Introduction to Git

Database Systems
DataLab, CS, NTHU
Spring, 2020

Outline

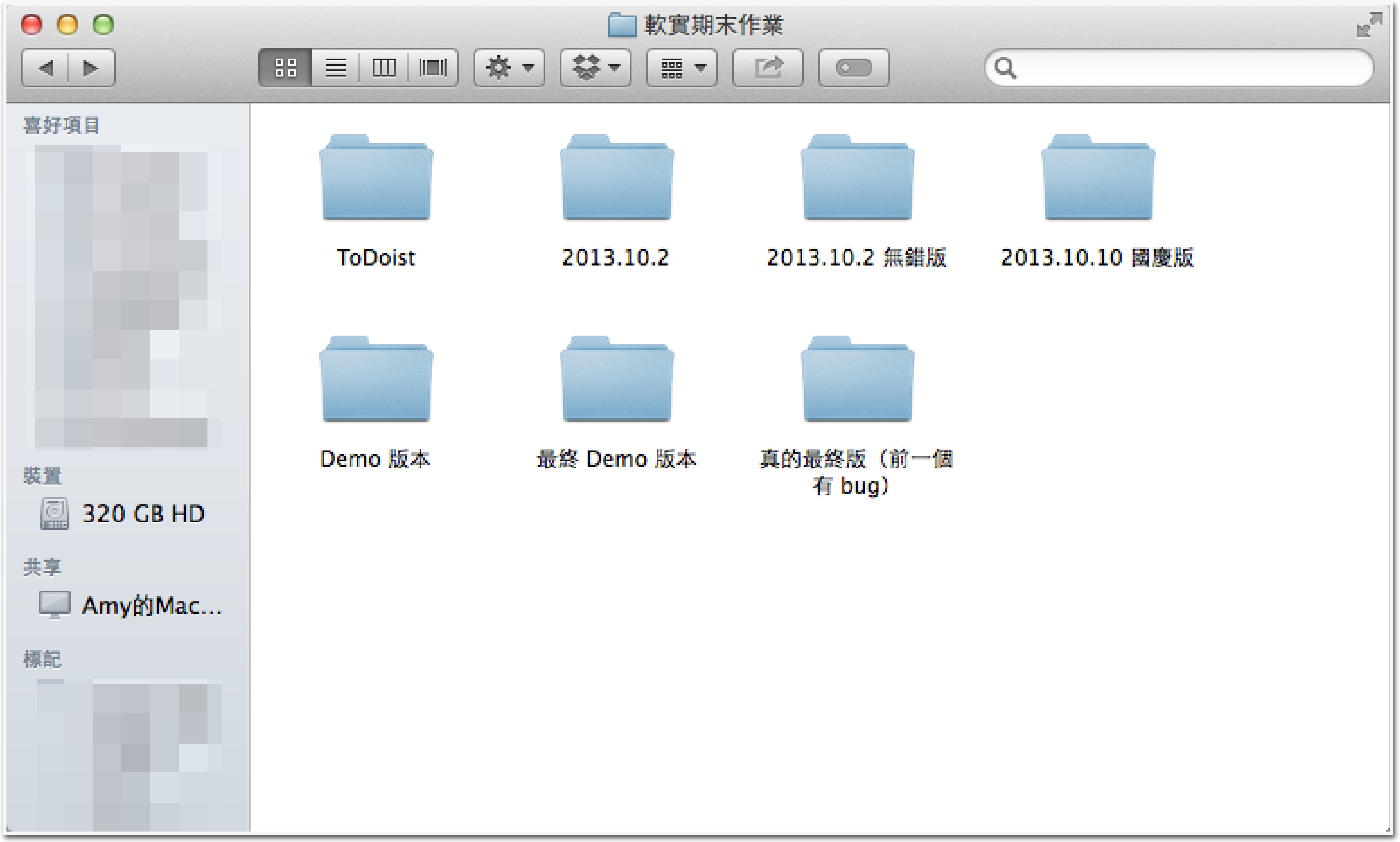
- Version control system
- Git basics
- Git branch
- Remote repository

Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Why Version Control ?

Students’ VCS

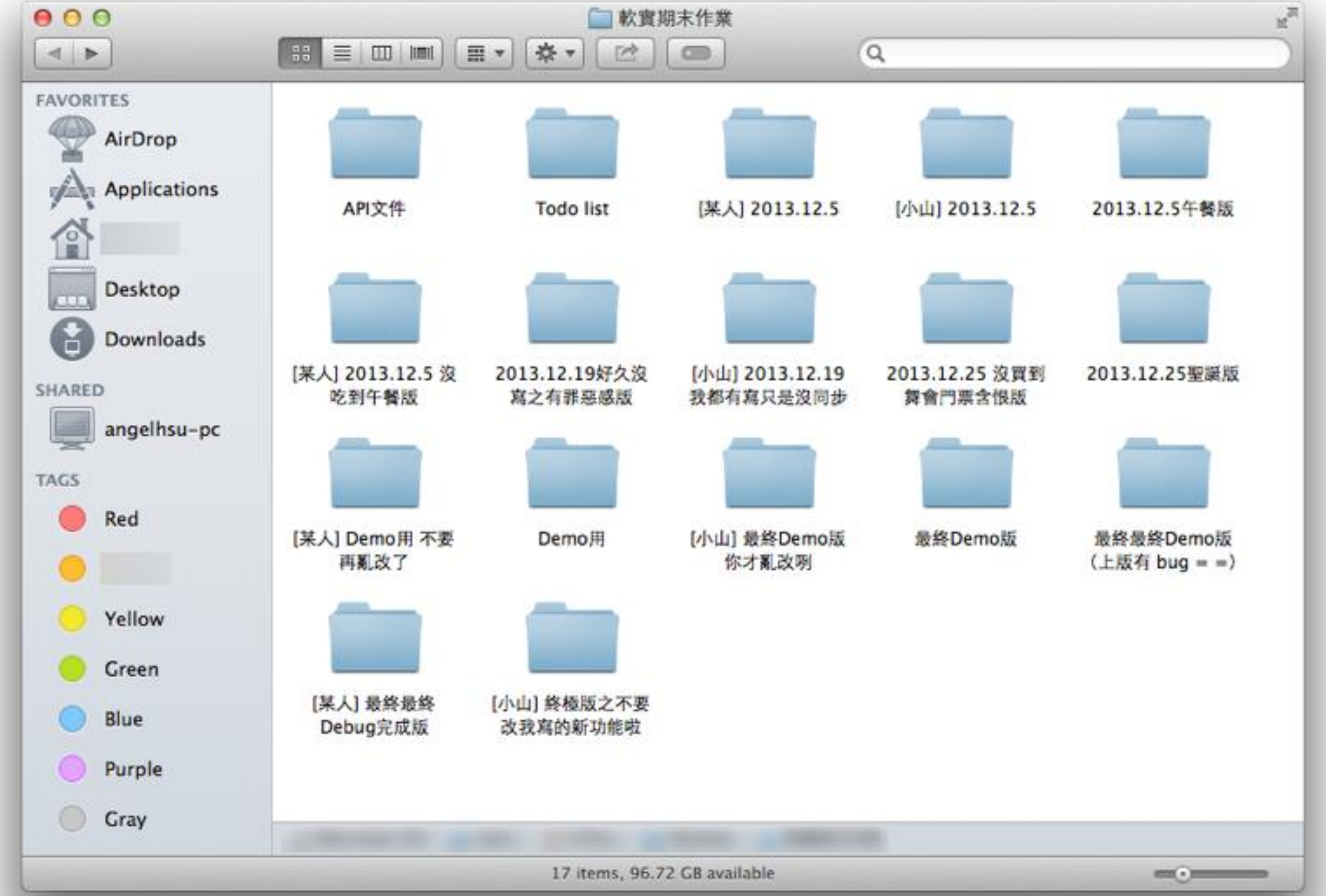


How to work with others?

?

?

Dropbox VCS in Reality



軟實期末作業
FAVORITES
AirDrop
Applications
Desktop
Downloads
SHARED
angelhsu-pc
TAGS
Red
Yellow
Green
Blue
Purple
Gray
API文件
Todo list
[某人] 2013.12.5
[小山] 2013.12.5
2013.12.5午餐版
[某人] 2013.12.5 沒吃到午餐版
2013.12.19好久沒寫之有罪惡感版
[小山] 2013.12.19 我都有寫只是沒同步
2013.12.25 沒買到舞會門票含恨版
2013.12.25聖誕版
[某人] Demo用 不要再亂改了
Demo用
[小山] 最終Demo版 你才亂改咧
最終Demo版
最終最終Demo版（上版有 bug = =）
[某人] 最終最終 Debug完成版
[小山] 終極版之不要改我寫的新功能啦
17 items, 96.72 GB available

Version Control System

- Store the projects, keep your revision history
- Synchronization between modifications made by different developers

https://memecrunch.com/meme/8QNIV/now-git

Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Git

- Git is a version control system which is
    - Fast
    - Easy to use
    - Distributed
    - Able to handle large project
      (ex. Linux Kernel 27.8 million lines)
- A git repository is a mini database that tracks your files

Installation

- Please check this link
  - http://git-scm.com/book/en/Getting-Started-Installing-Git

Configuration

- Modify ~/.gitconfig
- Or, type in following commands

```
git config --global user.name  “your name”
git config --global user.email  “your@email.com”
```

For more information, please refer this link

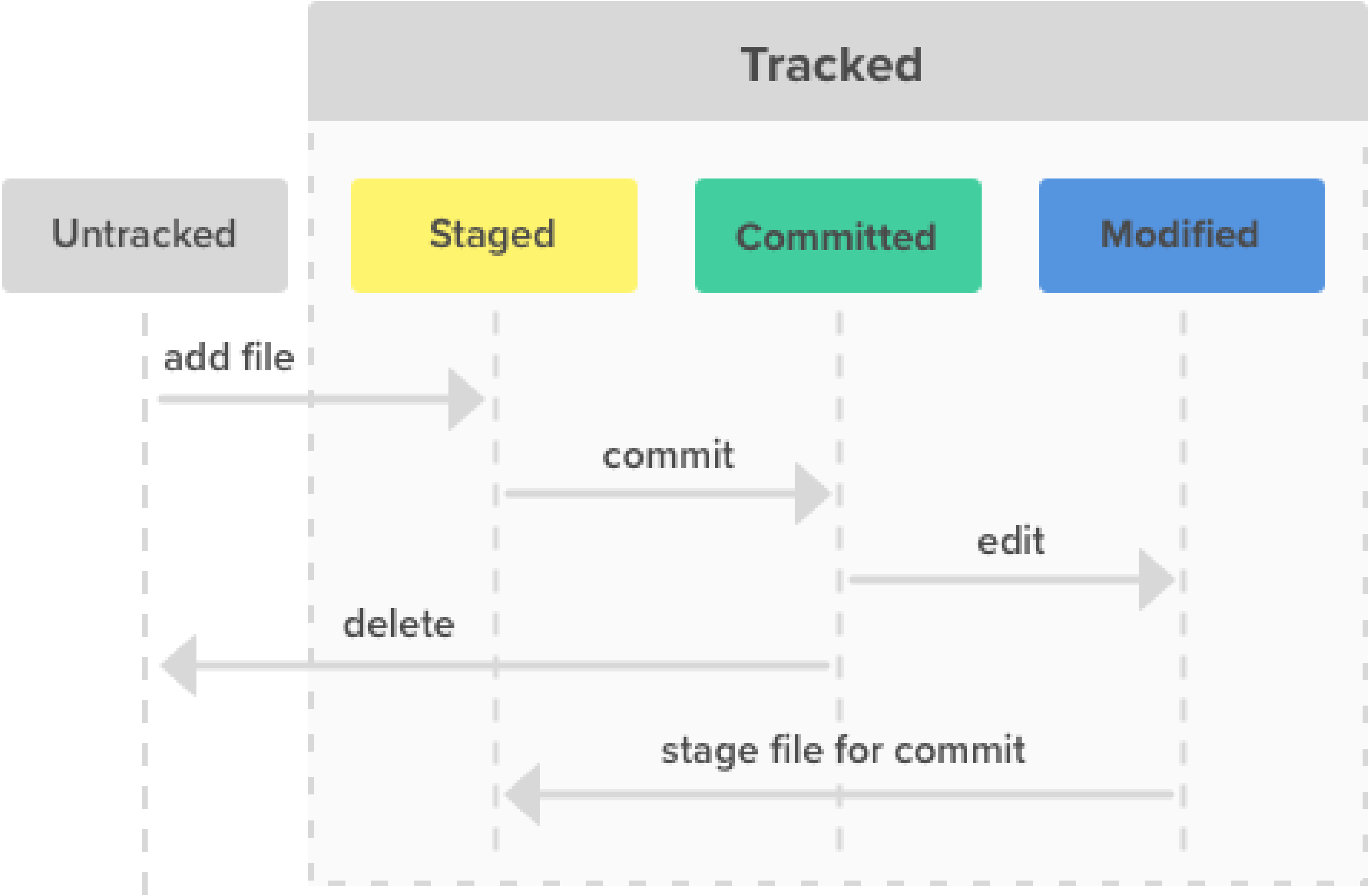
Creating a new Repository

- Two ways to create a repository
  - Initializing a Repository in an Existing Directory

    ```
    git init
    ```

  - Cloning an Existing Repository
    - We will talk about it later
- The repository information will be stored in the .git directory

Committing A Version



Committing A Version

- Staging (adding) a file

  ```
  git add [file name]
  ```

- Staging all files in the current directory

  ```
  git add -A
  ```

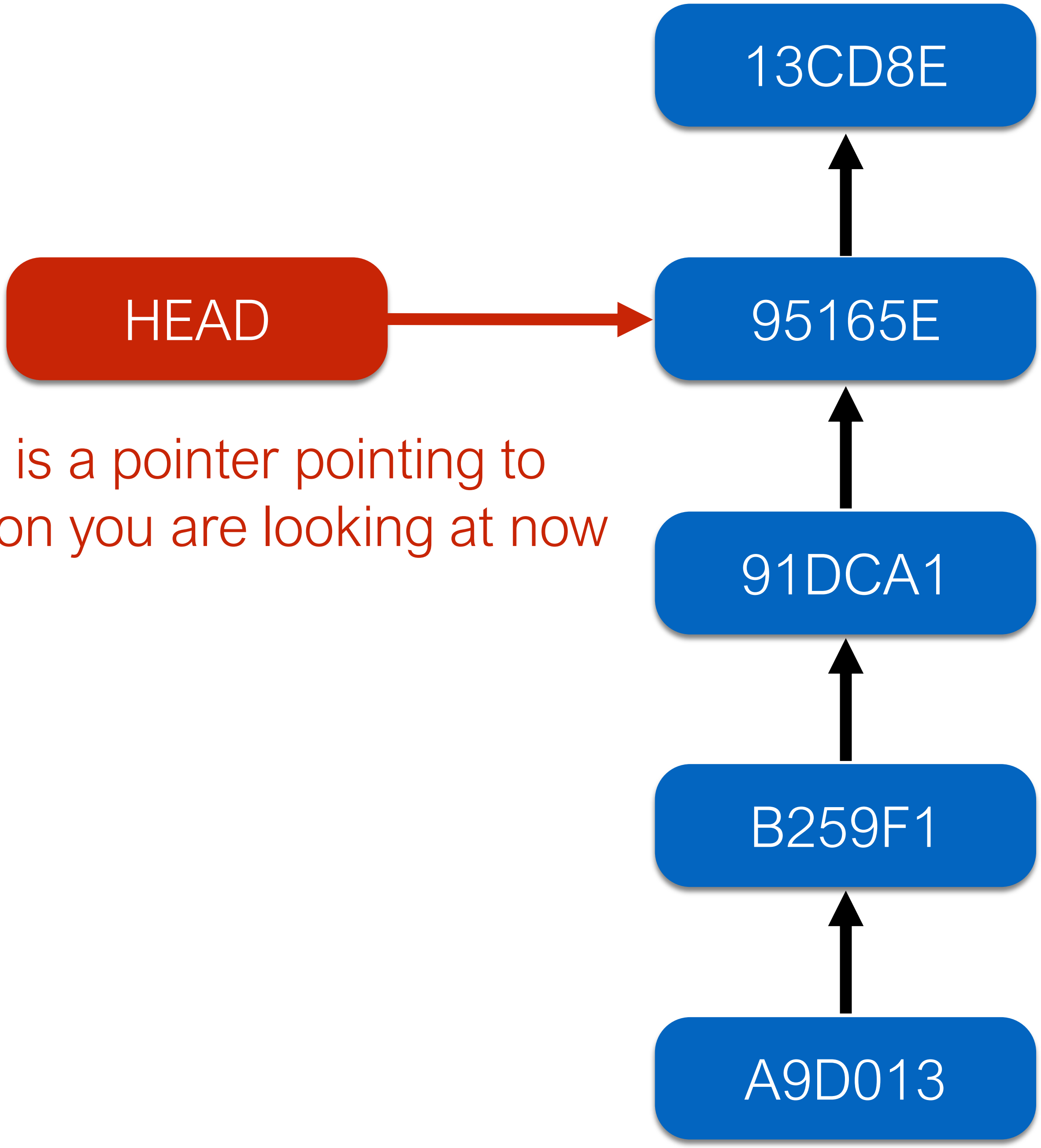
- Committing

  ```
  git commit -m “[message]”
  ```

