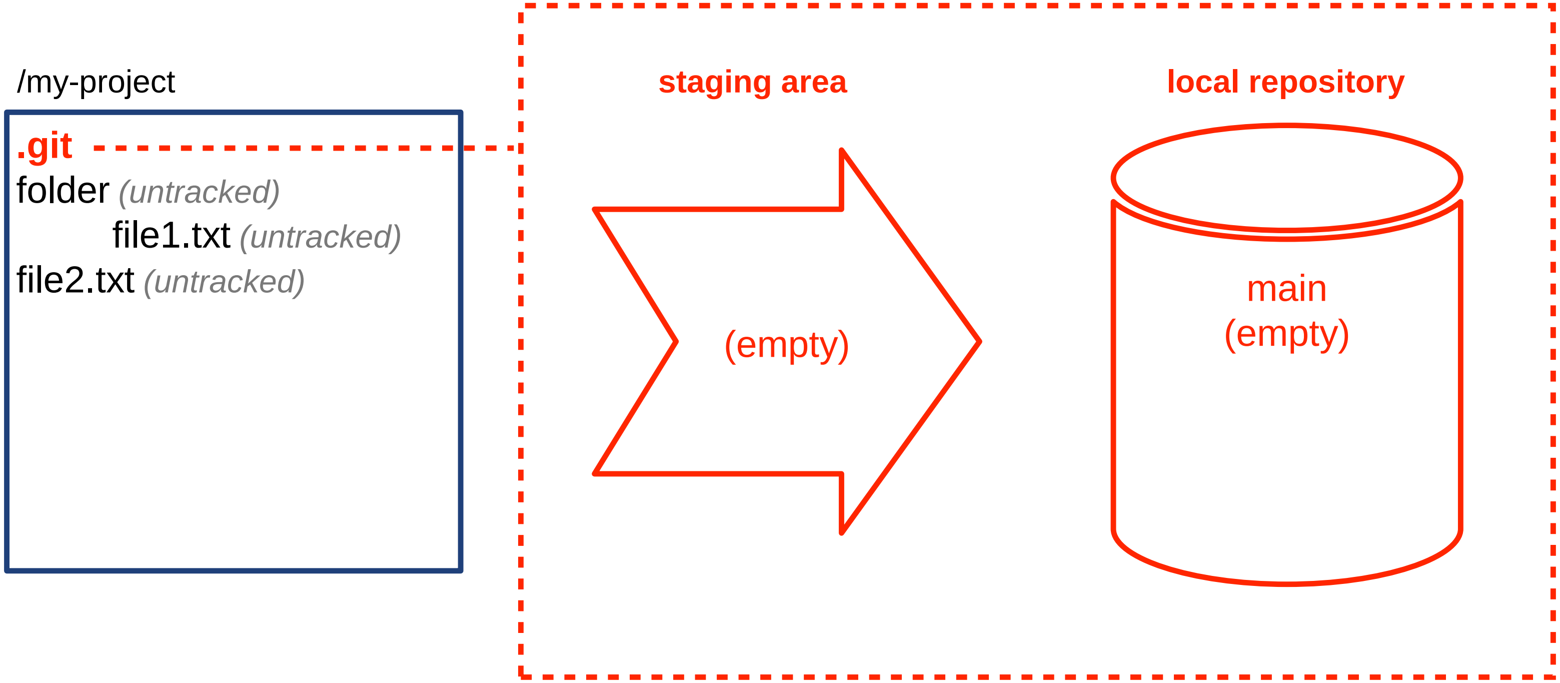
Collaboration



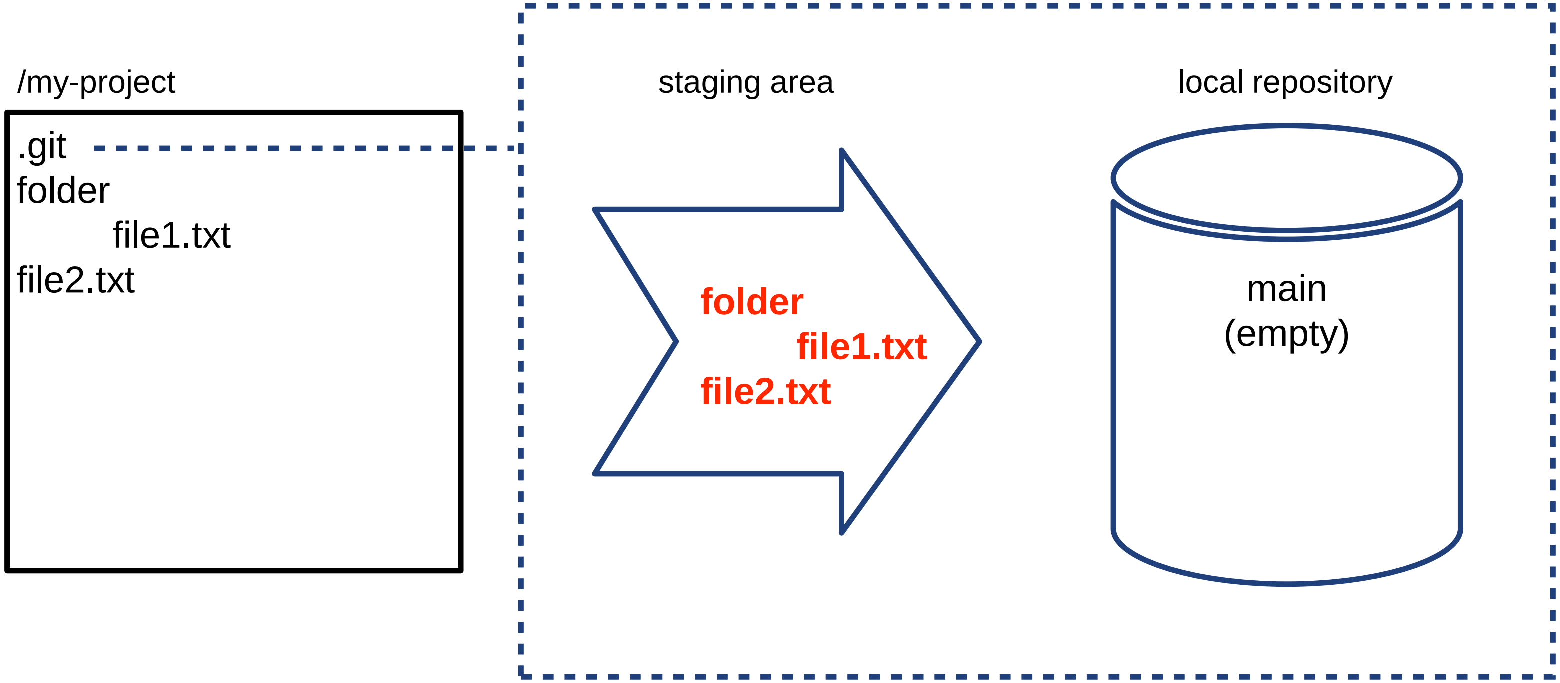


```
/my-project
folder
        file1.txt
file2.txt
```

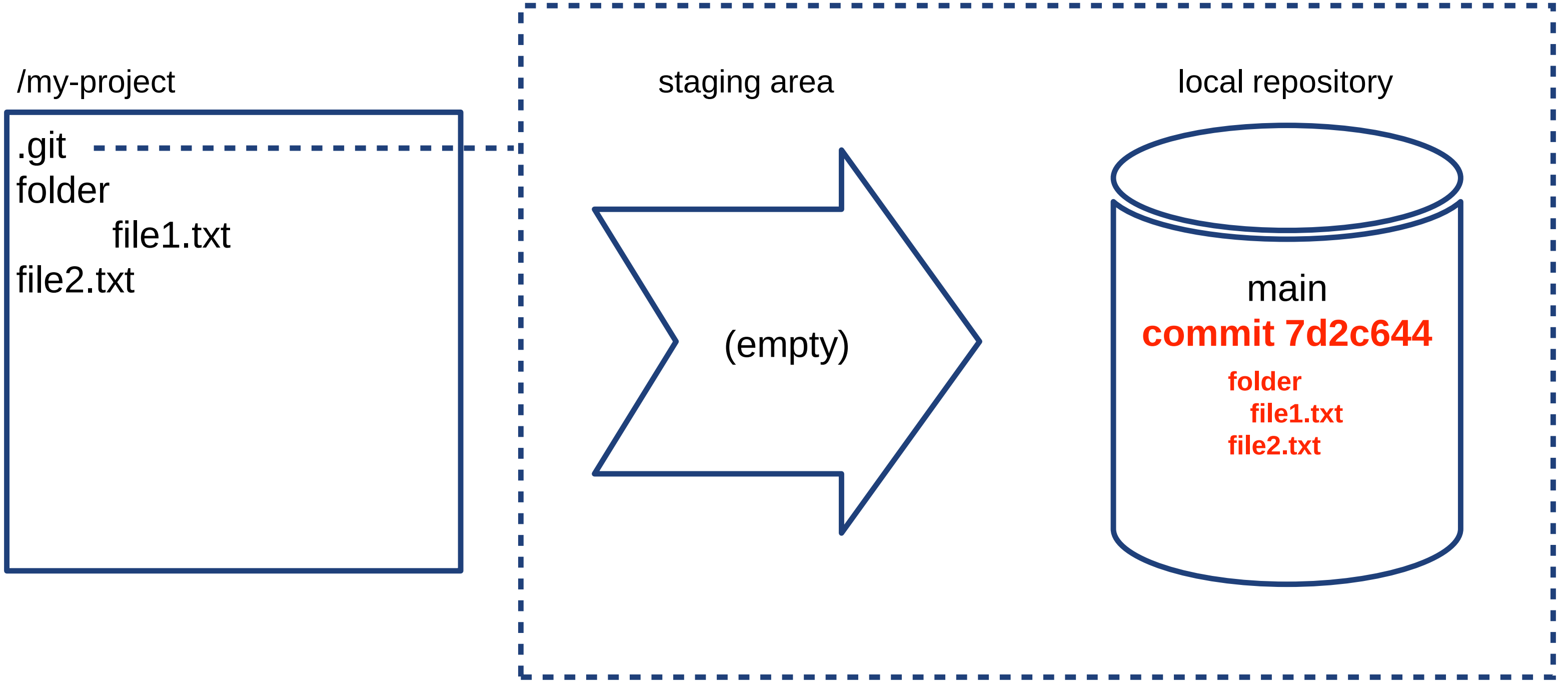
1. Initialize local Git repository: `git init`