Status

- Checking the current status and the current branch
  `git status`

A History Tree



13CD8E

HEAD

95165E

91DCA1

B259F1

A9D013

HEAD is a pointer pointing to the version you are looking at now

The versions you committed form a history tree

Logs

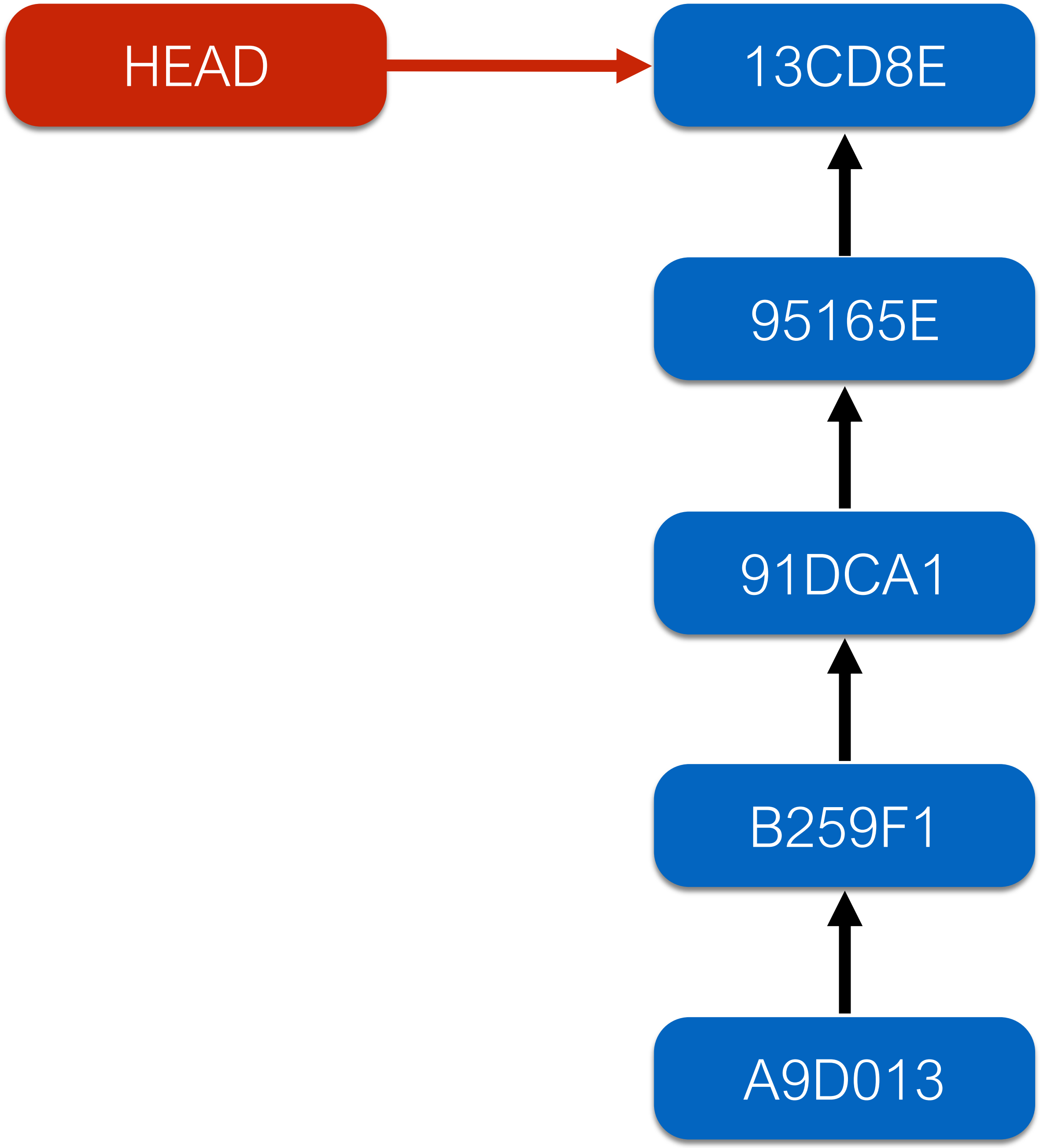
- Listing the log

  ```
  git log
  ```