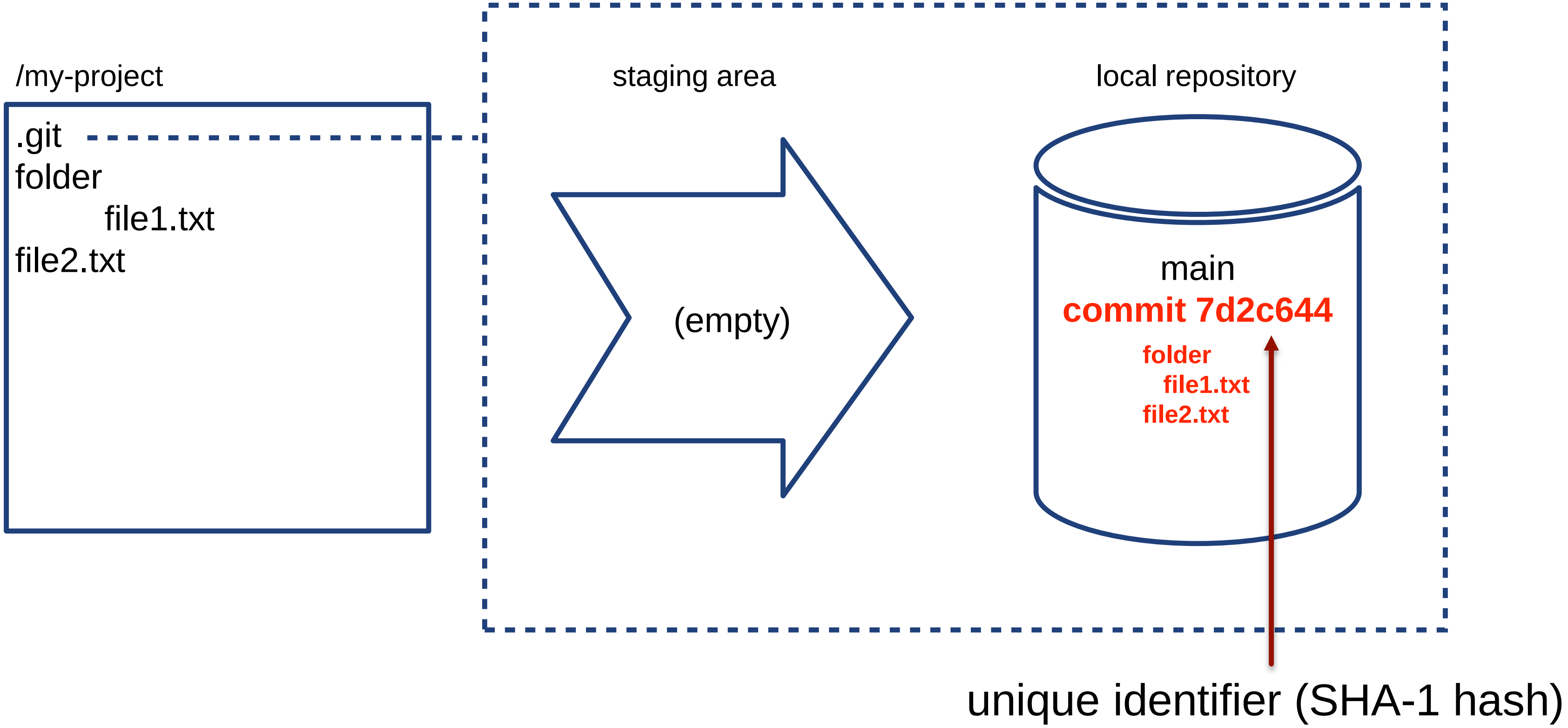
2. Tell Git to track *all* files: `git add .`



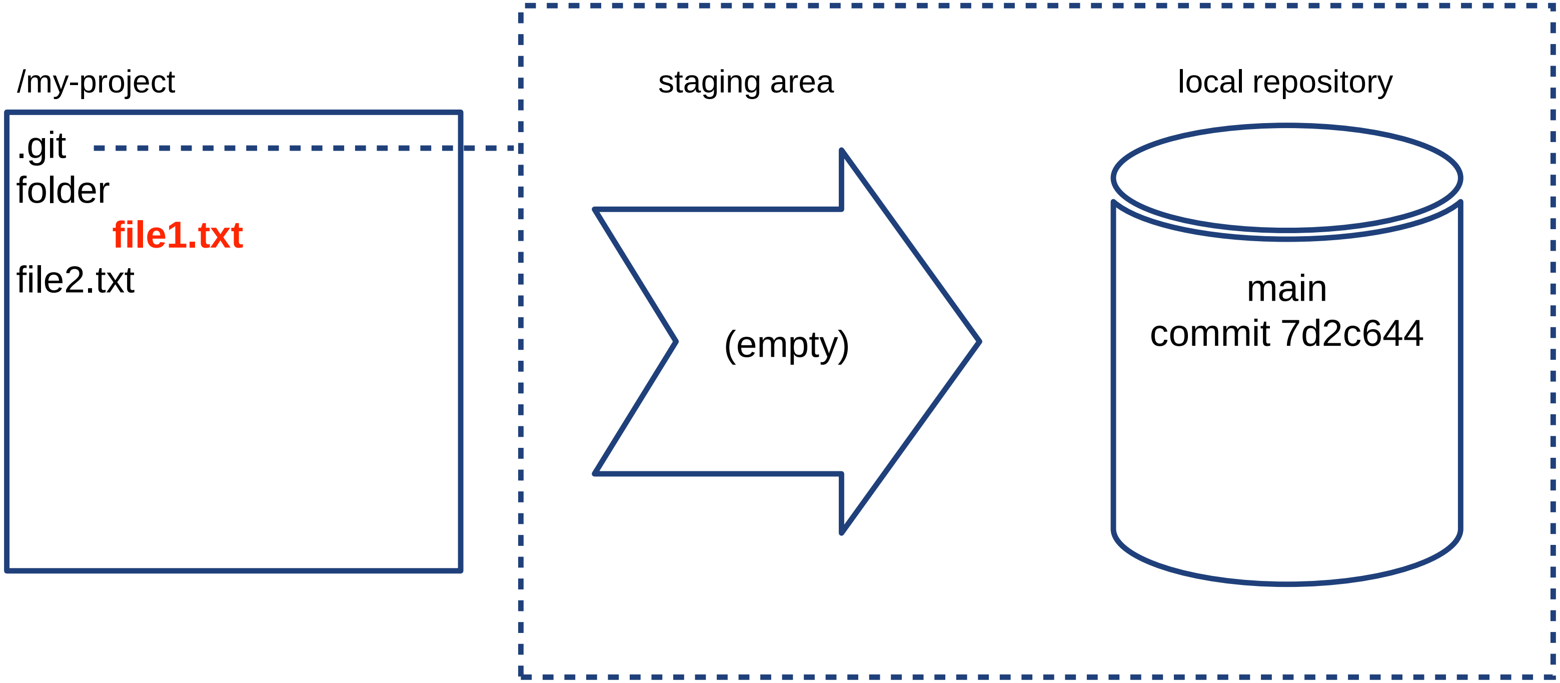
3. Commit *staged* files: `git commit -m "New!"`