- Listing each log in one line

  ```
  git log --oneline
  ```

Checking Out A Version

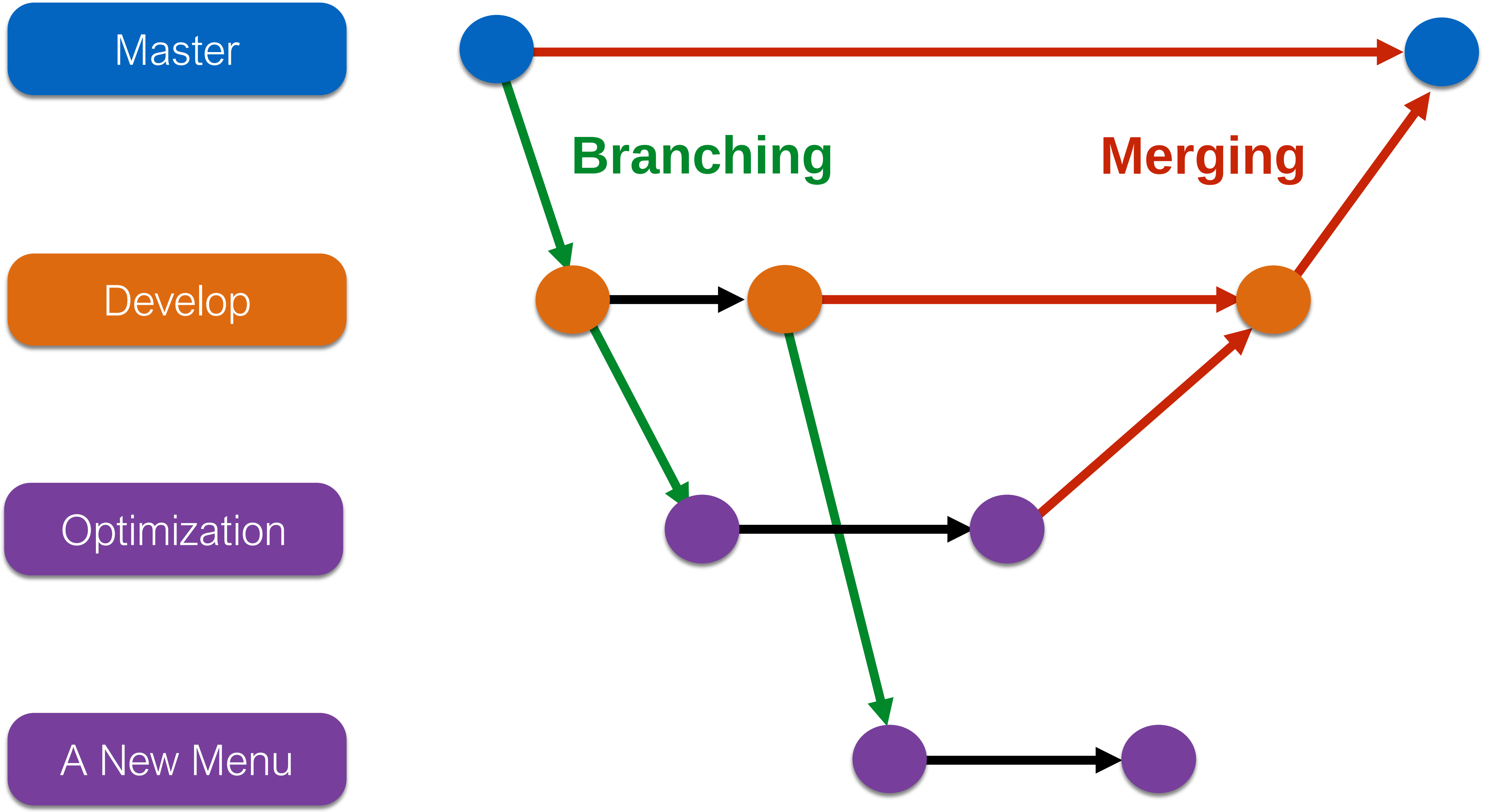


```
git checkout 91DCA1
```

Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Branches



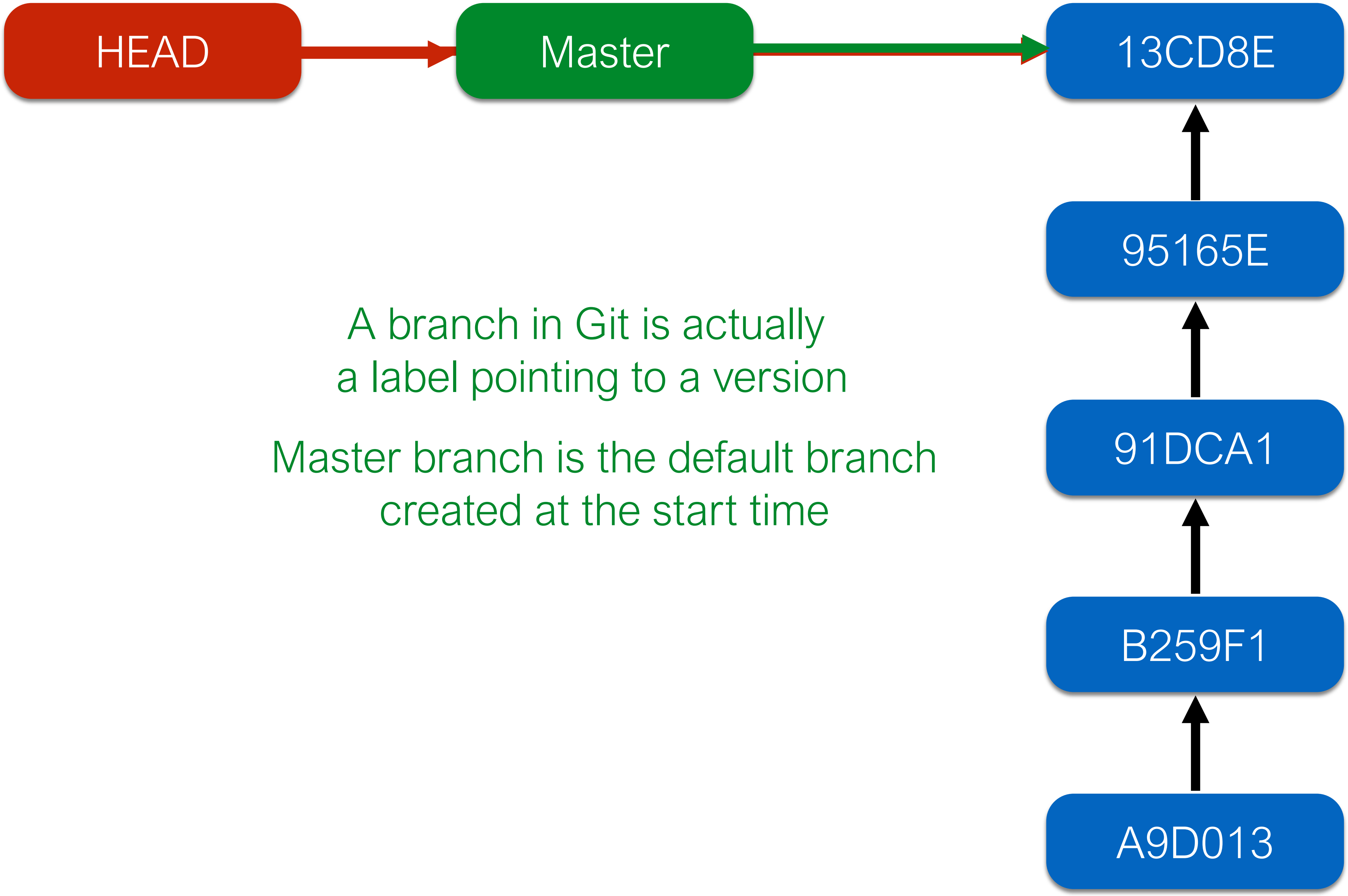
Master

Develop

Optimization

A New Menu

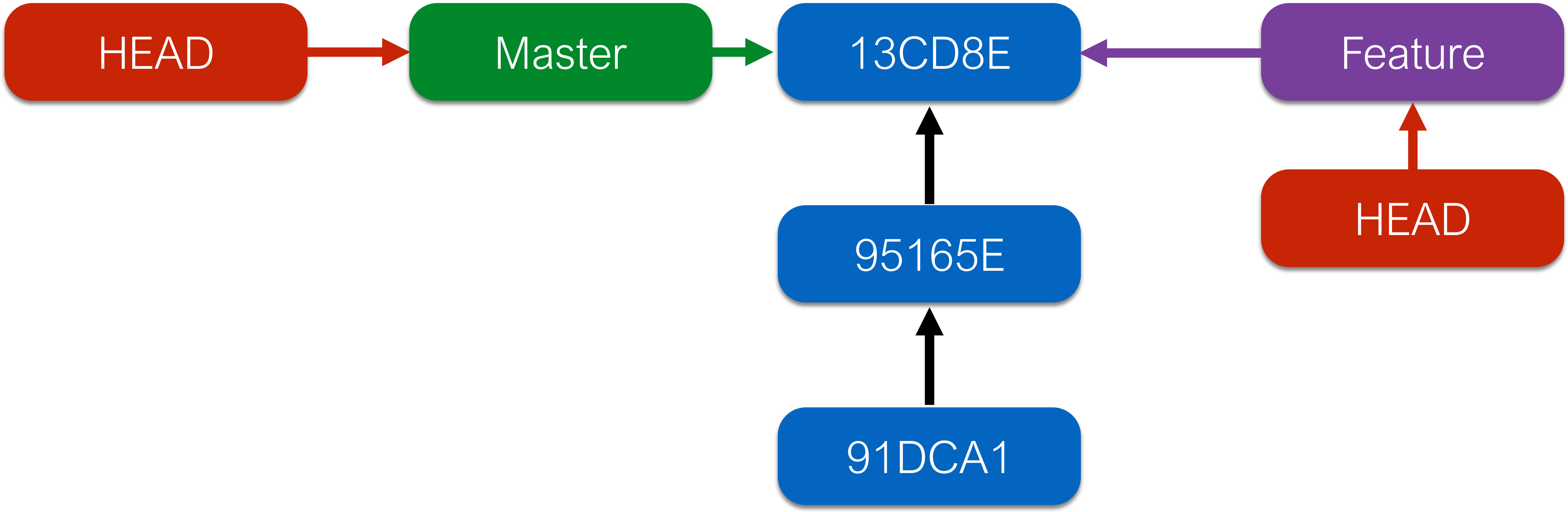
The Master Branch



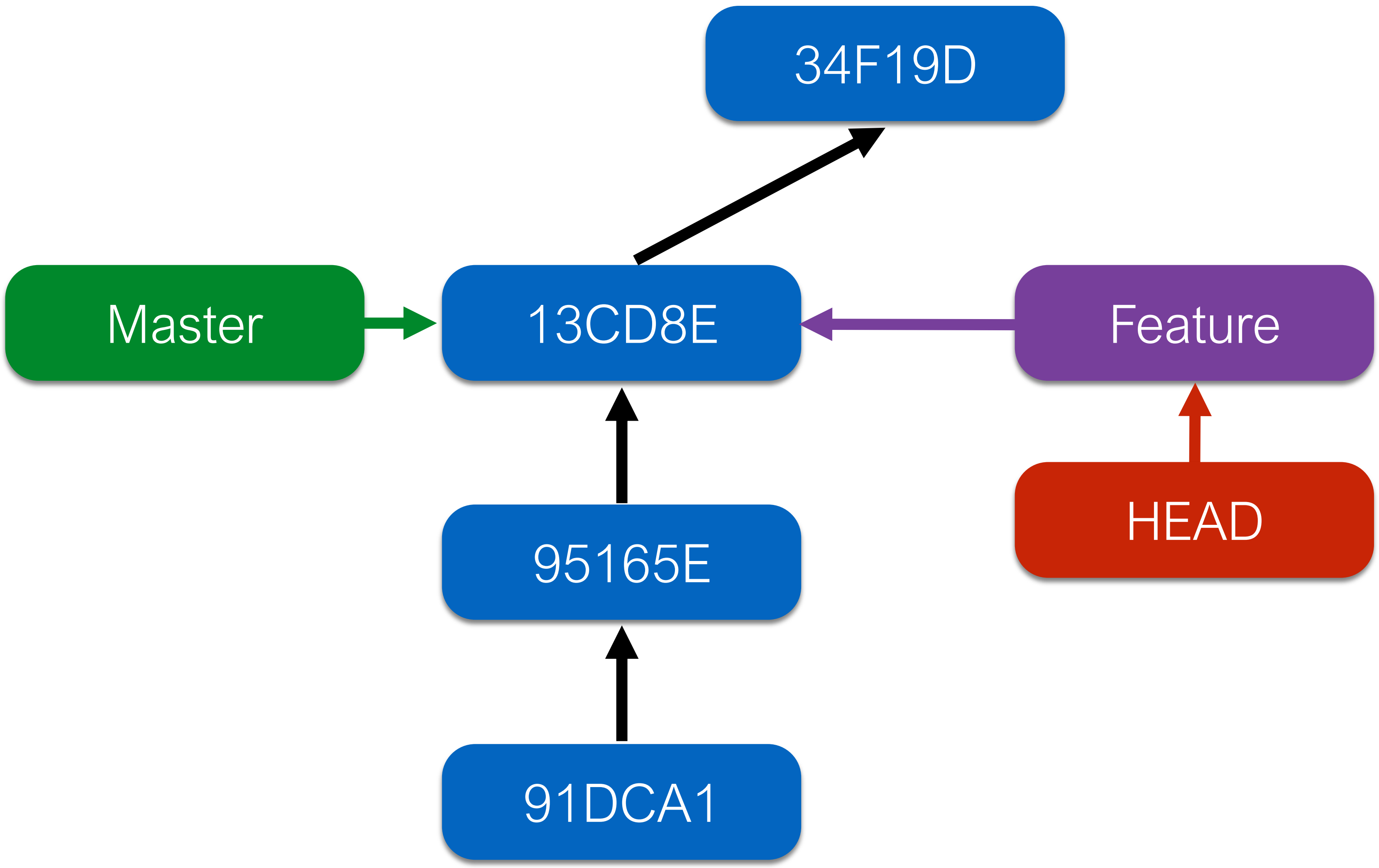
A branch in Git is actually
a label pointing to a version

Master branch is the default branch
created at the start time

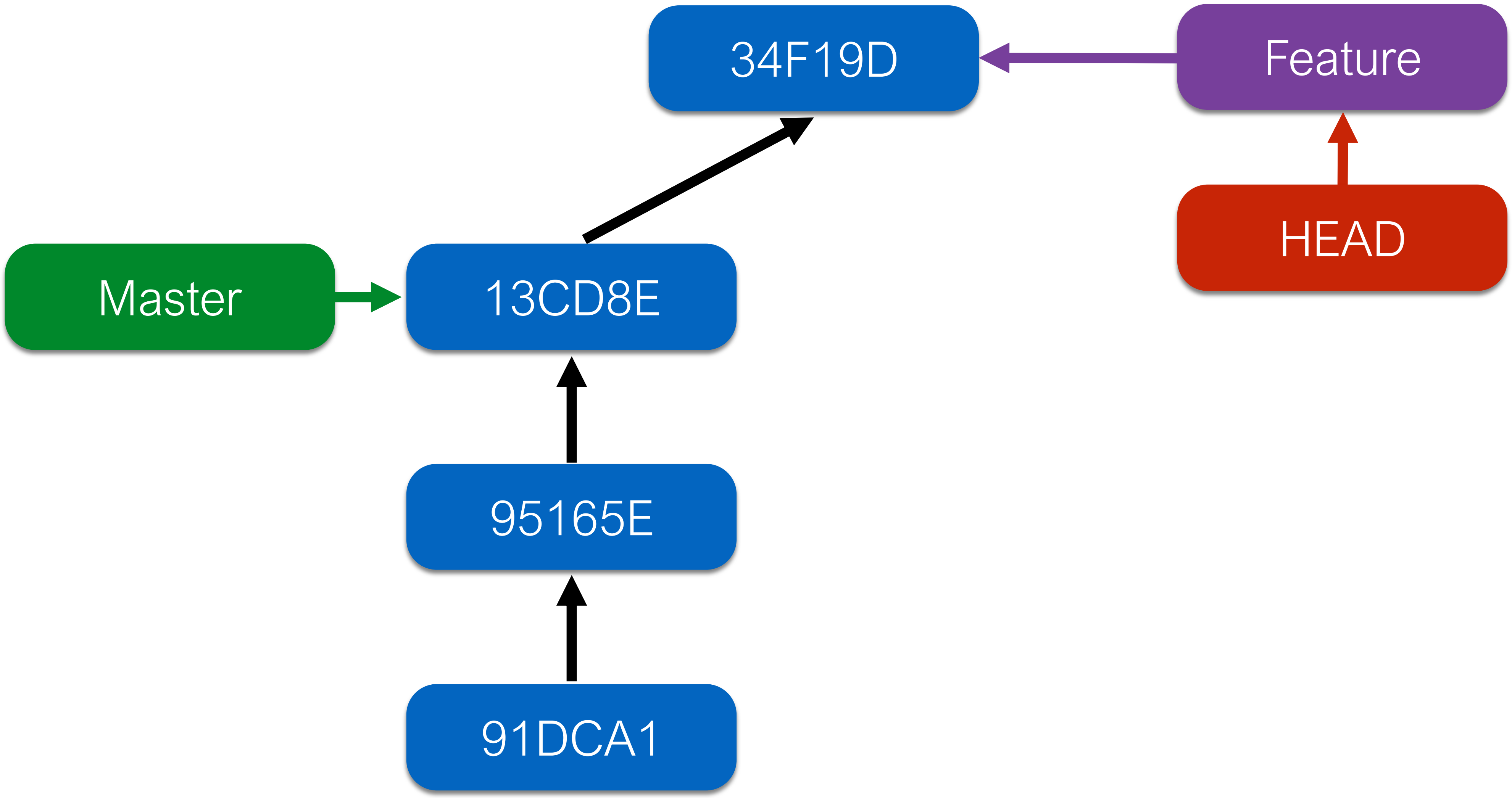
Branching



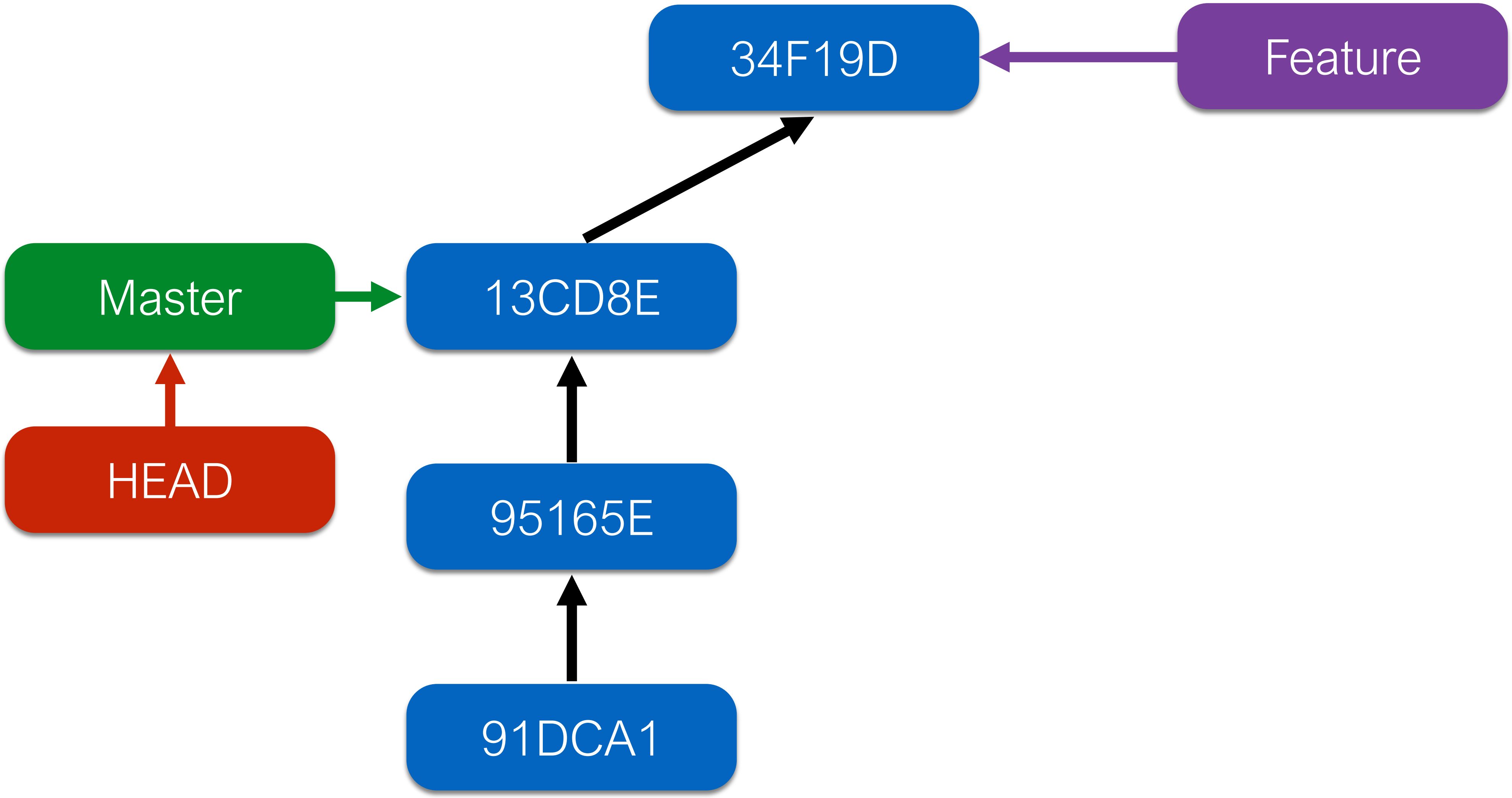
Branching



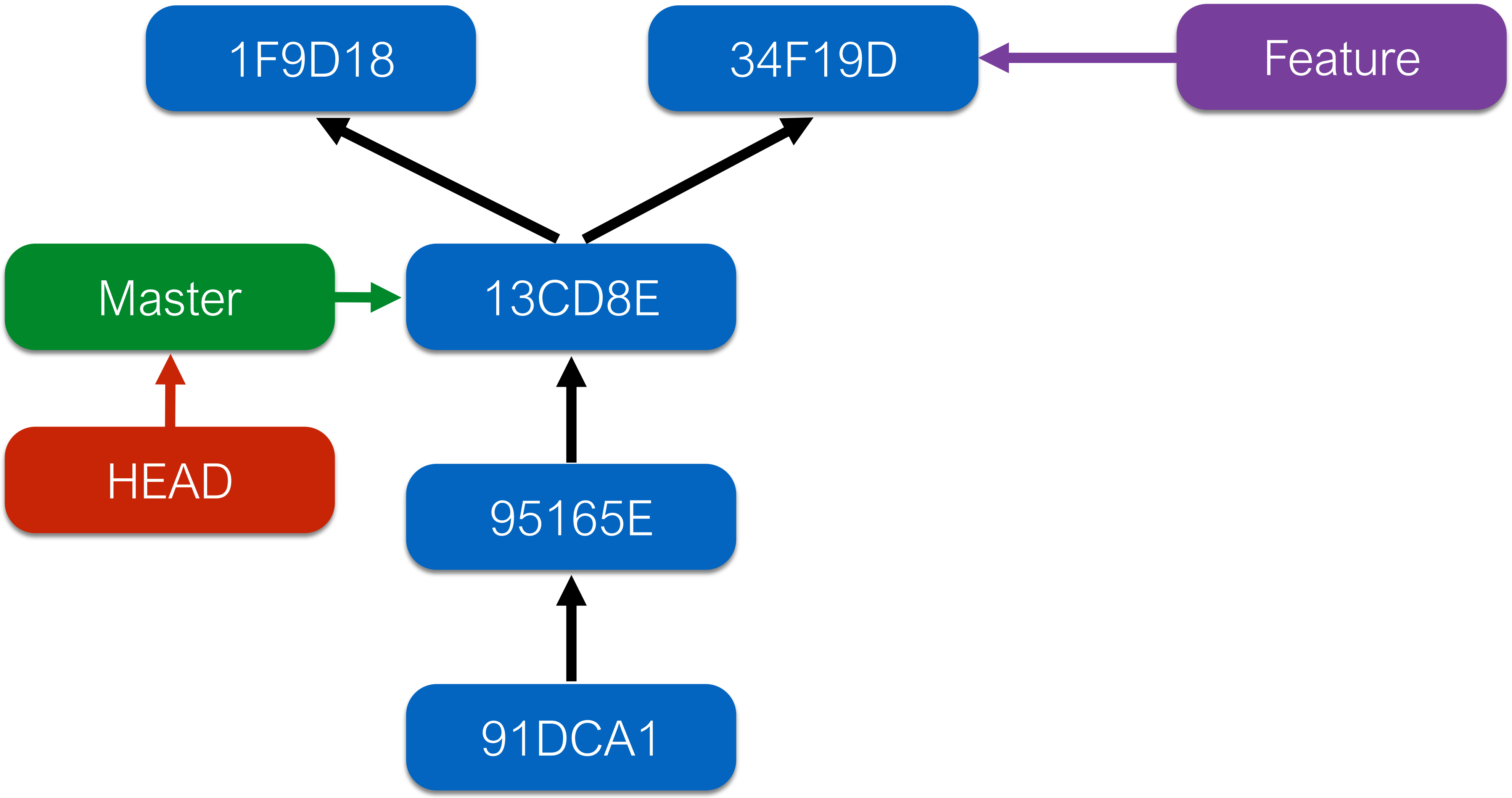
Branching



Branching



Branching



Git Branching

- Creating a new branch (label)

```
git branch [branch name]
```

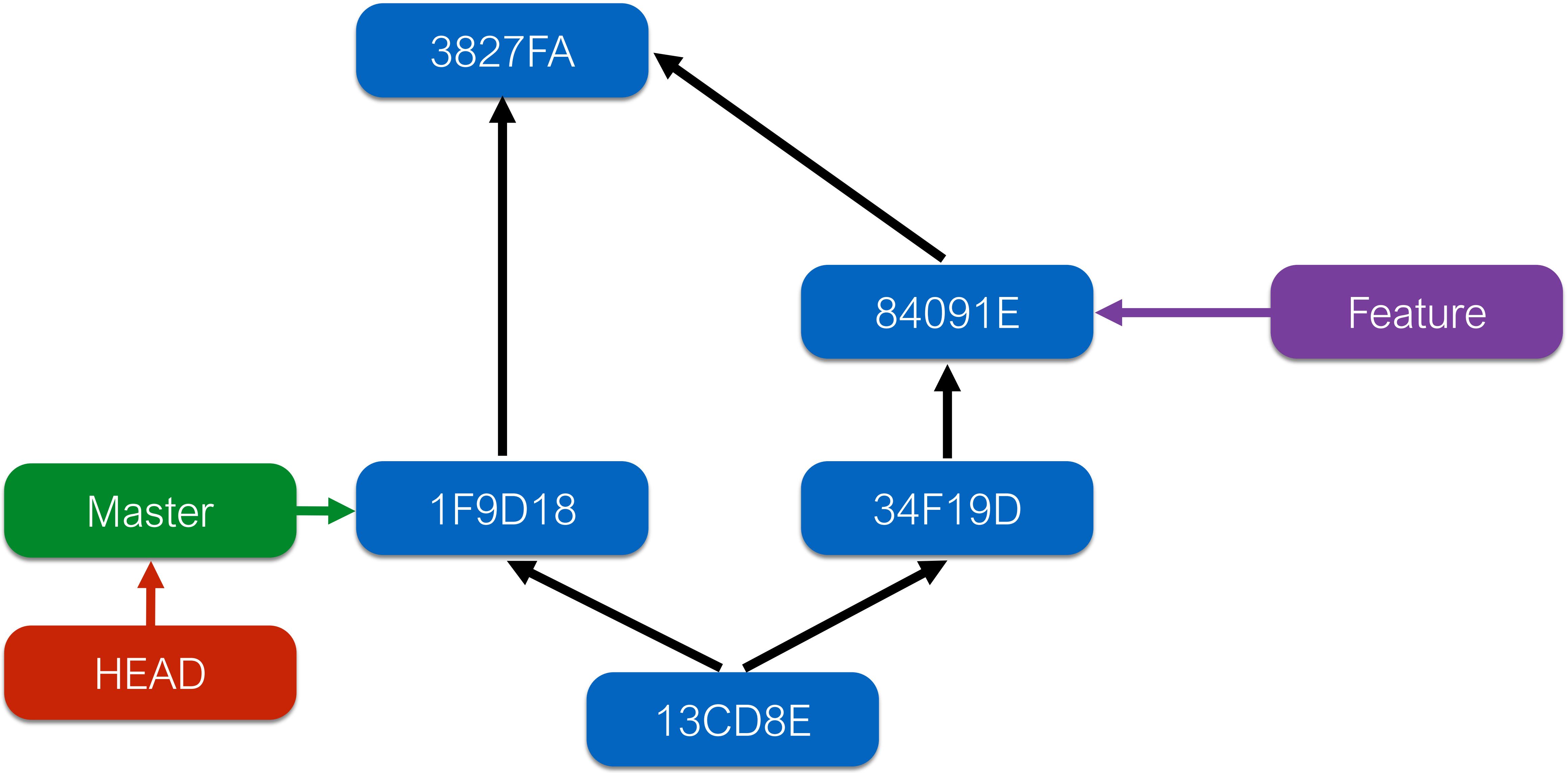
- Checking out the branch (move the HEAD)

```
git checkout [branch name]
```

- Combining the above commands (create & checkout)

```
git checkout -b [branch name]
```

Merging



3827FA

84091E

Feature

Master

1F9D18

34F19D

HEAD

13CD8E

Git Merging

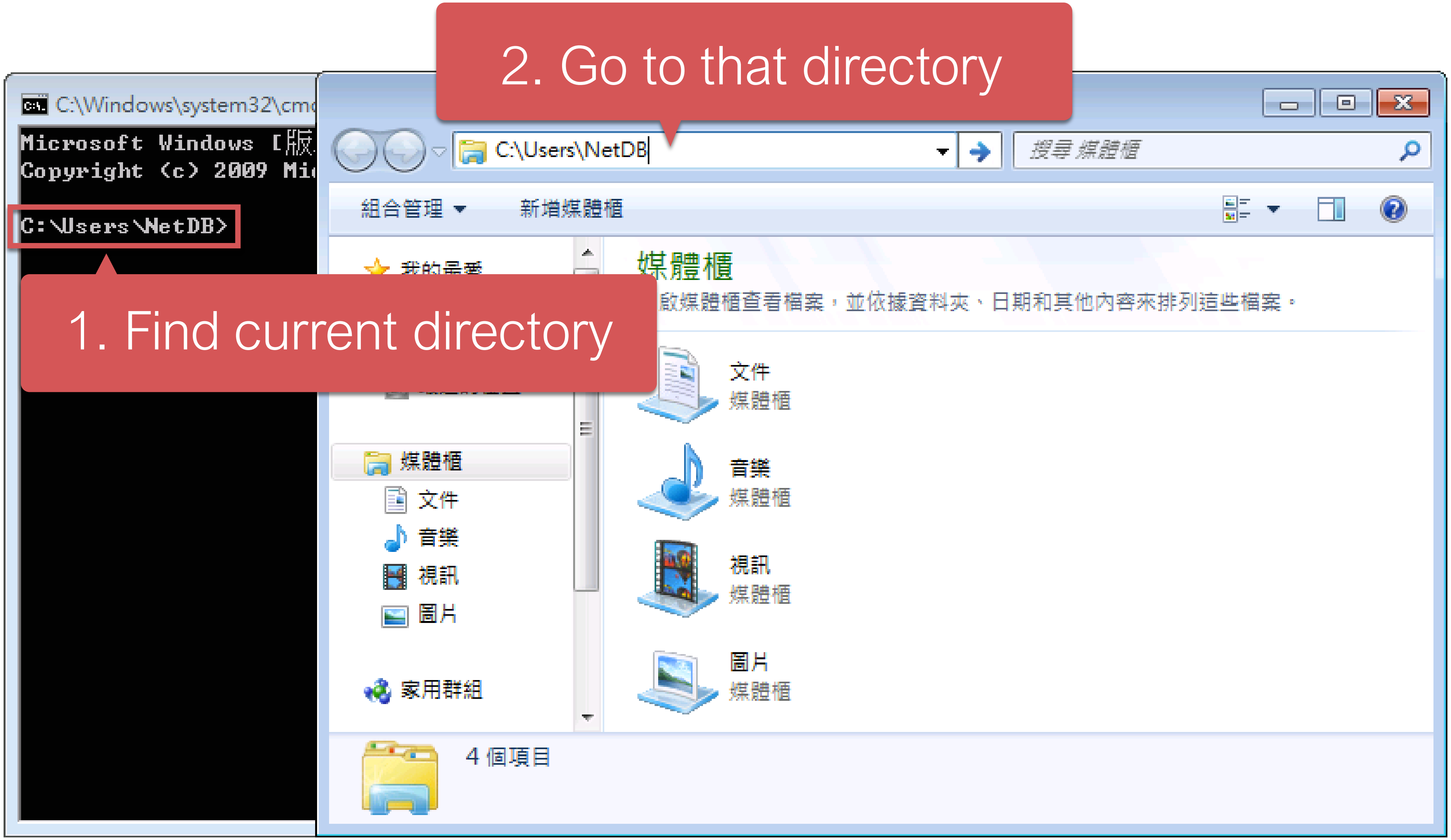
- Merging Steps
    - Checking out a branch to merge

      ```
      git checkout [branch 1 name]
      ```

    - Merging another branch

      ```
      git merge [branch 2 name]
      ```

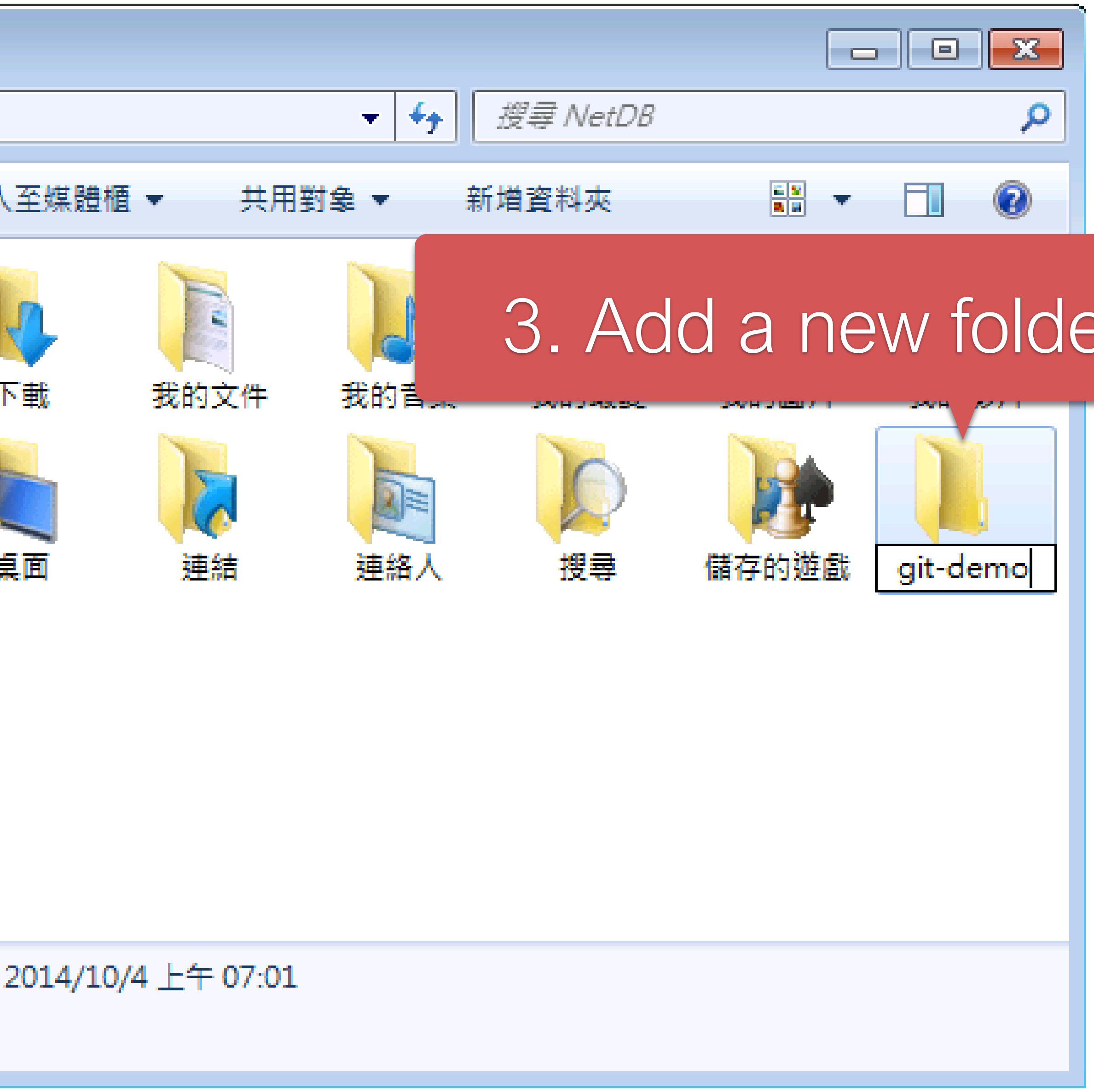
Try Git!