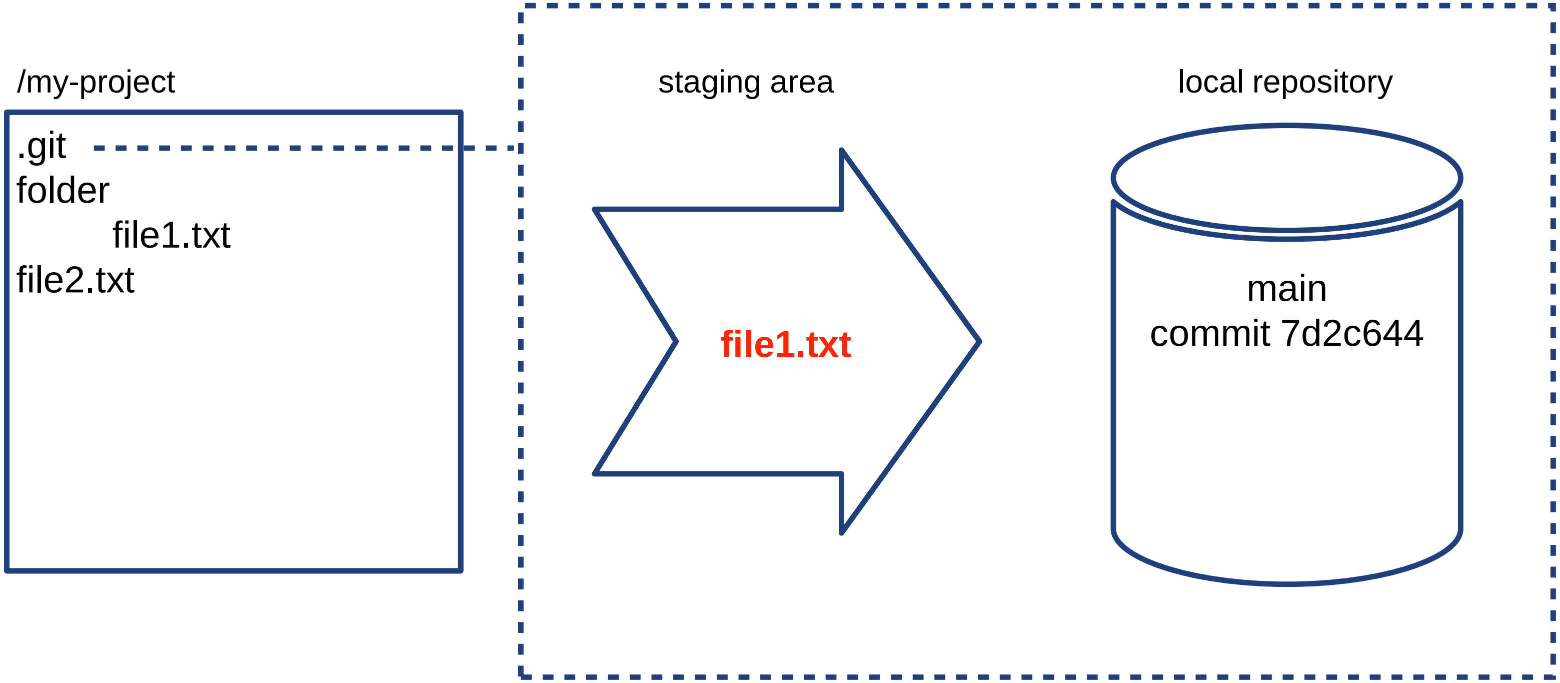
3. Commit *staged* files: `git commit -m "New!"`



4. Keep changing project files

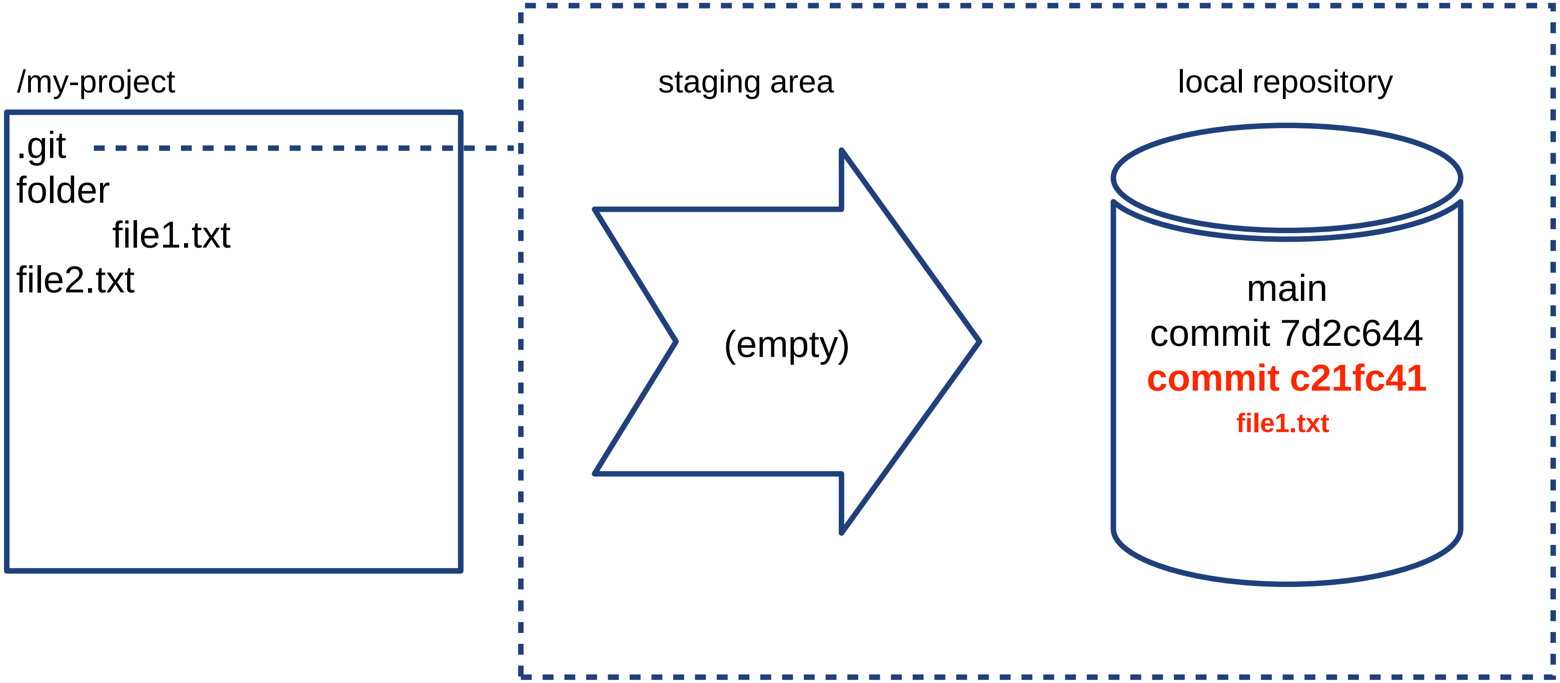


5. Stage changes of tracked files: `git add --update`

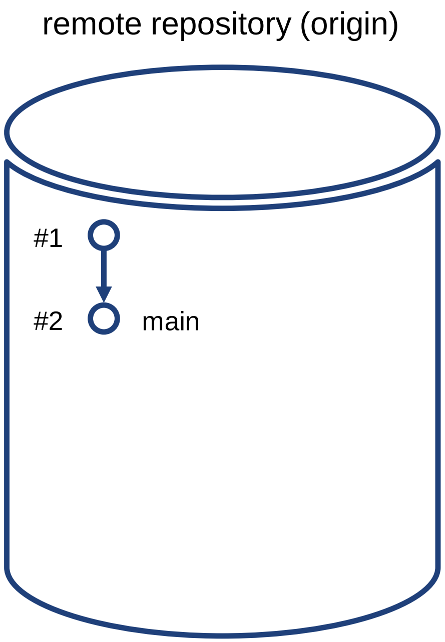


Or `git add –u` for short.