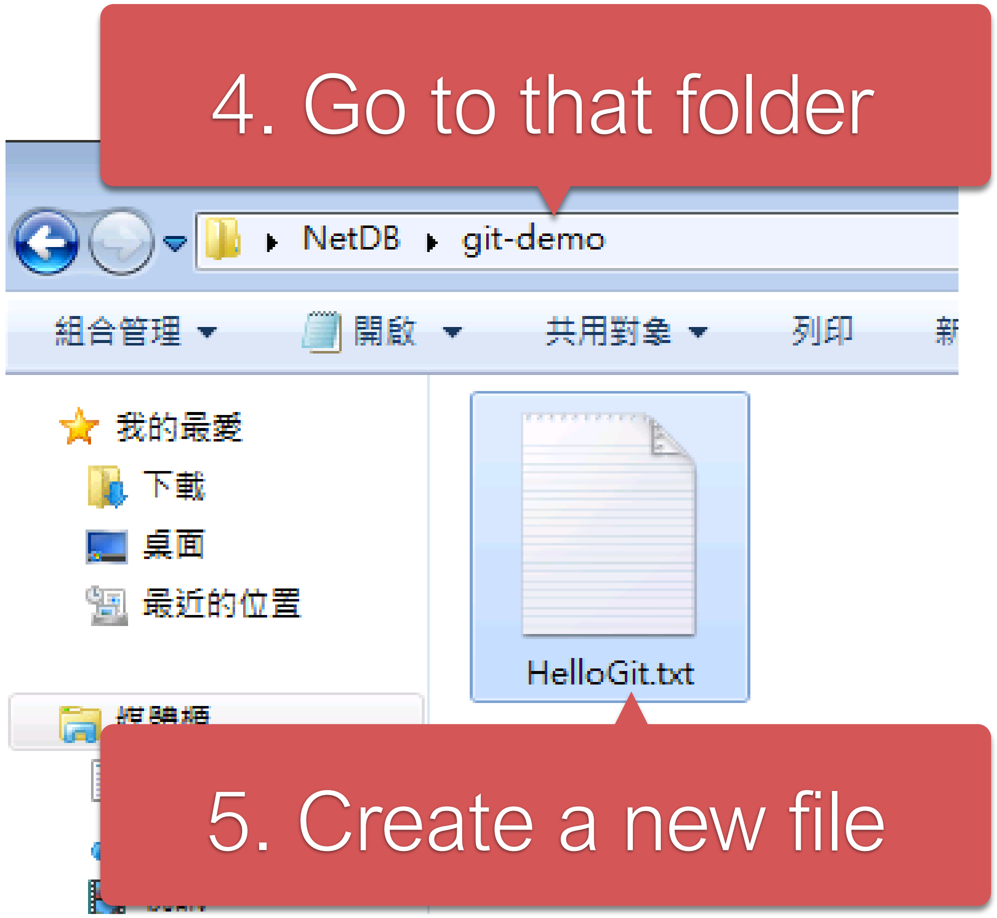
2. Go to that directory

1. Find current directory

```
C:\Windows\system32\cmd
Microsoft Windows [版
Copyright (c) 2009 Mi

C:\Users\NetDB>
```

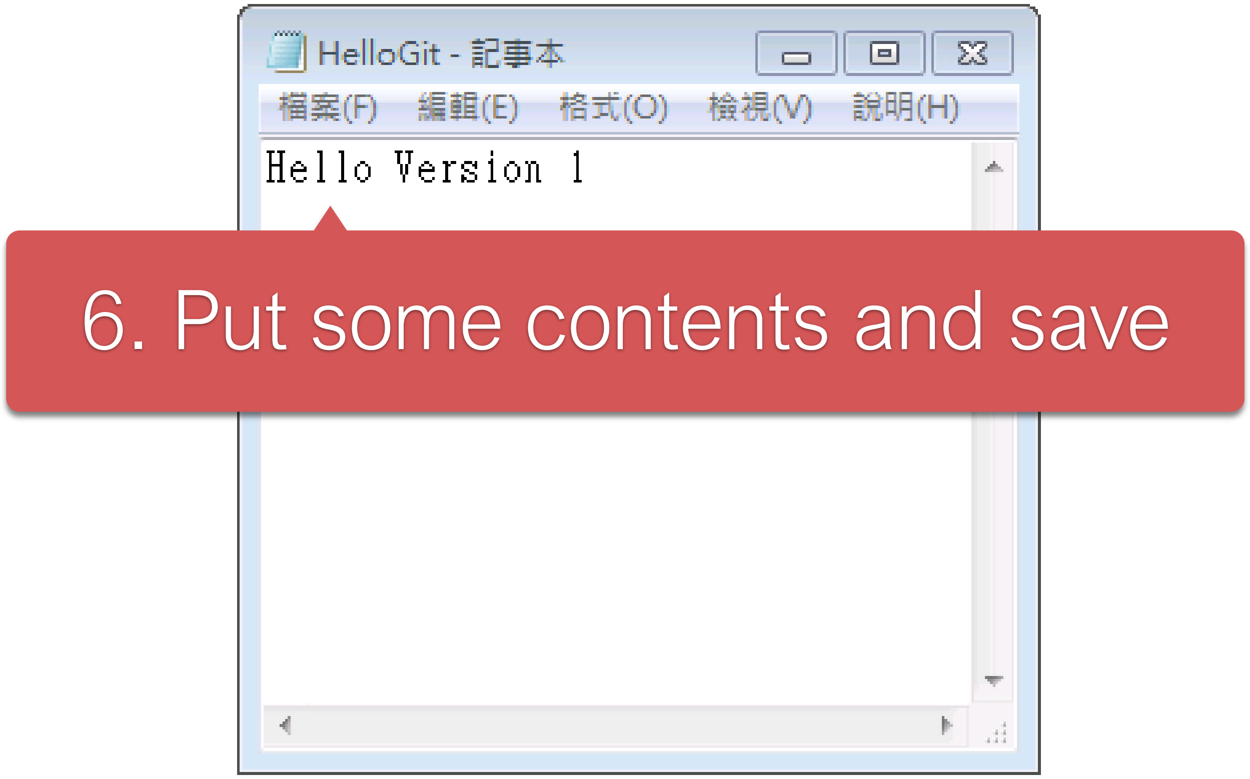


3. Add a new folder



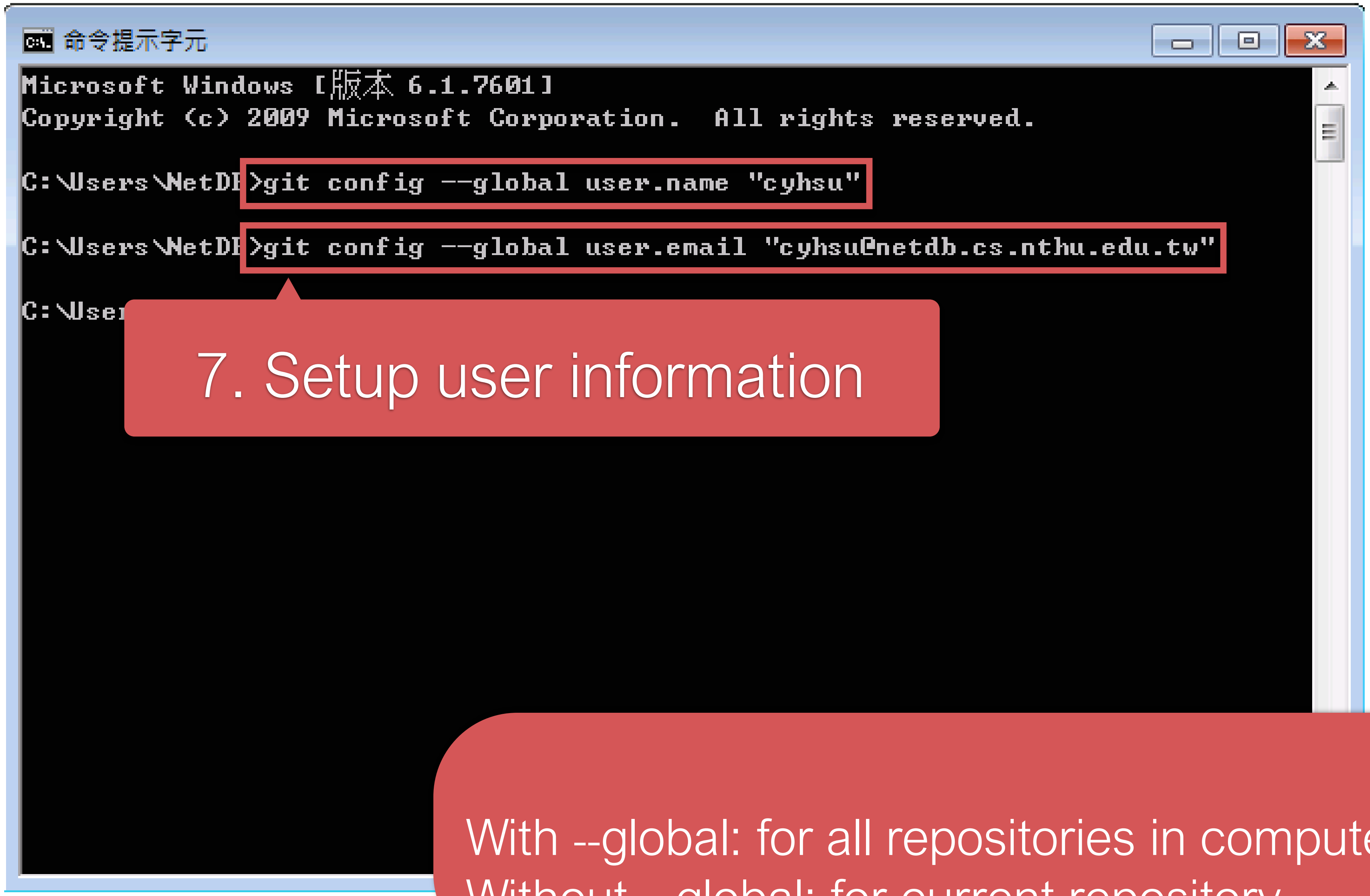
4. Go to that folder

5. Create a new file



```
Hello Version 1
```

6. Put some contents and save

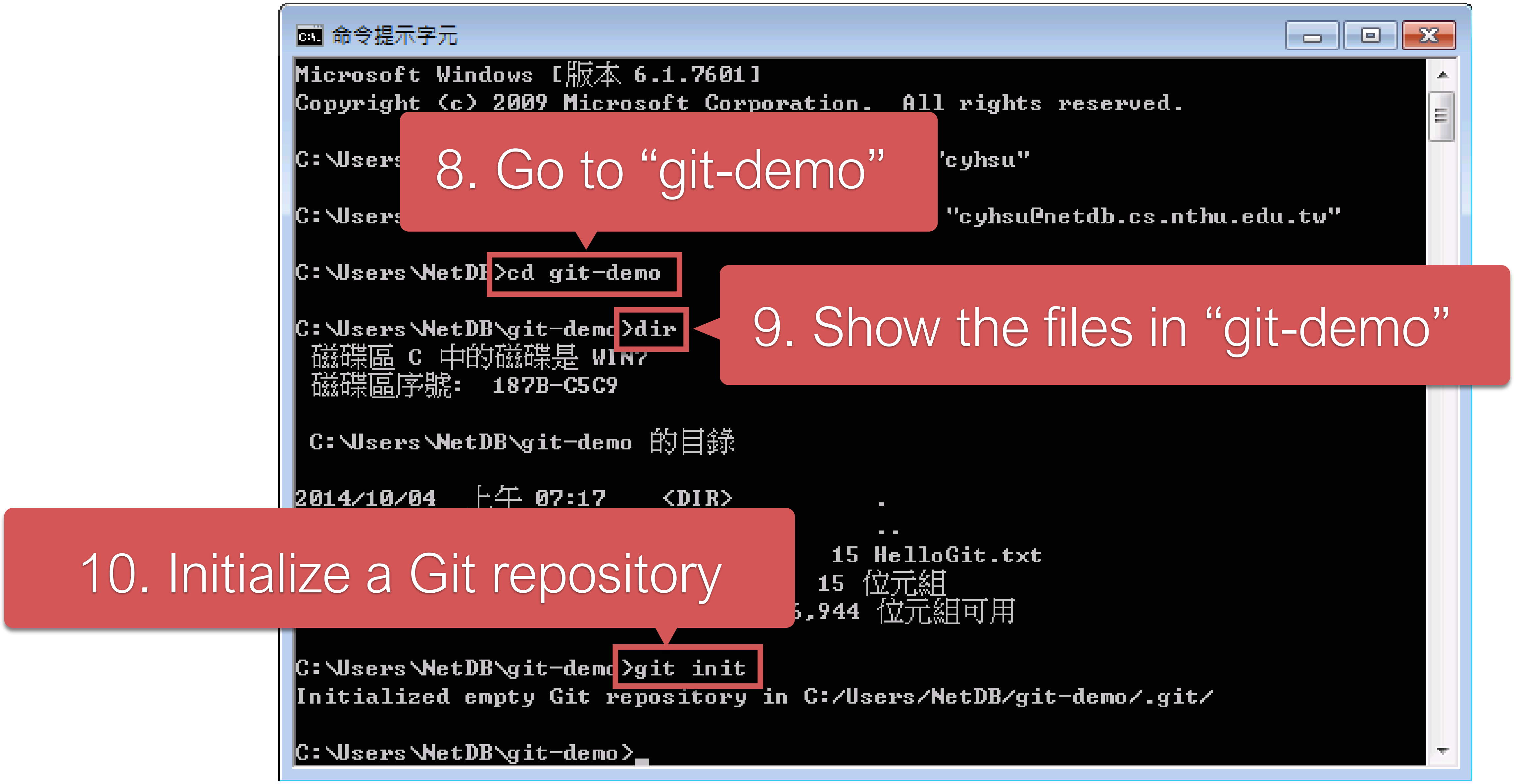


```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\NetDB>git config --global user.name "cyhsu"

C:\Users\NetDB>git config --global user.email "cyhsu@netdb.cs.nthu.edu.tw"

C:\User
```

7. Setup user information

With --global: for all repositories in computer
Without --global: for current repository

```
$ git config --global user.name "name"
$ git config --global user.email "email"
```



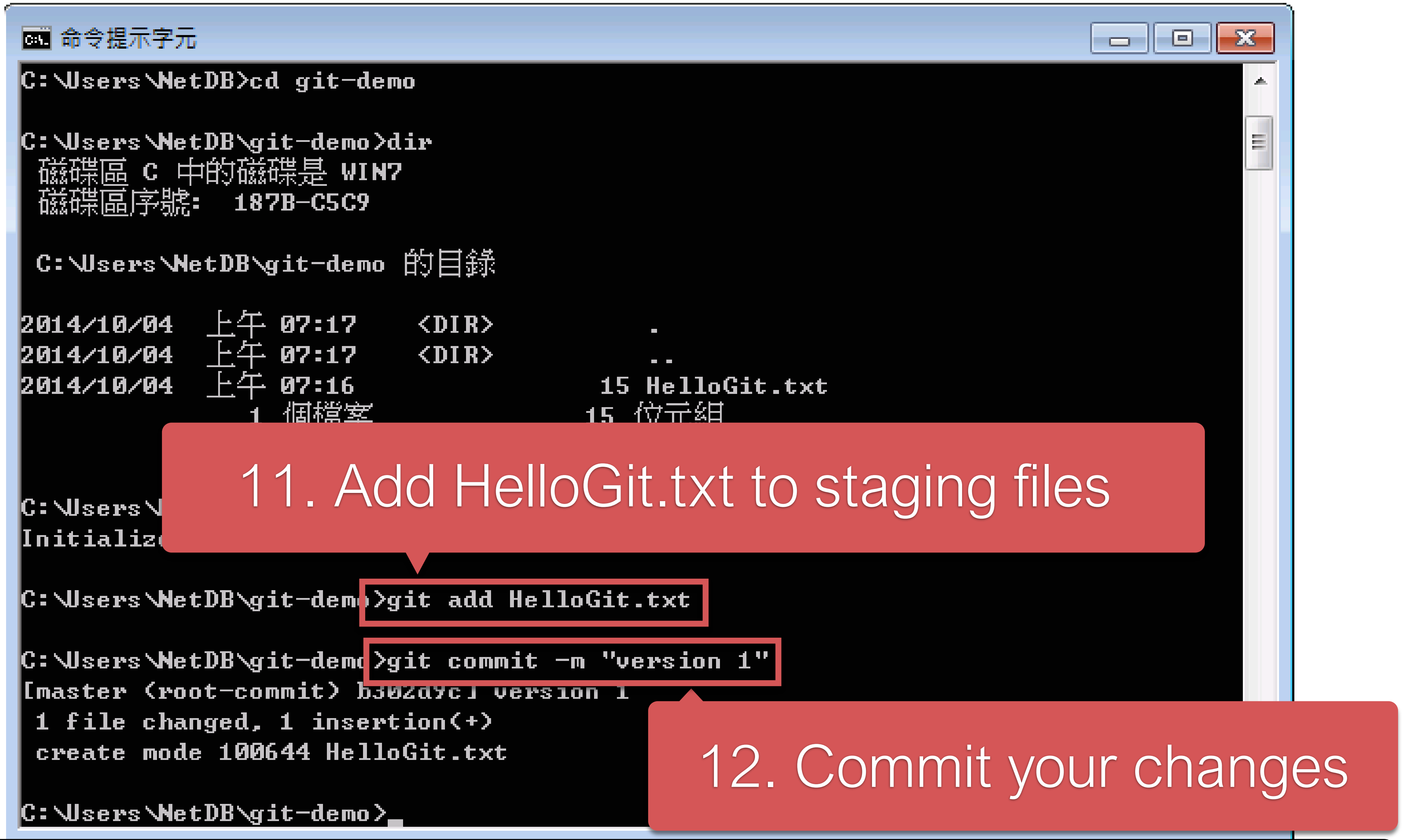
```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users                                    'cyhsu"

C:\Users                                     "cyhsu@netdb.cs.nthu.edu.tw"

C:\Users\NetDB>cd git-demo

C:\Users\NetDB\git-demo>dir
 磁碟區 C 中的磁碟是 WIN7
 磁碟區序號:  187B-C5C9

 C:\Users\NetDB\git-demo 的目錄

2014/10/04  上午 07:17    <DIR>          .
                                         ..
                                      15 HelloGit.txt
                                     15 位元組
                                 ,944 位元組可用

C:\Users\NetDB\git-demo>git init
Initialized empty Git repository in C:/Users/NetDB/git-demo/.git/

C:\Users\NetDB\git-demo>_
```

8. Go to “git-demo”

9. Show the files in “git-demo”

10. Initialize a Git repository

```
$ cd git-demo    # go to git-demo directory
$ dir            # list the files
$ git init       # initialize a repository
```

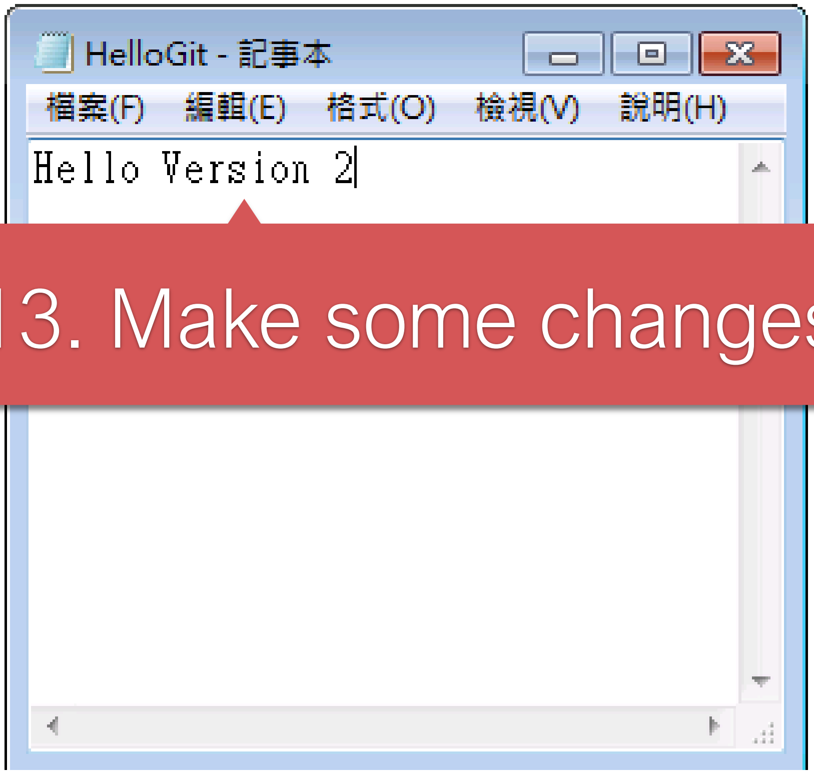


```
C:\Users\NetDB>cd git-demo

C:\Users\NetDB\git-demo>dir
 磁碟區 C 中的磁碟是 WIN7
 磁碟區序號:  187B-C5C9

 C:\Users\NetDB\git-demo 的目錄

2014/10/04  上午 07:17    <DIR>          .
2014/10/04  上午 07:17    <DIR>          ..
2014/10/04  上午 07:16                15 HelloGit.txt
               1 個檔案              15 位元組


C:\Users\
Initializ

C:\Users\NetDB\git-demo>git add HelloGit.txt

C:\Users\NetDB\git-demo>git commit -m "version 1"
[master (root-commit) b302d9c] version 1
 1 file changed, 1 insertion(+)
 create mode 100644 HelloGit.txt

C:\Users\NetDB\git-demo>_
```