6. Commit staged files: `git commit -m "Newer!"`

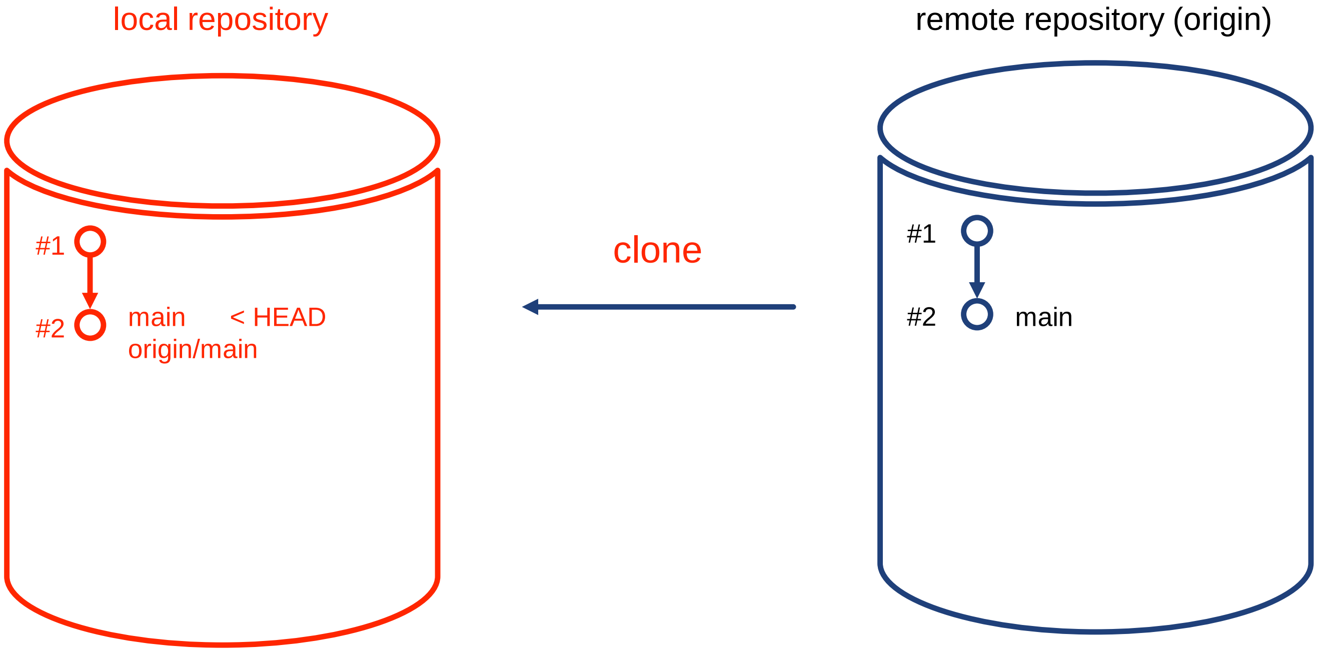


Working with remote Git repository



1. Clone remote repository:

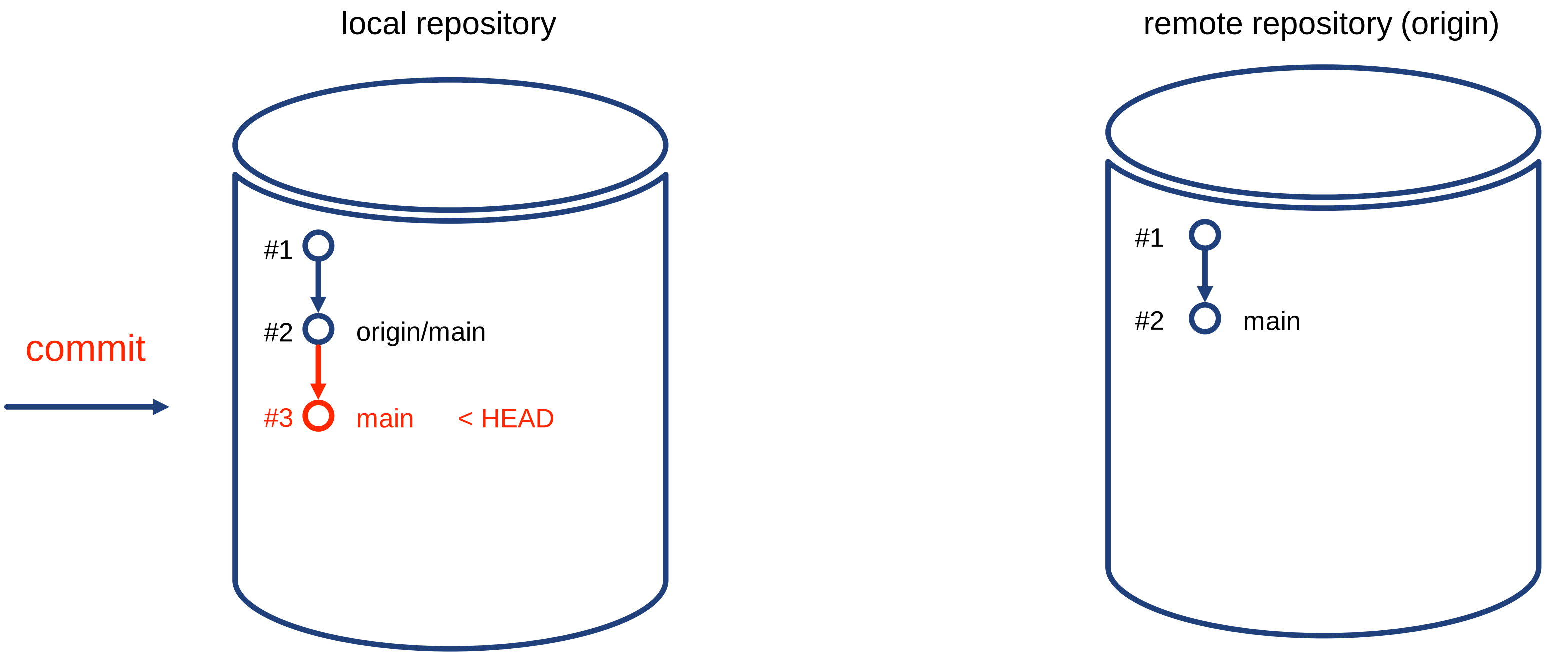
```
git clone git@example.com:group/myproject.git
```



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

Working with remote Git repository

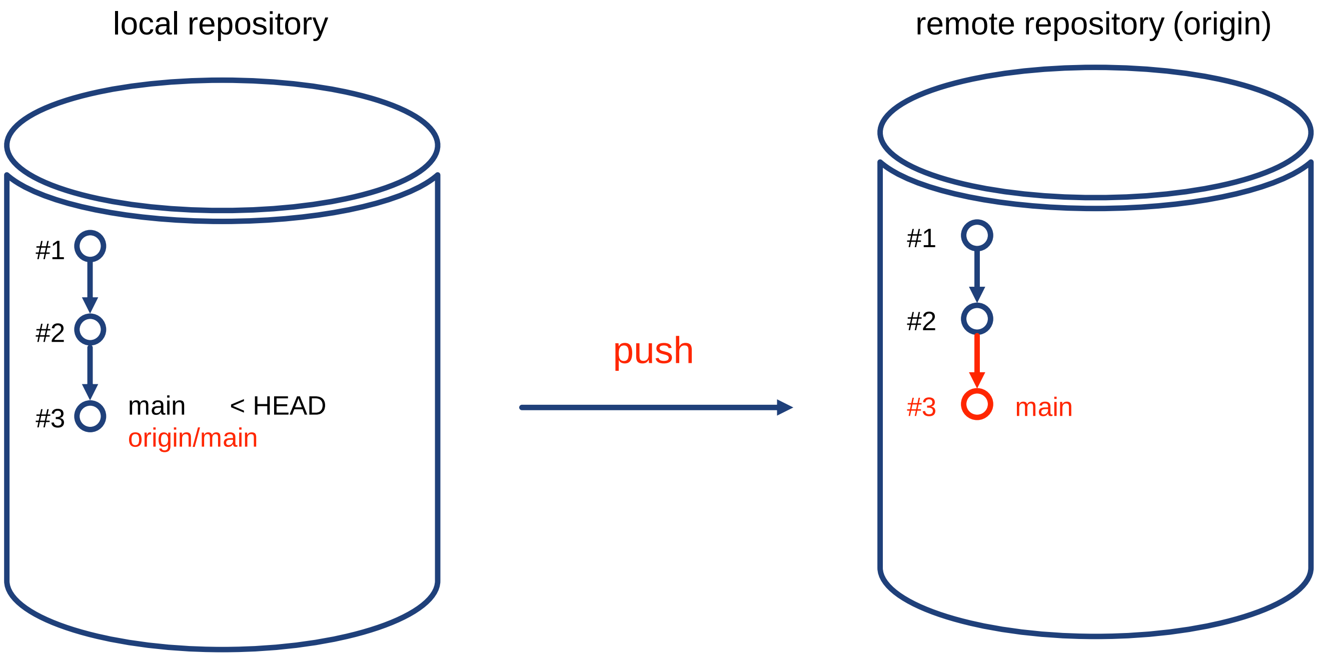
2. Commit changes locally: `git commit`



```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
```

Working with remote Git repository

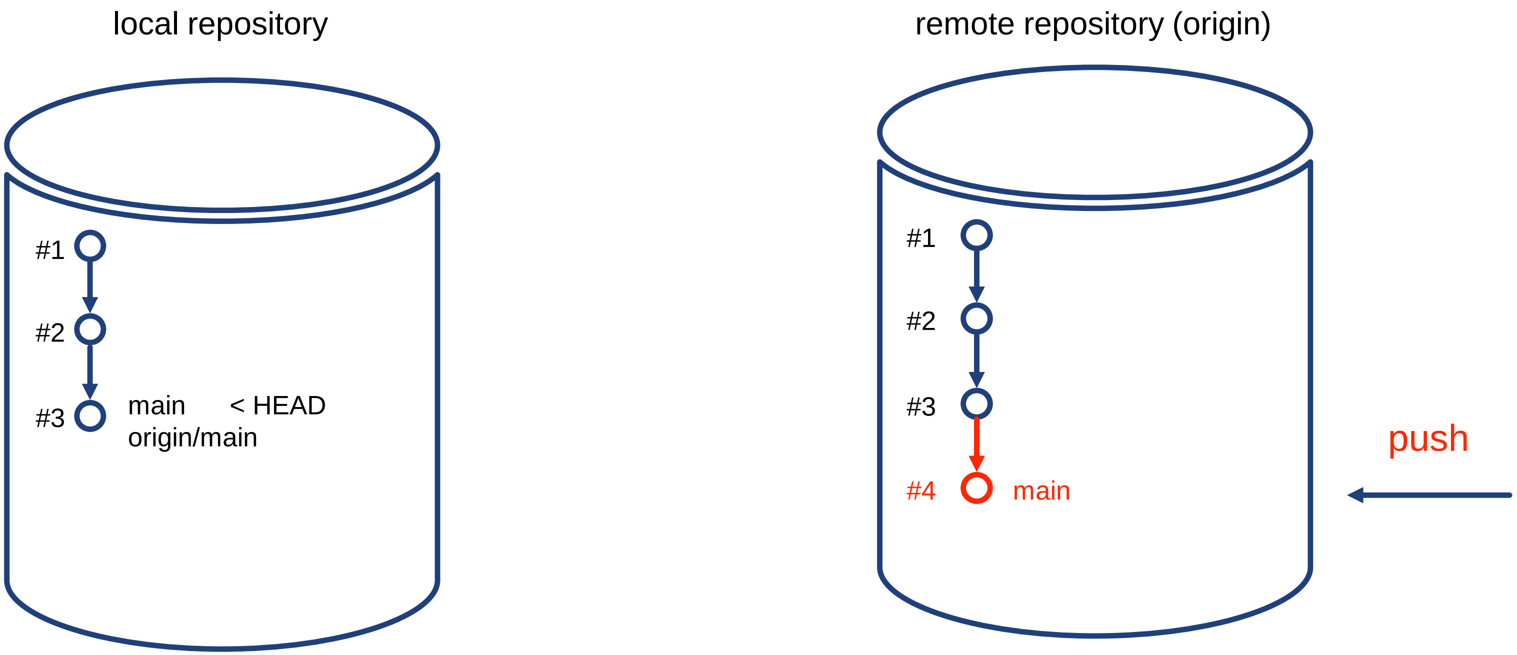
3. Push changes to remote repository: `git push`