11. Add HelloGit.txt to staging files

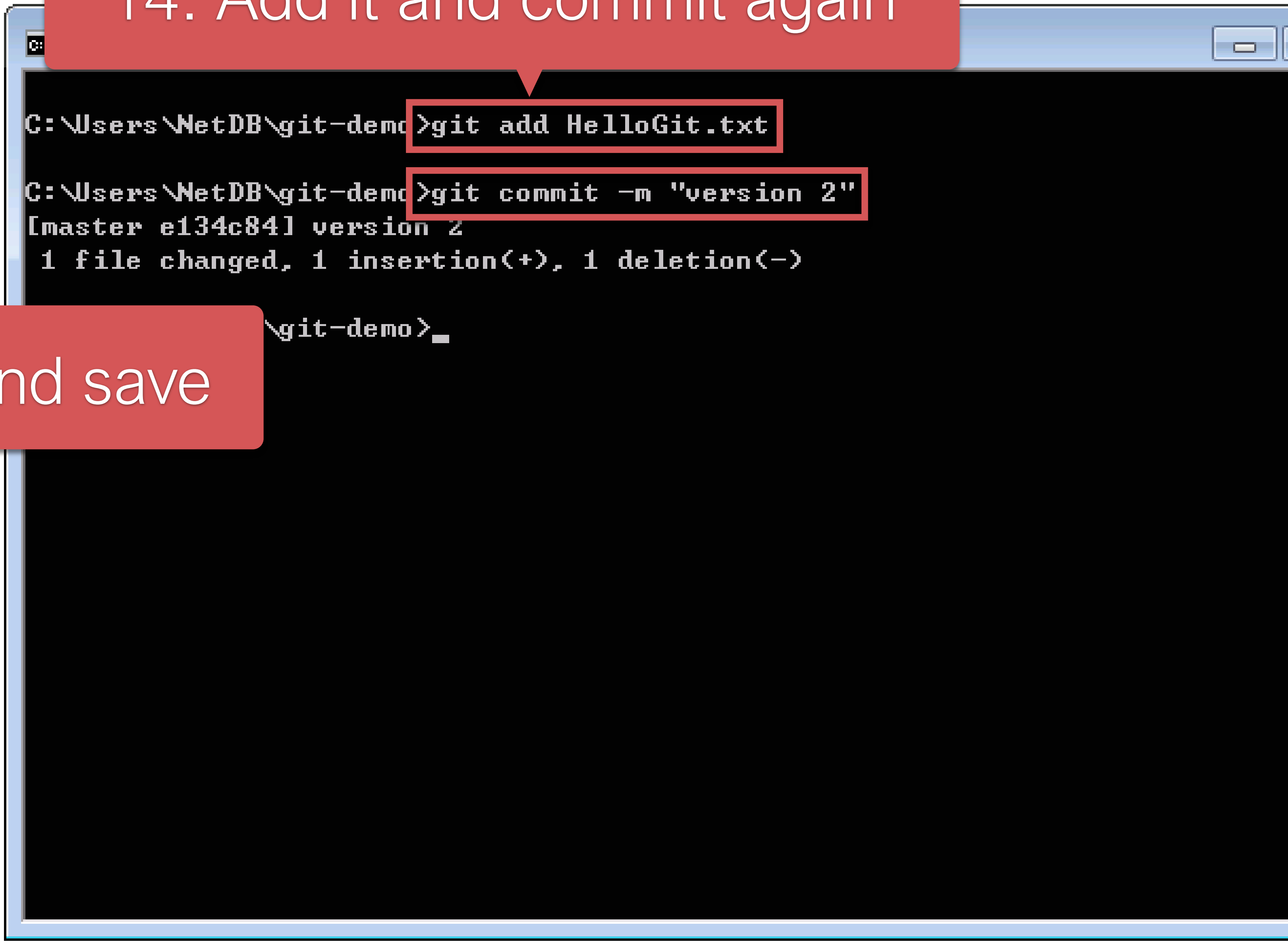
12. Commit your changes

```
# Add HelloGit.txt to staging files
$ git add HelloGit.txt

# Commit the changes to the repository
# where "version 1" is the commit message
$ git commit -m "version 1"
```



```
Hello Version 2
```

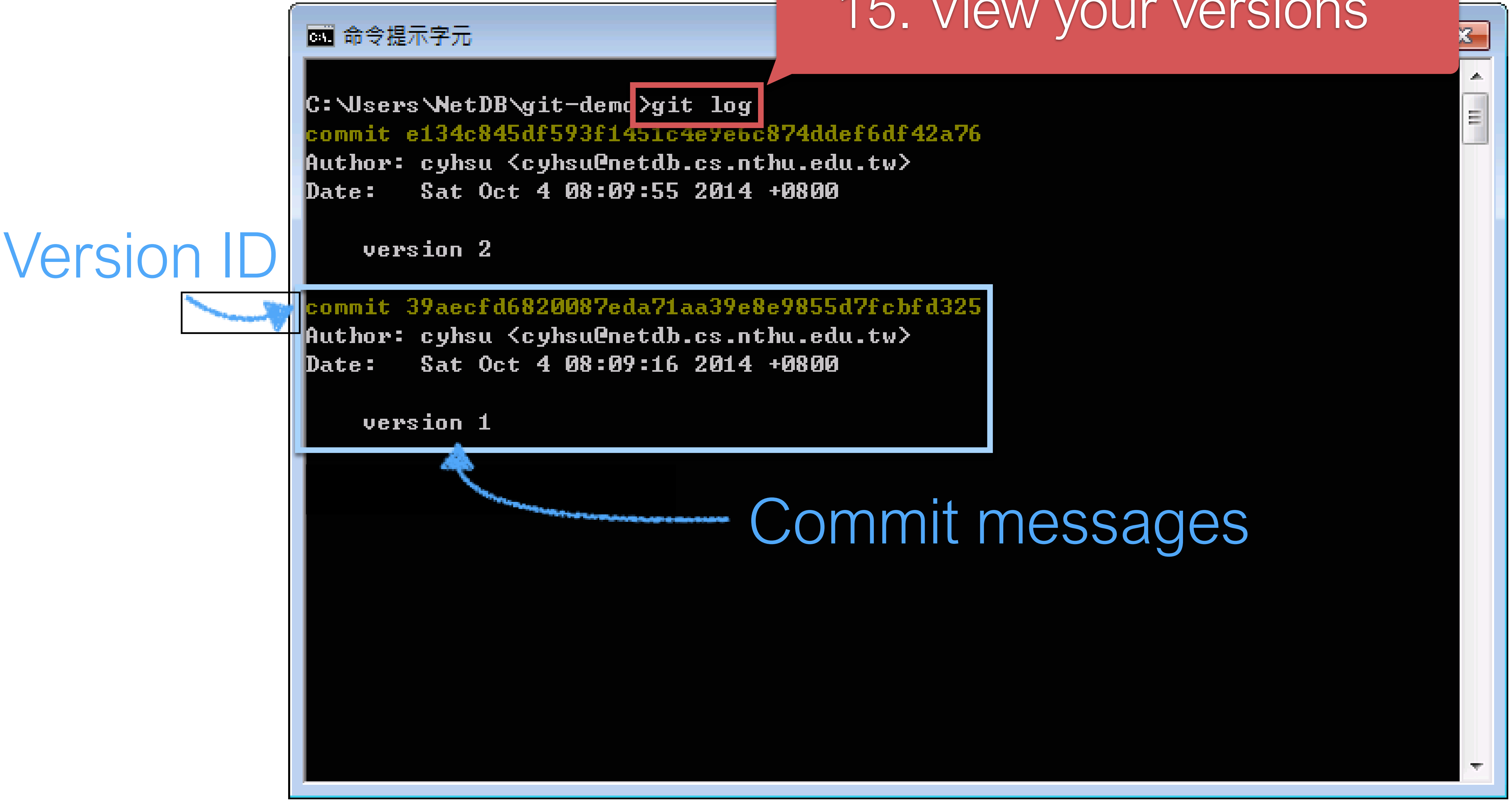
13. Make some changes and save



```
C:\Users\NetDB\git-demo>git add HelloGit.txt

C:\Users\NetDB\git-demo>git commit -m "version 2"
[master e134c84] version 2
 1 file changed, 1 insertion(+), 1 deletion(-)

                  \git-demo>_
```

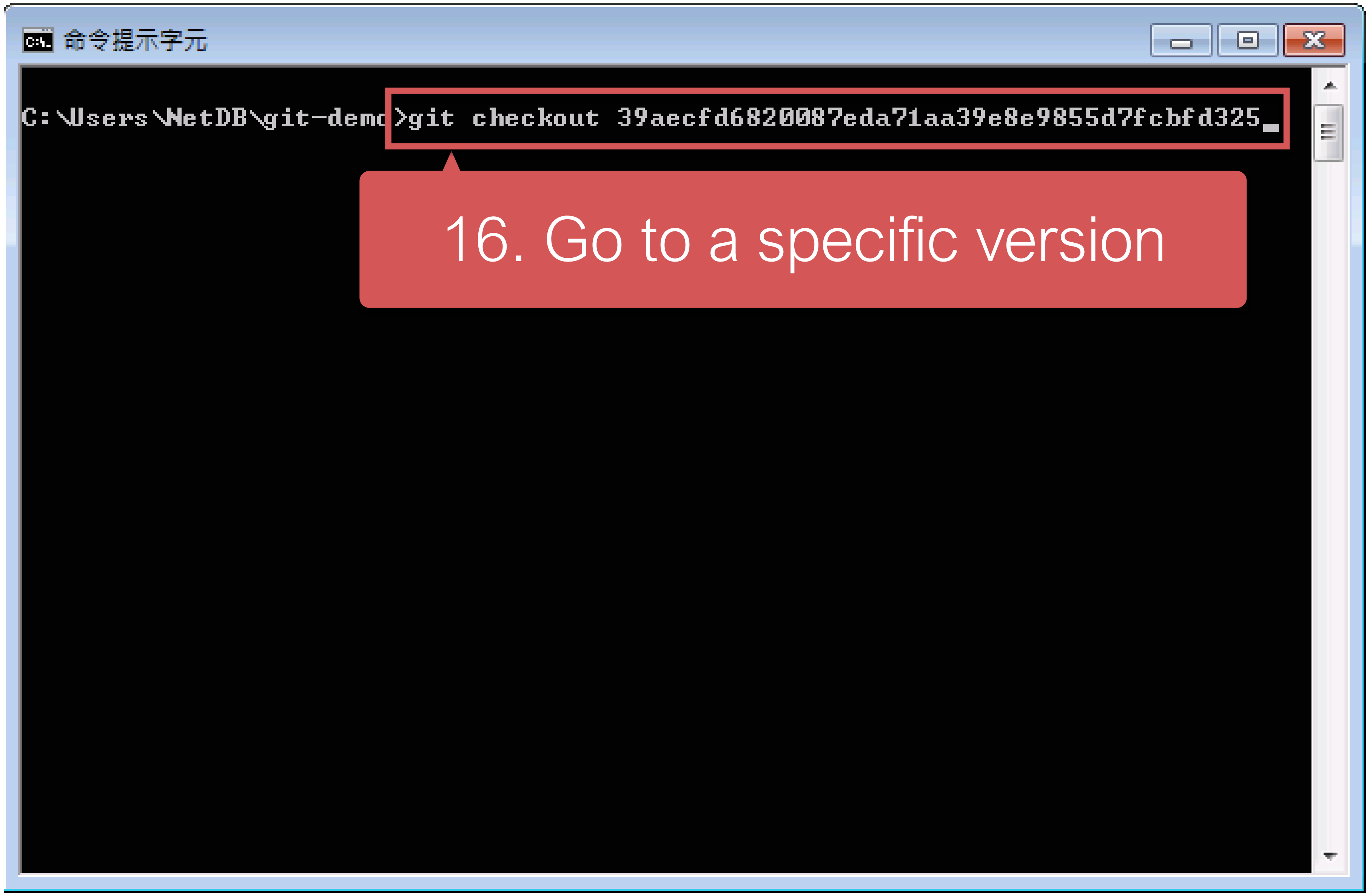
14. Add it and commit again

15. View your versions



```
C:\Users\NetDB\git-demo>git log
commit e134c845df593f1451c4e9e6c874ddef6df42a76
Author: cyhsu <cyhsu@netdb.cs.nthu.edu.tw>
Date:   Sat Oct 4 08:09:55 2014 +0800

    version 2

commit 39aecfd6820087eda71aa39e8e9855d7fcbfd325
Author: cyhsu <cyhsu@netdb.cs.nthu.edu.tw>
Date:   Sat Oct 4 08:09:16 2014 +0800

    version 1
```

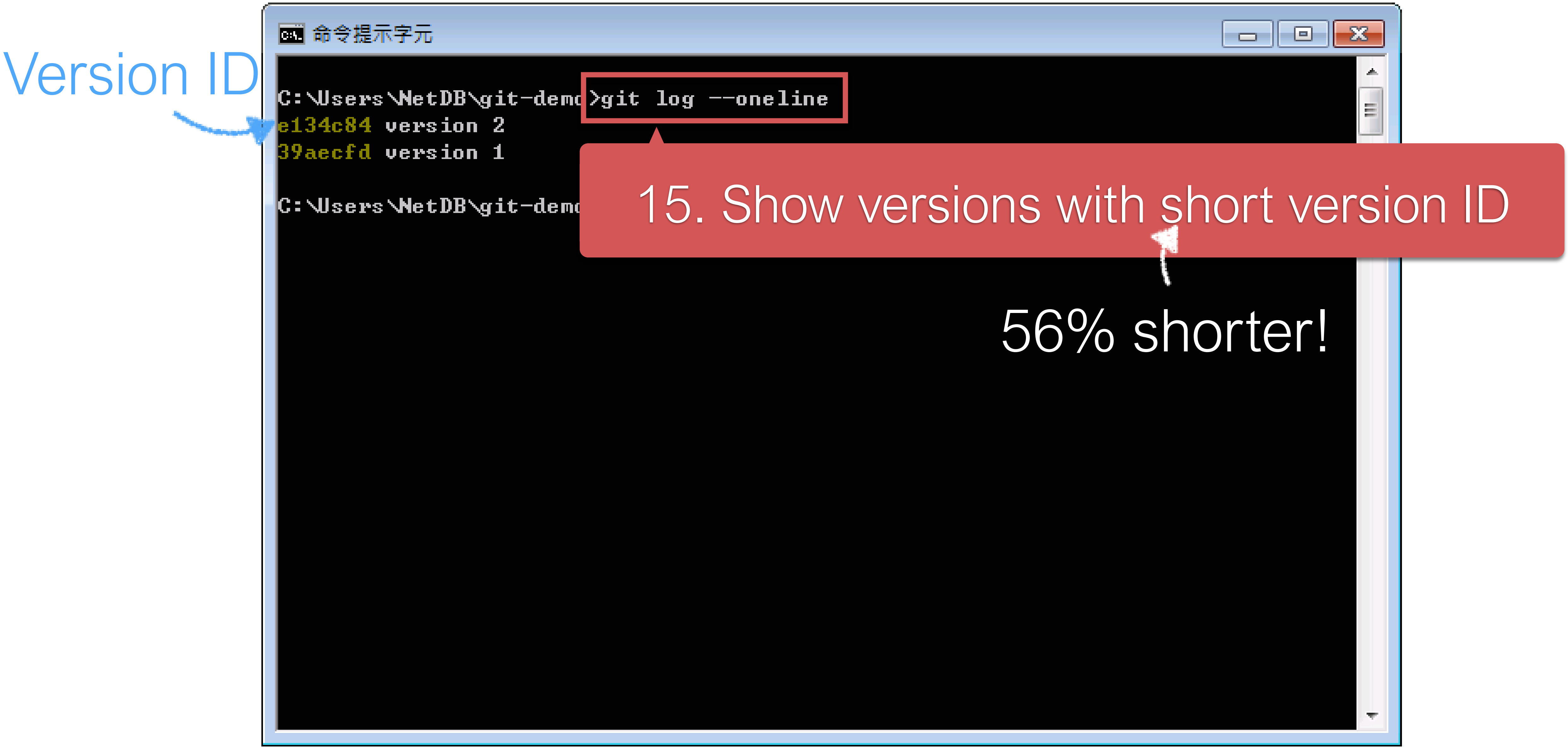
```
# Show the versions you've created so far
$ git log
```



```
C:\Users\NetDB\git-demo>git checkout 39aecfd6820087eda71aa39e8e9855d7fcbfd325
```