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

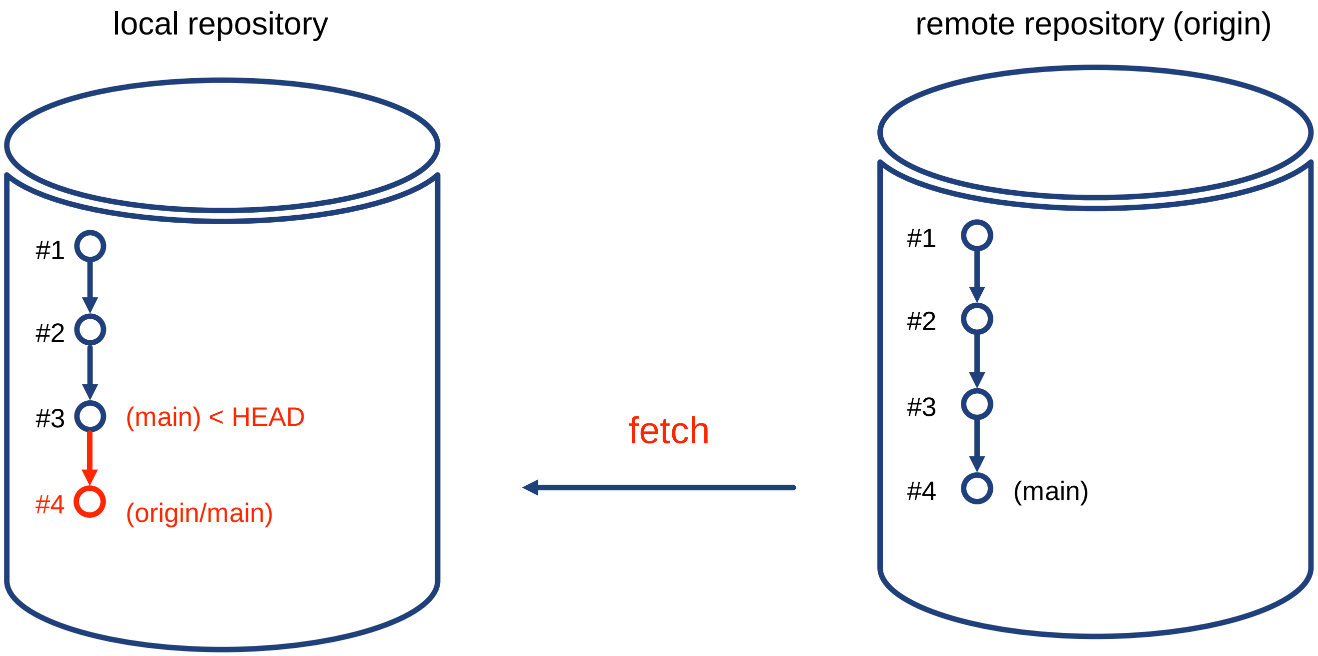
Working with remote Git repository

4. Someone pushes changes to the remote repository



Working with remote Git repository

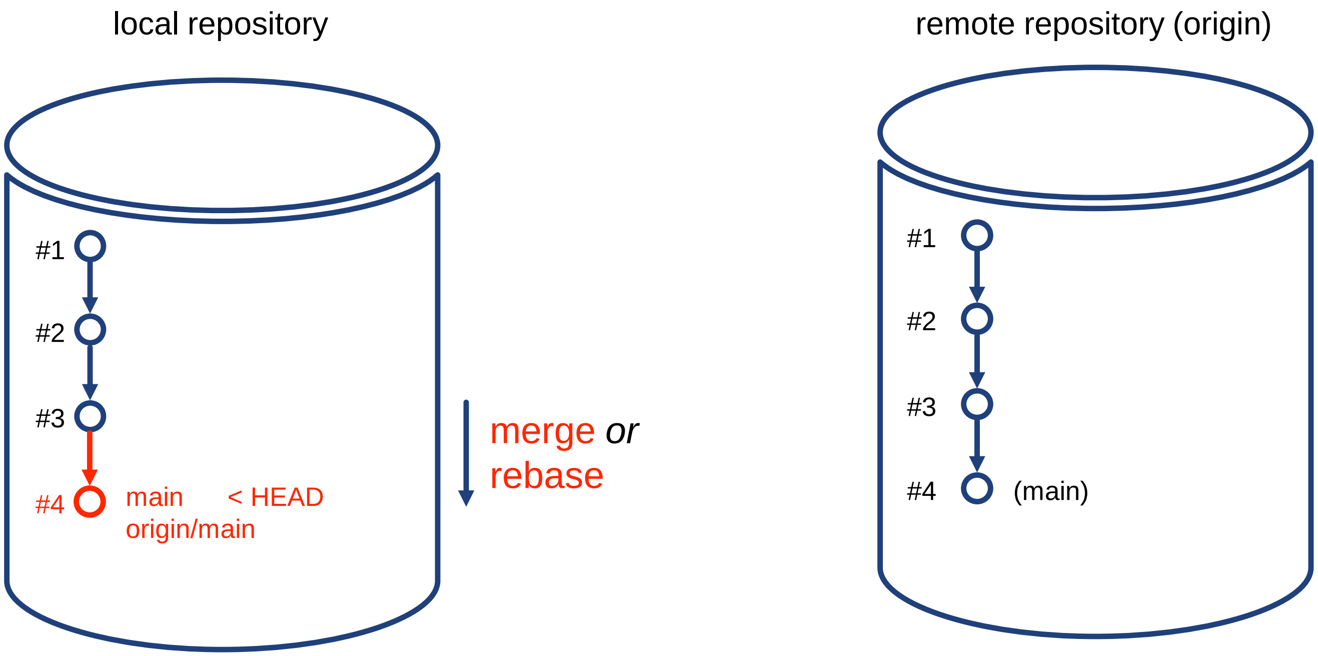
5. get latest from repo (without pull/merge): `git fetch`



```
$ git status
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
```