16. Go to a specific version

```
# Go to a specific version
$ git checkout {version_id}
```

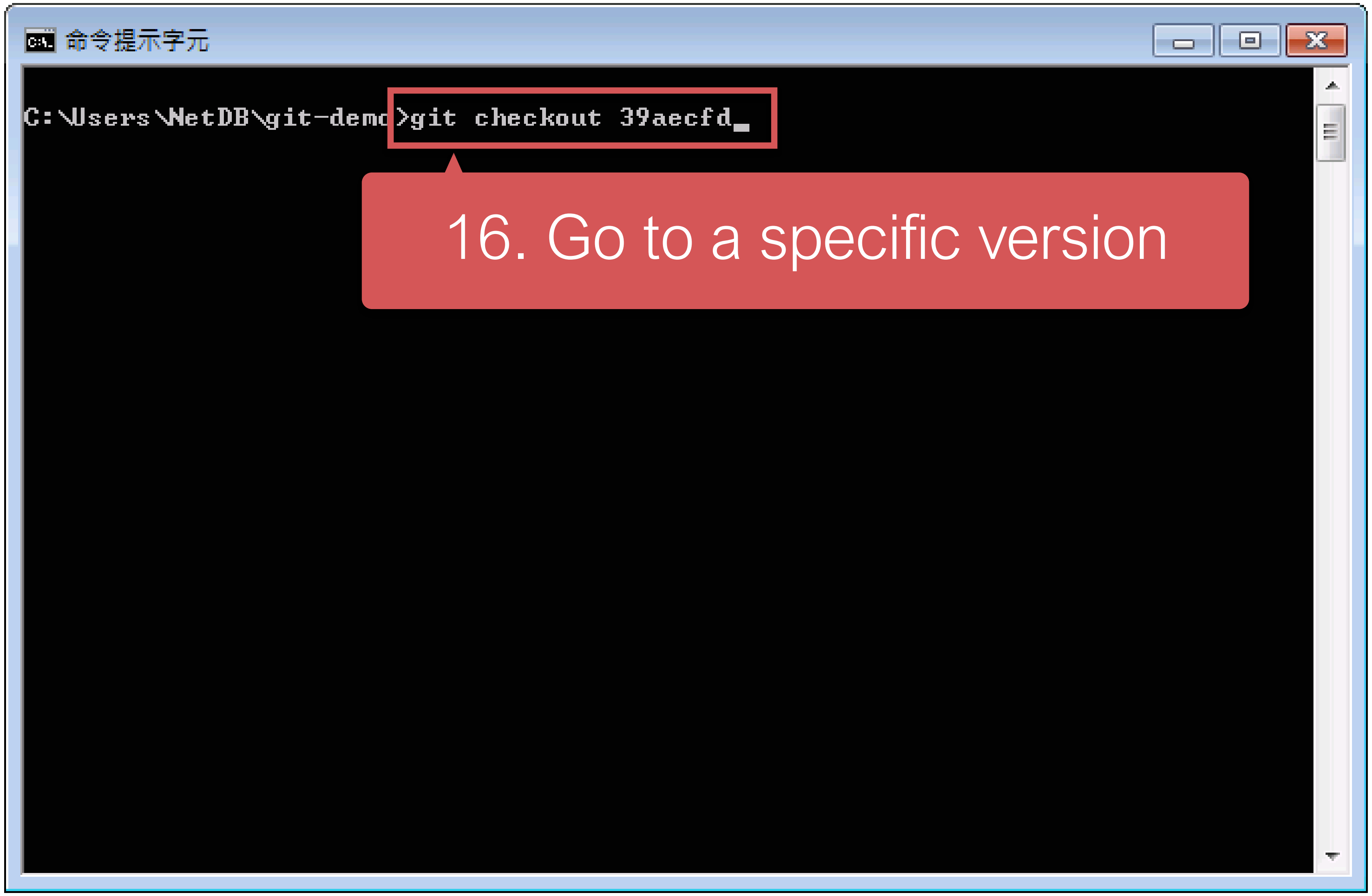


```
C:\Users\NetDB\git-demo>git log --oneline
e134c84 version 2
39aecfd version 1

C:\Users\NetDB\git-demo
```

15. Show versions with short version ID

56% shorter!

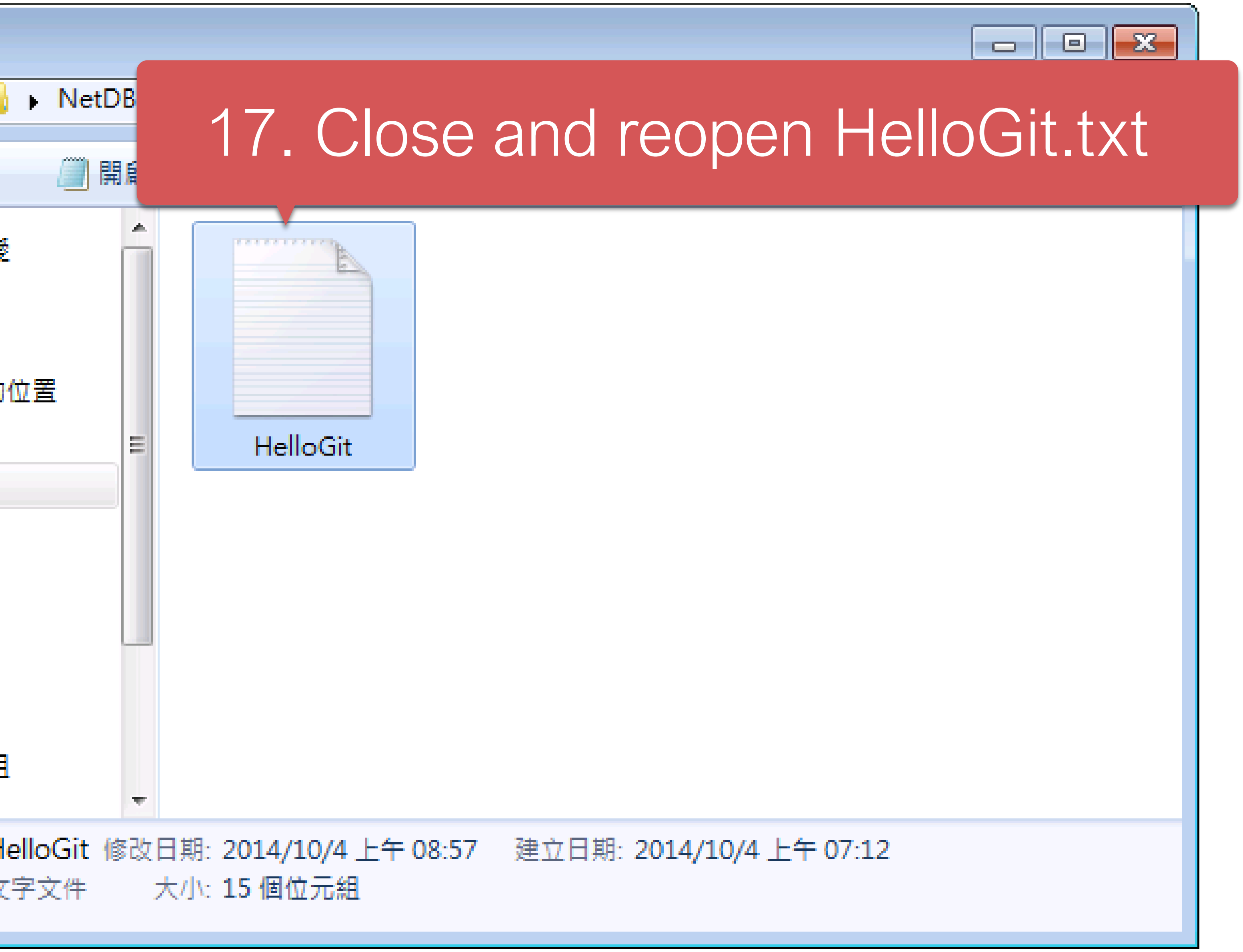
```
# Show versions with short version id
$ git log --oneline
```



```
C:\Users\NetDB\git-demo>git checkout 39aecfd_
```

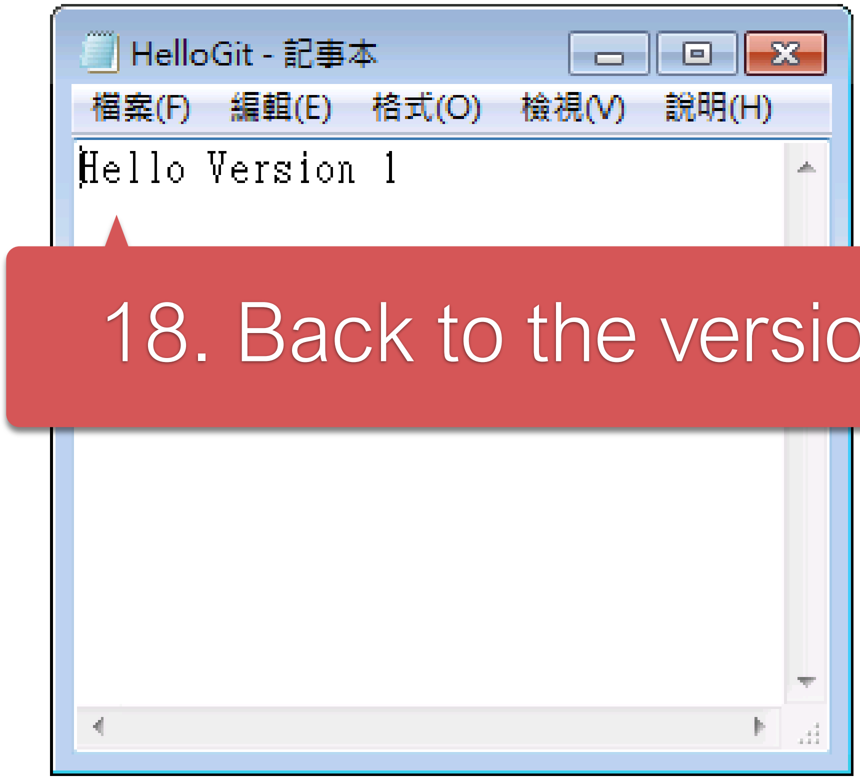
16. Go to a specific version

```
# Go to a specific version.
# In fact, you only need to type
# the first 5 characters.
$ git checkout {short_version_id}
```



17. Close and reopen HelloGit.txt

HelloGit 修改日期: 2014/10/4 上午 08:57 建立日期: 2014/10/4 上午 07:12
文字文件 大小: 15 個位元組



```
Hello Version 1
```

18. Back to the version 1 !

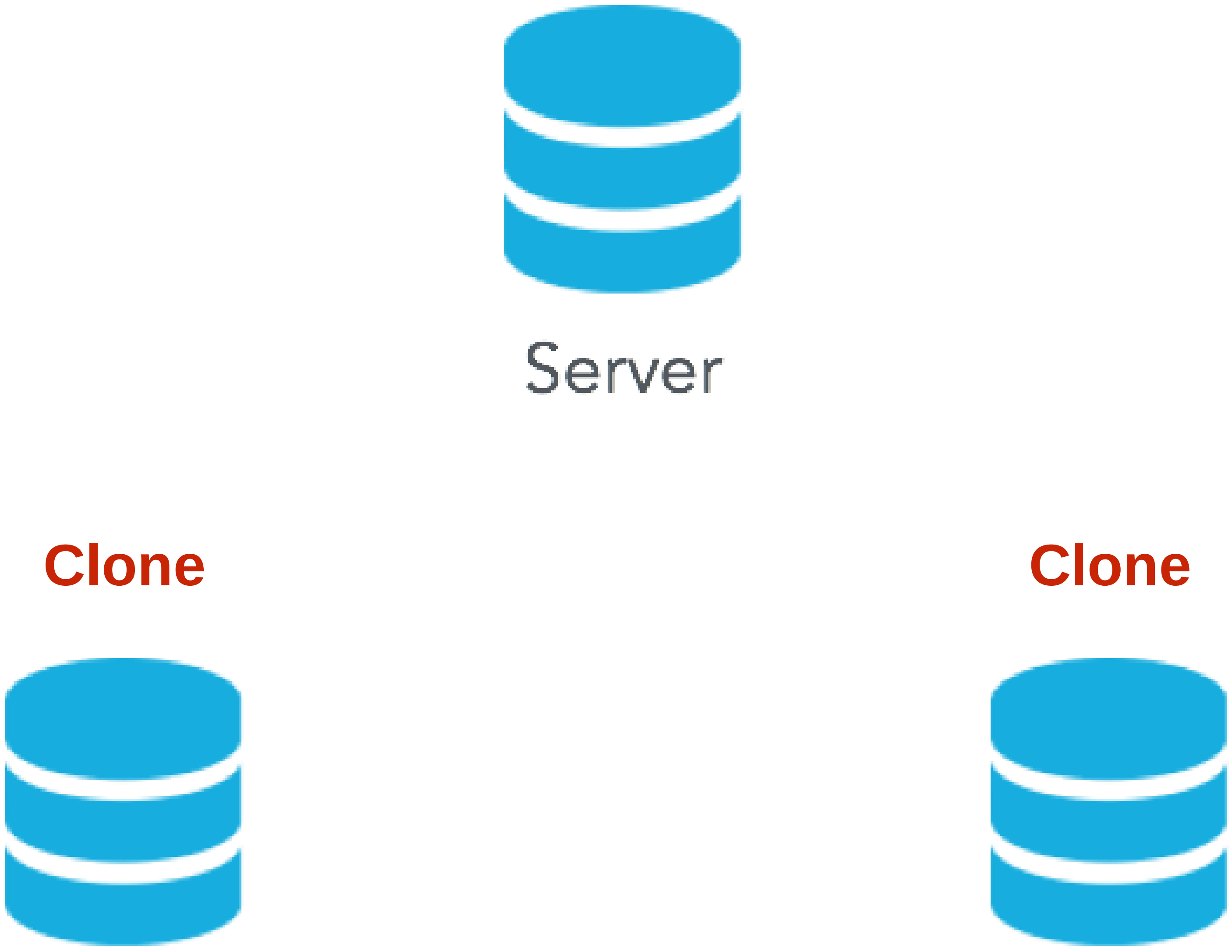
Outline

- Version control system
- Git basics
- Git branch
- Remote repository

Collaboration with Git

- To work with others using git, you’ ll need a server that store the repository.
- Git is distributed, which means
  - Everyone can store a copy of the repository downloaded from the server
  - They can do their jobs independently