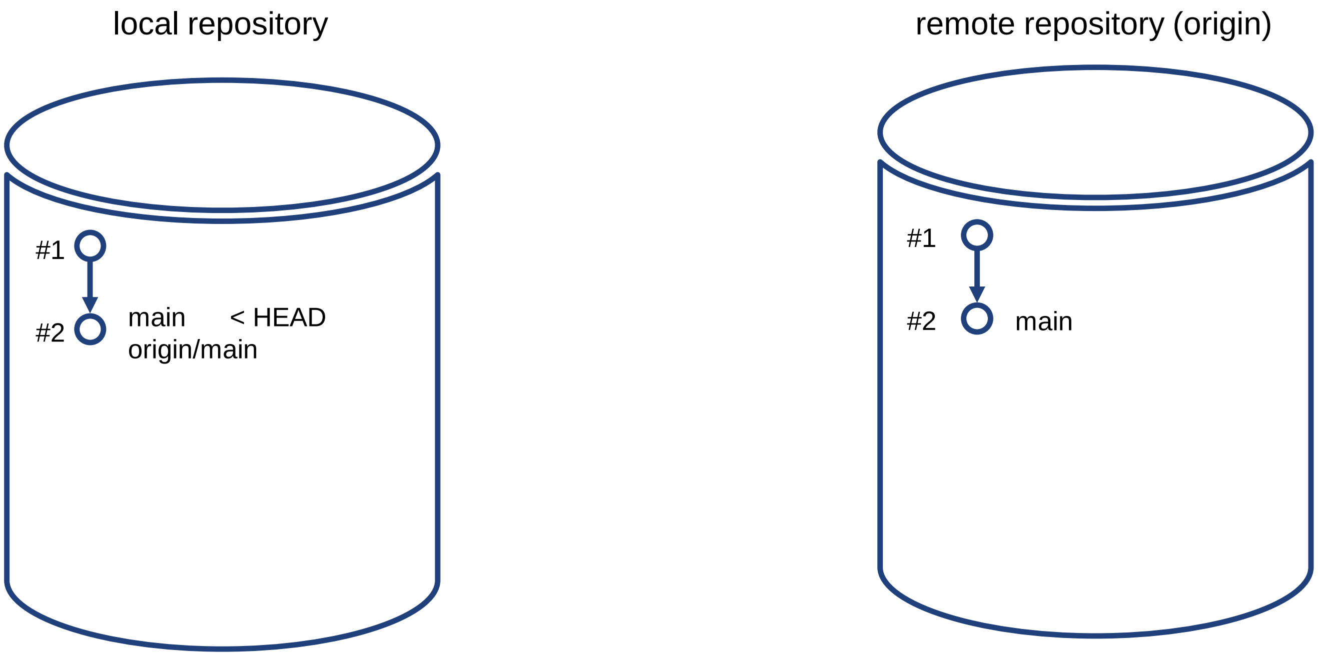
Working with remote Git repository

6. Merge latest changes: `git merge origin/main [--ff-only]`



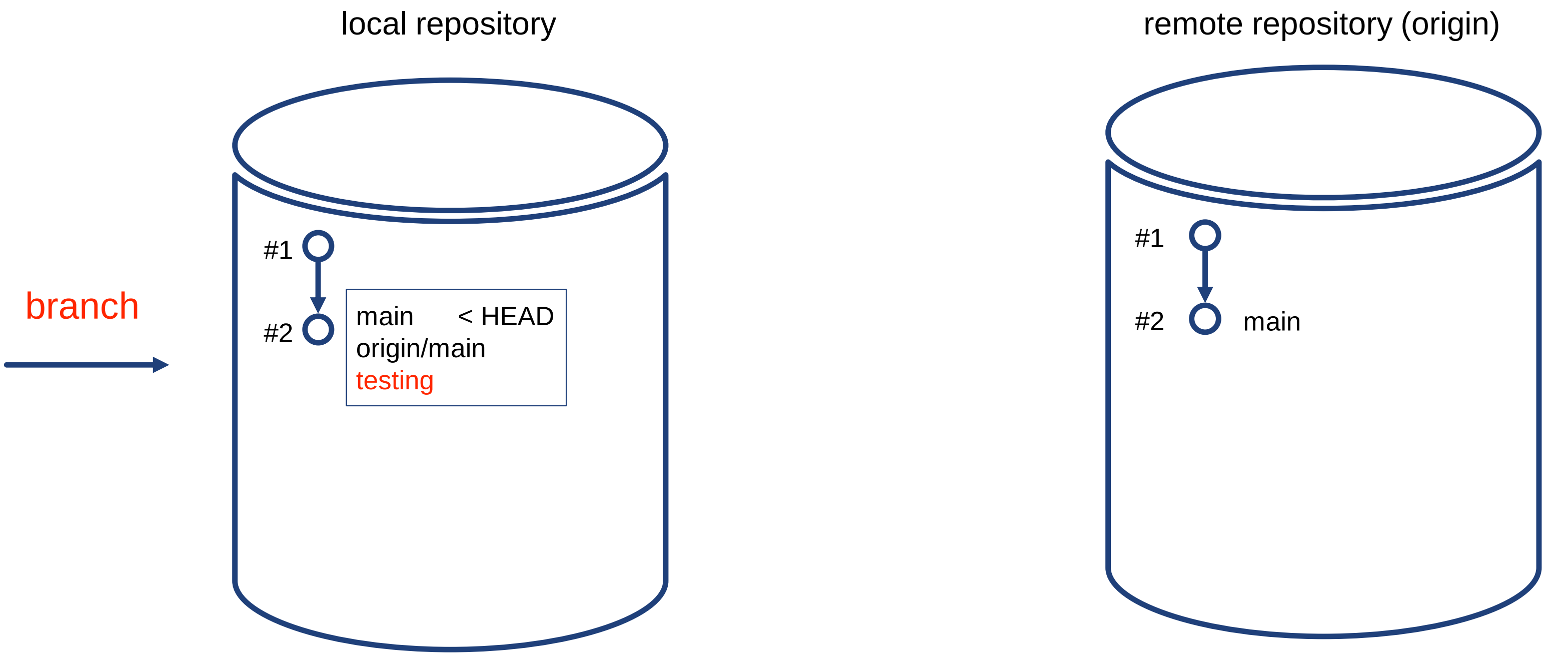
```
$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

Working with branches

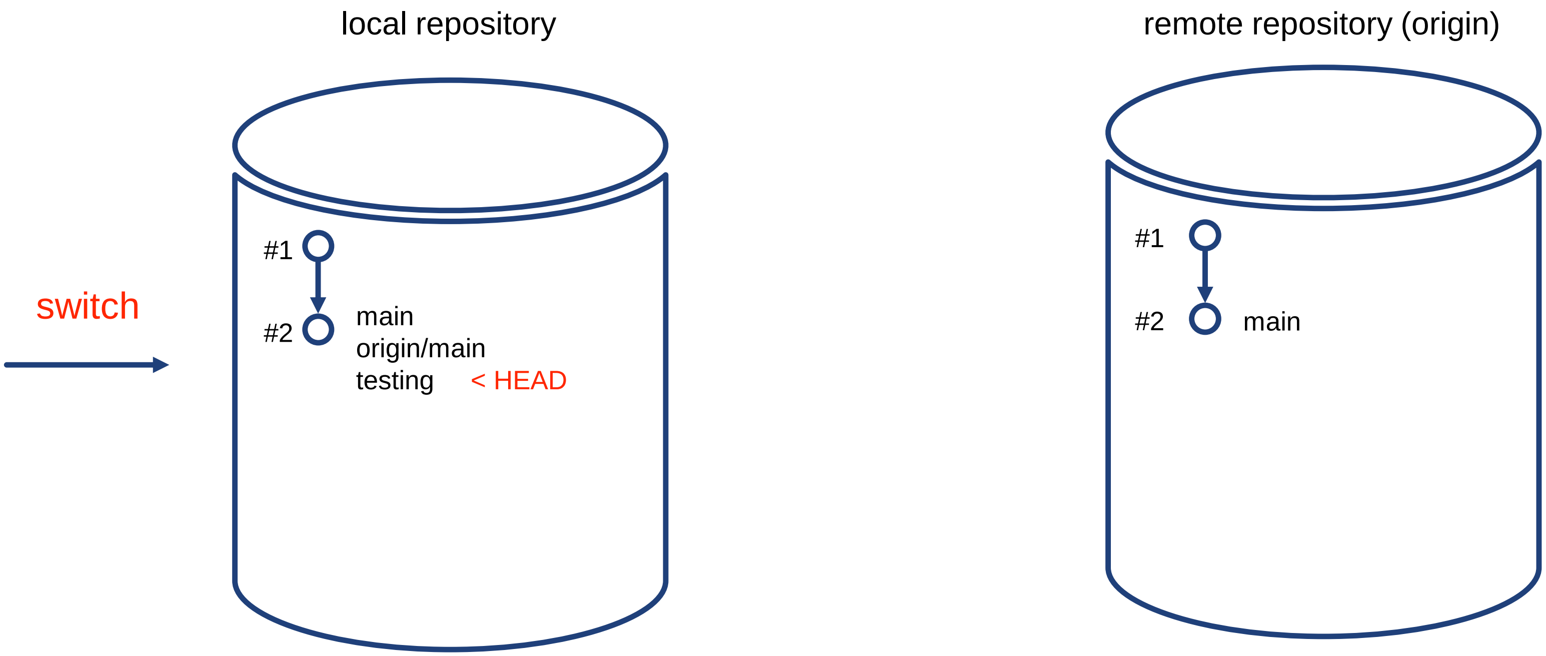


```
$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

1. Create a branch: `git branch testing`

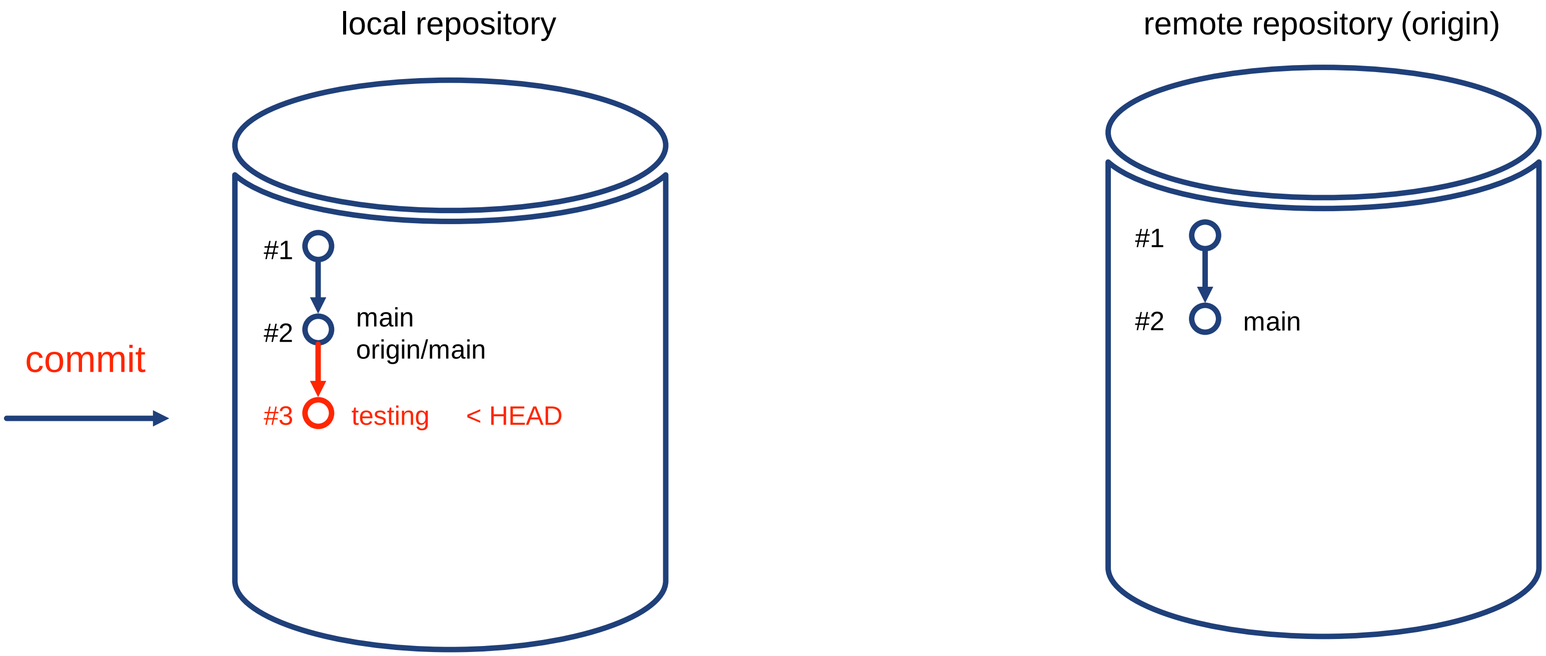


2. Switch to the branch: `git switch testing`



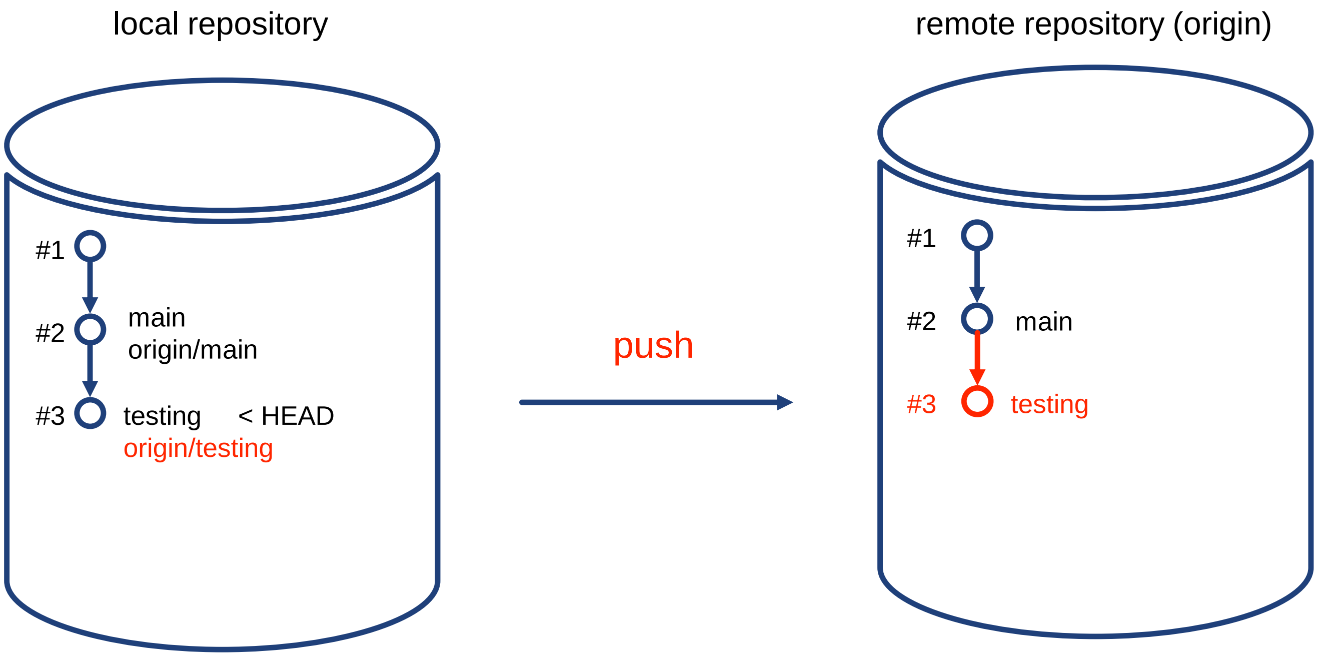
```
$ git status
On branch testing
```

3. Commit changes to the branch: `git commit`



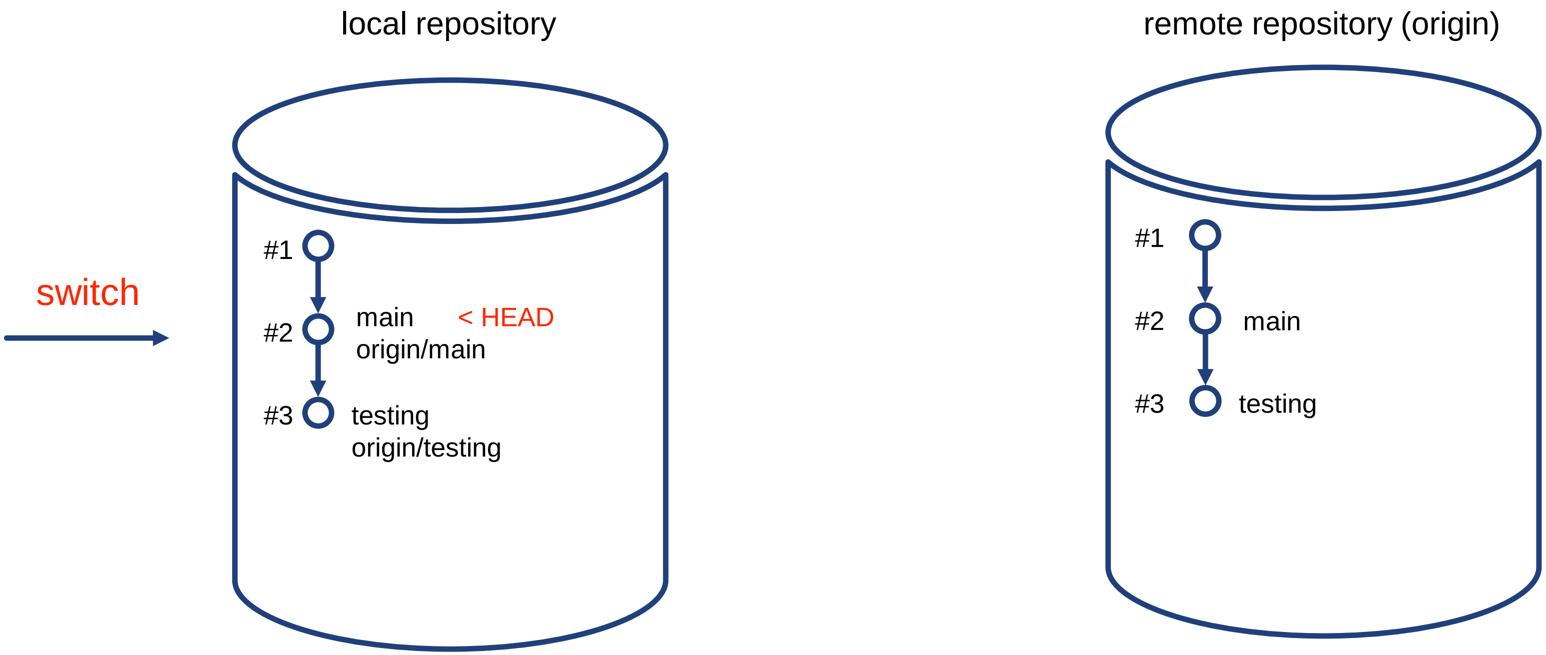
```
$ git status
On branch testing
```

4. Push to remote repository: `git push`



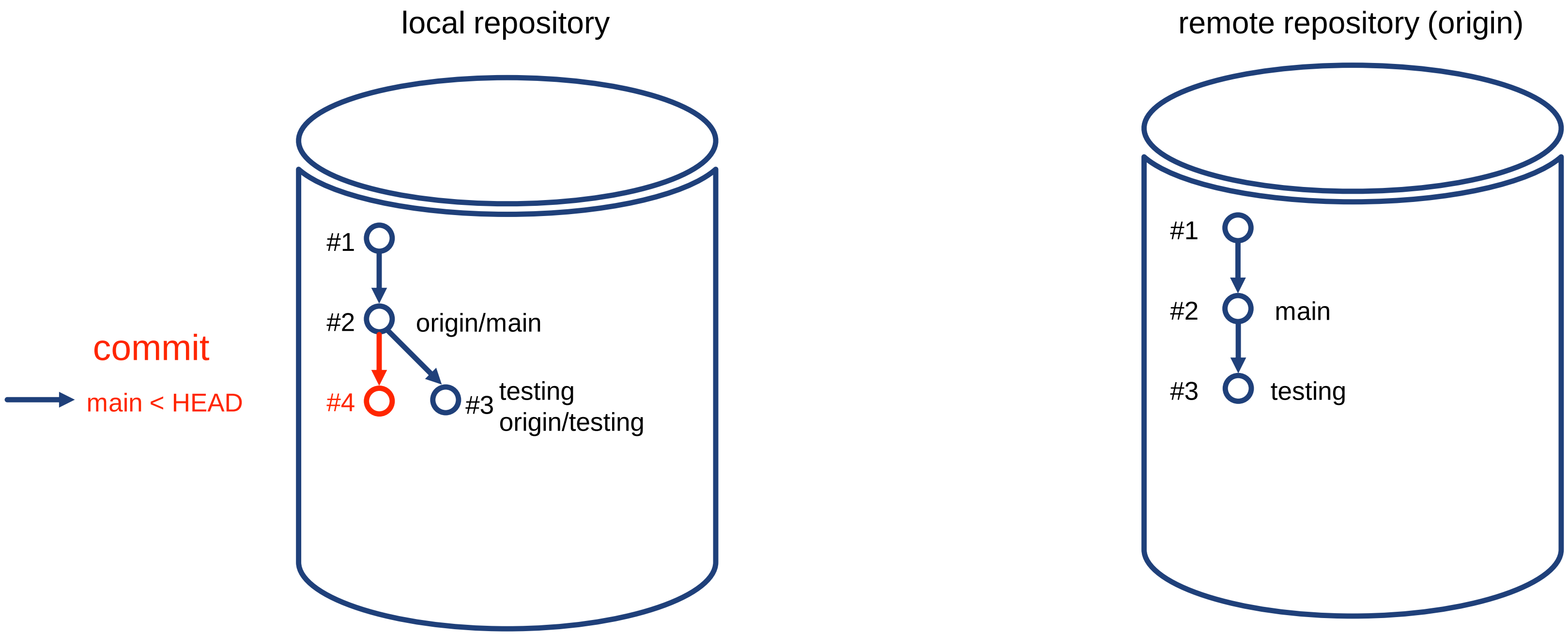
```
$ git status
On branch testing
Your branch is up to date with 'origin/testing'.
```

5. Switch back to main: `git switch main`



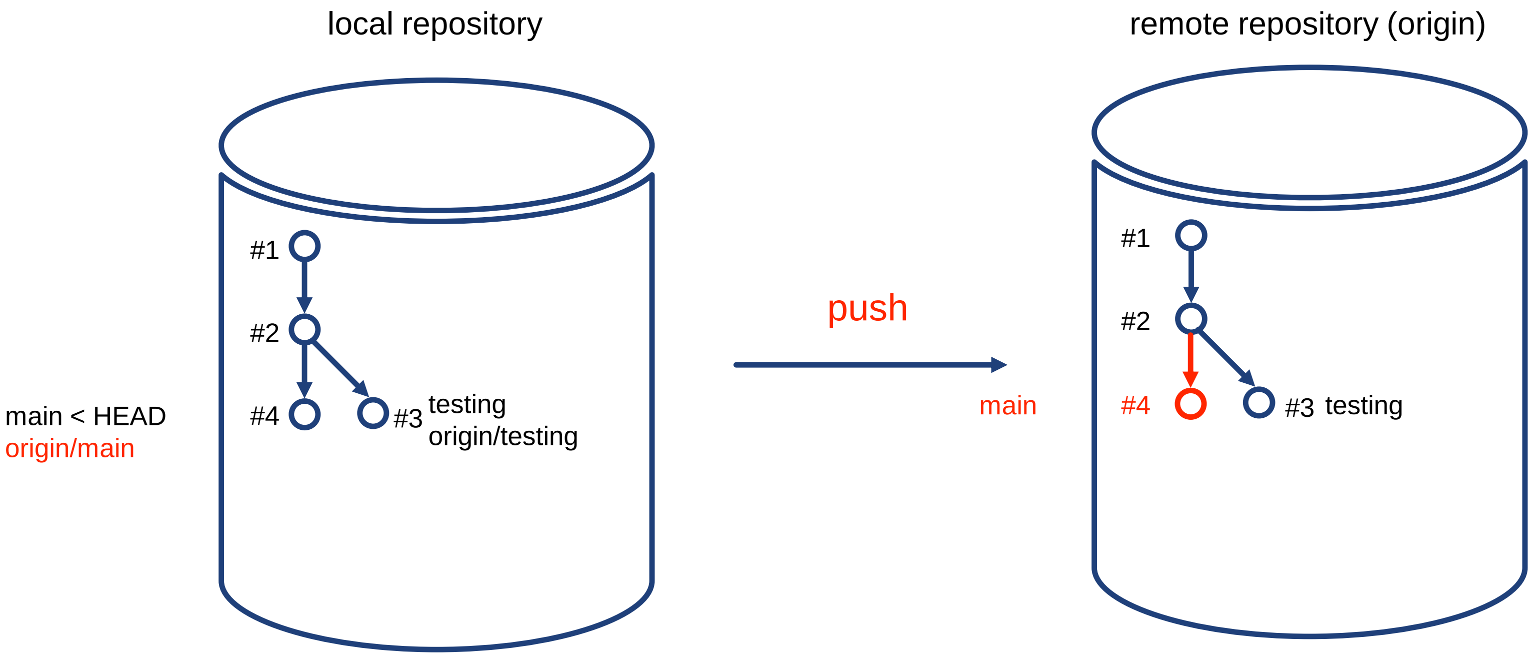
```
$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

6. Commit changes to main: `git commit`



```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
```

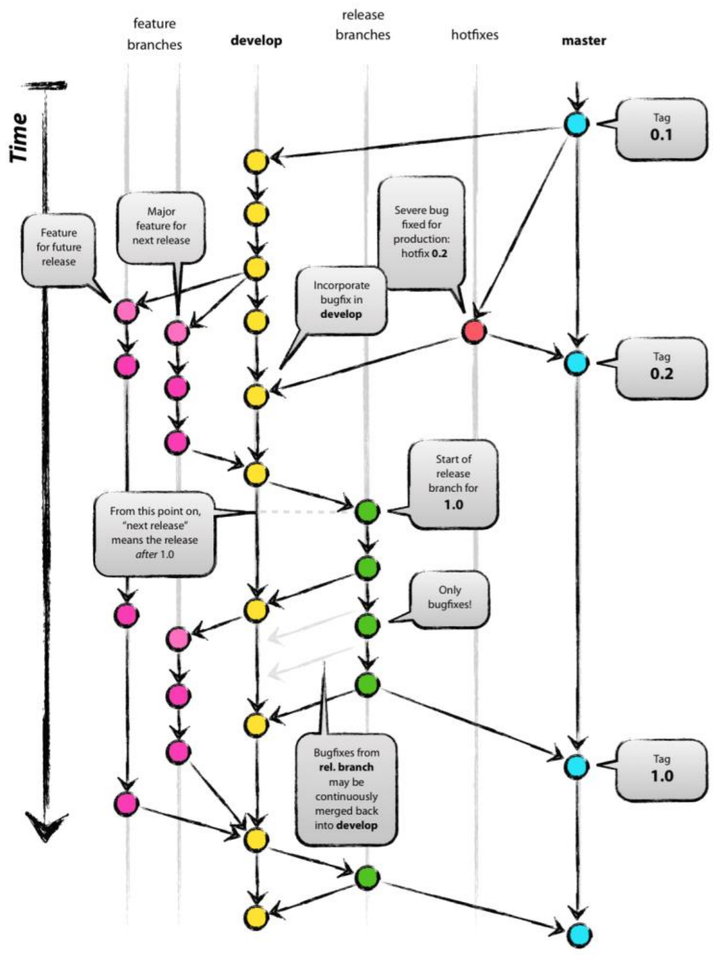
7. Push main to remote: `git push`



```
$ git status
On branch main
Your branch is up to date with 'origin/testing'.
```

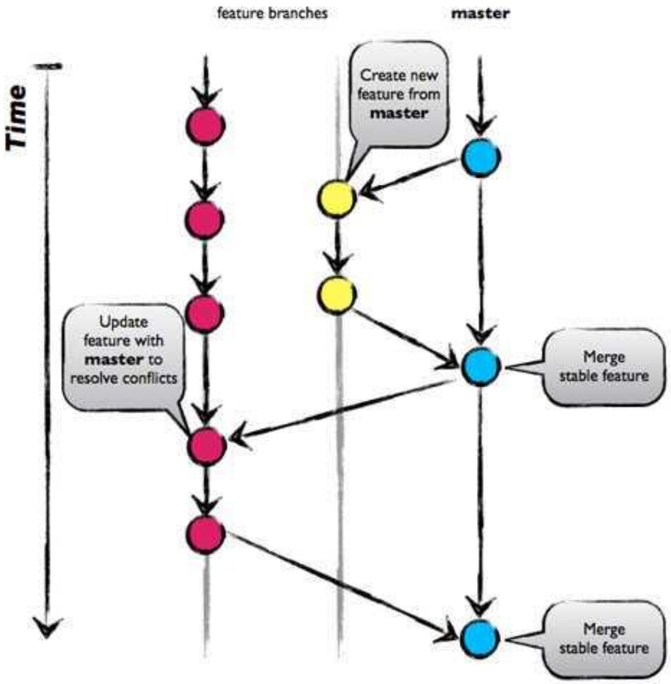
Branching strategies

As your project grows in complexity, you may need a proper branching strategy.

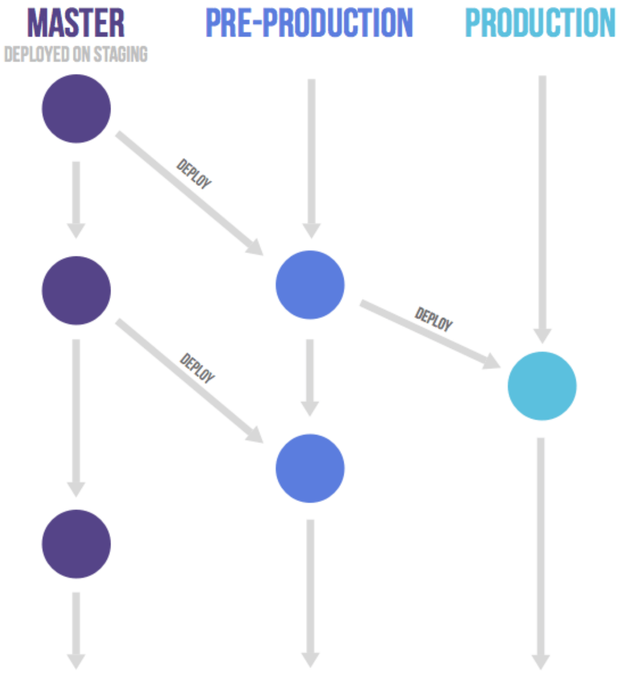

feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release *after* 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

GitFlow

Branching strategies


feature branches
master
Time
Create new feature from master
Update feature with master to resolve conflicts
Merge stable feature
Merge stable feature

But don’t make it more complicated than needed.



GitHub Flow

GitLab Flow

Why Git as VCS?

- Distributed
  - Local repository: speed + freedom
  - Remote repository: collaboration + backup
- Fast
  - Most operations are local.
  - Operations are lightweight, e.g. a branch is just a “moving label”.
- Industry standard

Best practices

- Atomic commits
  - Make scope-limited changes that are easy to grasp (and hence summarize).
  - Don’t “also” reformat/refactor code you need not change right now.
  - Keep the `diff`-s small. No distractions because of things like white-space.
- Document the change
  - Start commit message with one-line summary. (Easy with atomic commits.)
  - Elaborate the reasoning in a follow-up paragraph, if needed.
  - Think of your future self: In which commit did I make that change?
- Branching
  - Branch often, you can always merge (or rebase).
  - One branch for each issue/task. Break it down into commits as you see fit.
  - In the `main` branch, a (simple) “linear commit history” is often desirable.
- Keep out large files
  - Use extension like Git-LFS or Git-annex for large binary or data files.

Git remote servers with web UI

- Git web hosts:

ETH:	gitlab.ethz.ch
SWITCH:	gitlab.switch.ch
GitLab:	gitlab.com
GitHub:	github.com
BitBucket:	bitbucket.org

- Benefits:
  - code browser
  - merge/pull requests
  - issue tracker
  - documentation hosting (wiki, web site)
  - continuous integration, continuous deployment (CI/CD)

Git hands-on session

Let’s dig in!

https://siscourses.ethz.ch/git-handson/hands_on_git.html