Collaboration workflow



Server

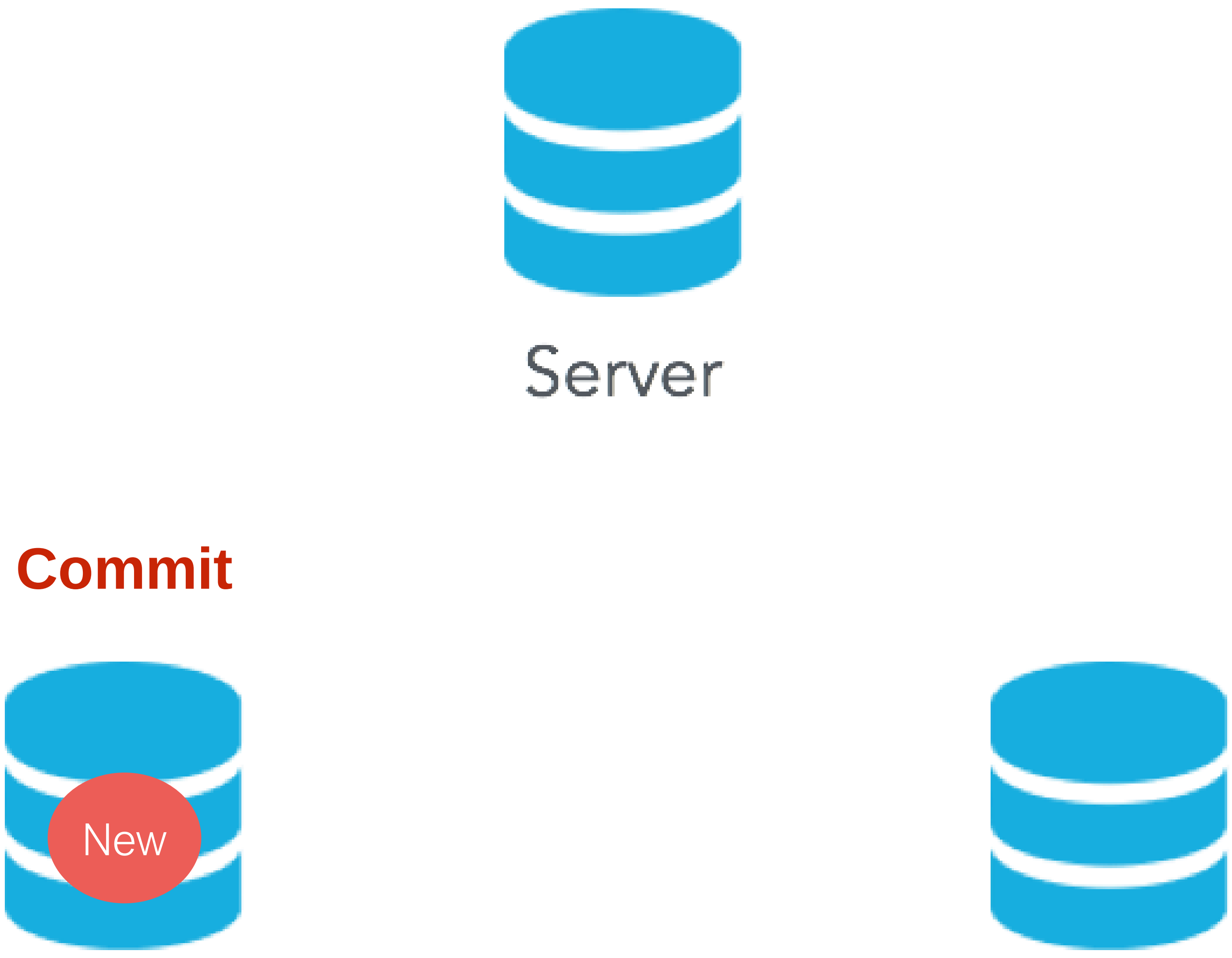
Clone

Clone

Local A

Local B

Collaboration workflow



Server

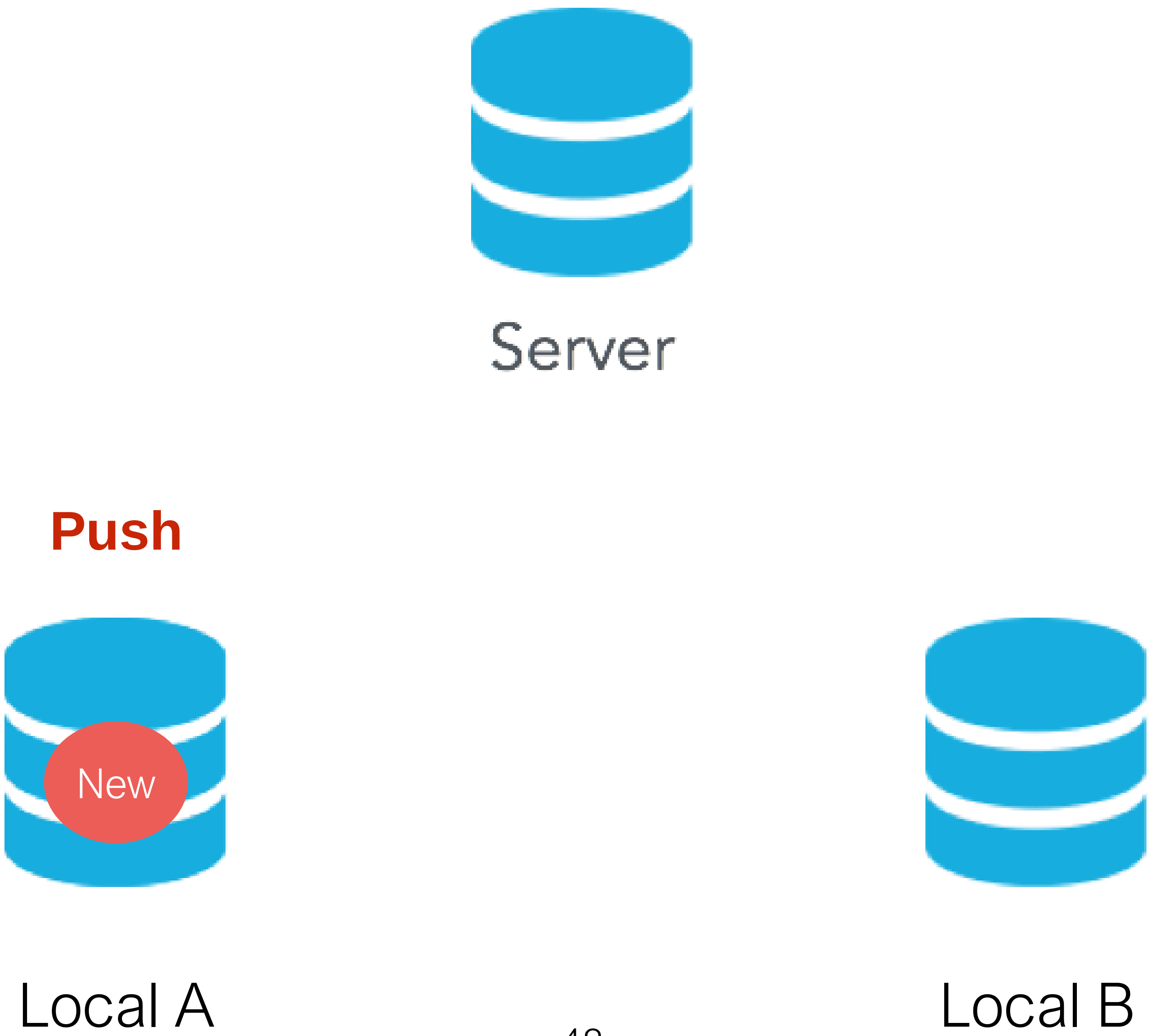
Commit

New

Local A

Local B

Collaboration workflow



Server

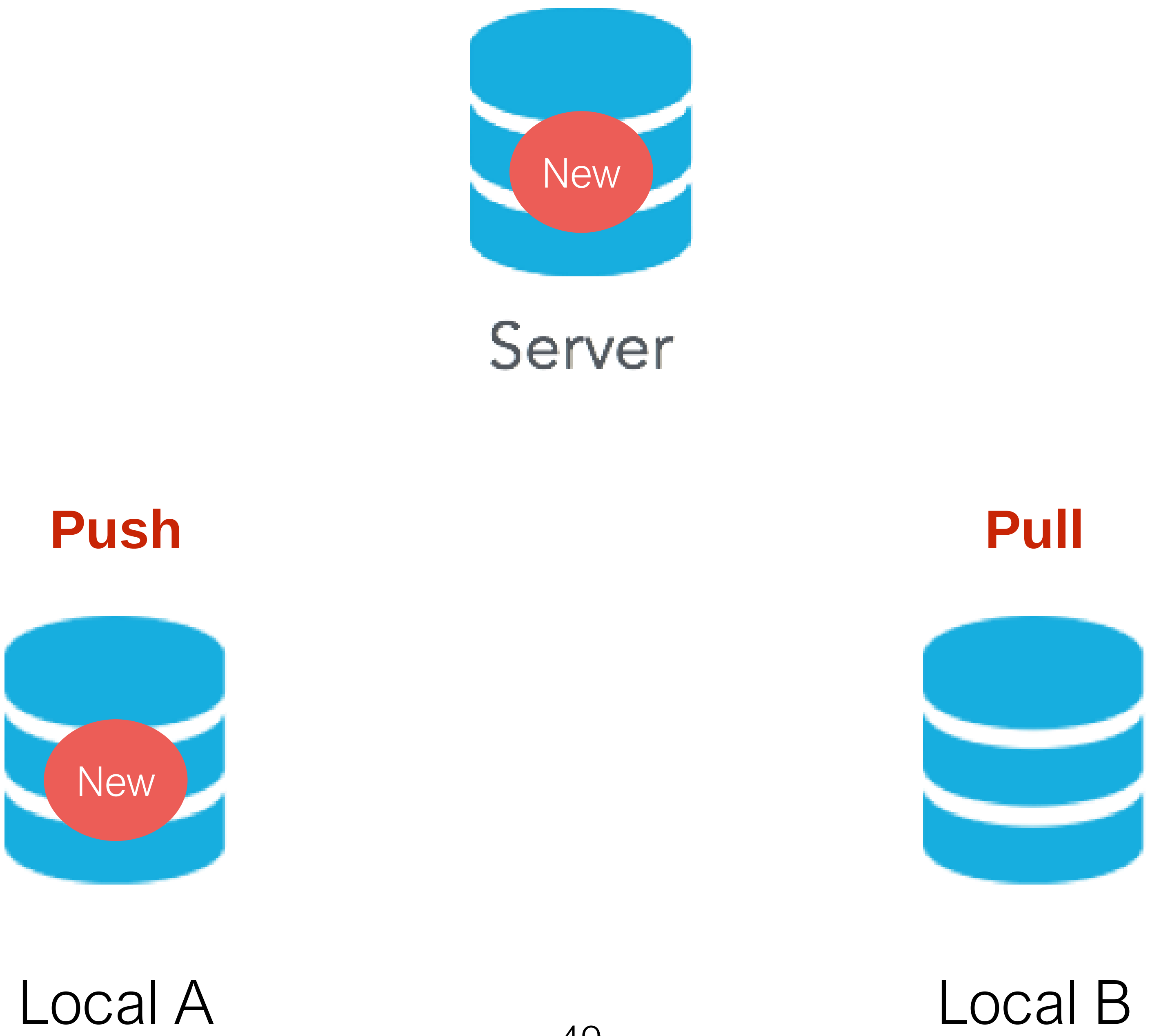
Push

New

Local A

Local B

Collaboration workflow



New

Server

Push

Pull

New

Local A

Local B

Cloning & Pushing

- Cloning the remote repositories

  ```
  git clone [Remote URL]
  ```

- The [Remote URL] is saved as **Origin**

  - After committing a few versions, you can push the branch back to **Origin**

    ```
    git push -u origin [Branch Name]
    ```

Fetch & Pull

- Updating a branch from the remote repository
  - Fetching the remote repository to local

    ```
    git fetch origin
    ```

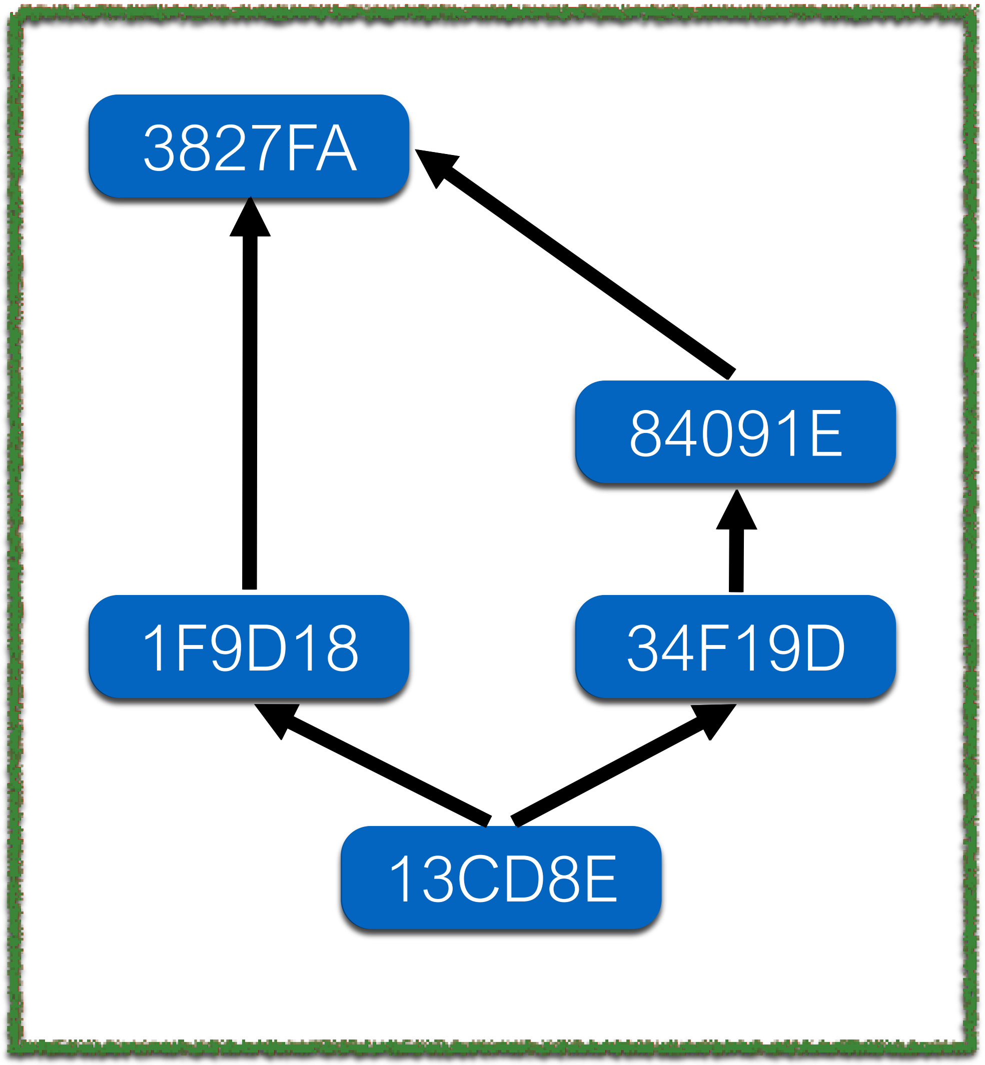
  - Merging the remote branch

    ```
    git merge origin/[Branch Name]
    ```

- Doing above commands in one command

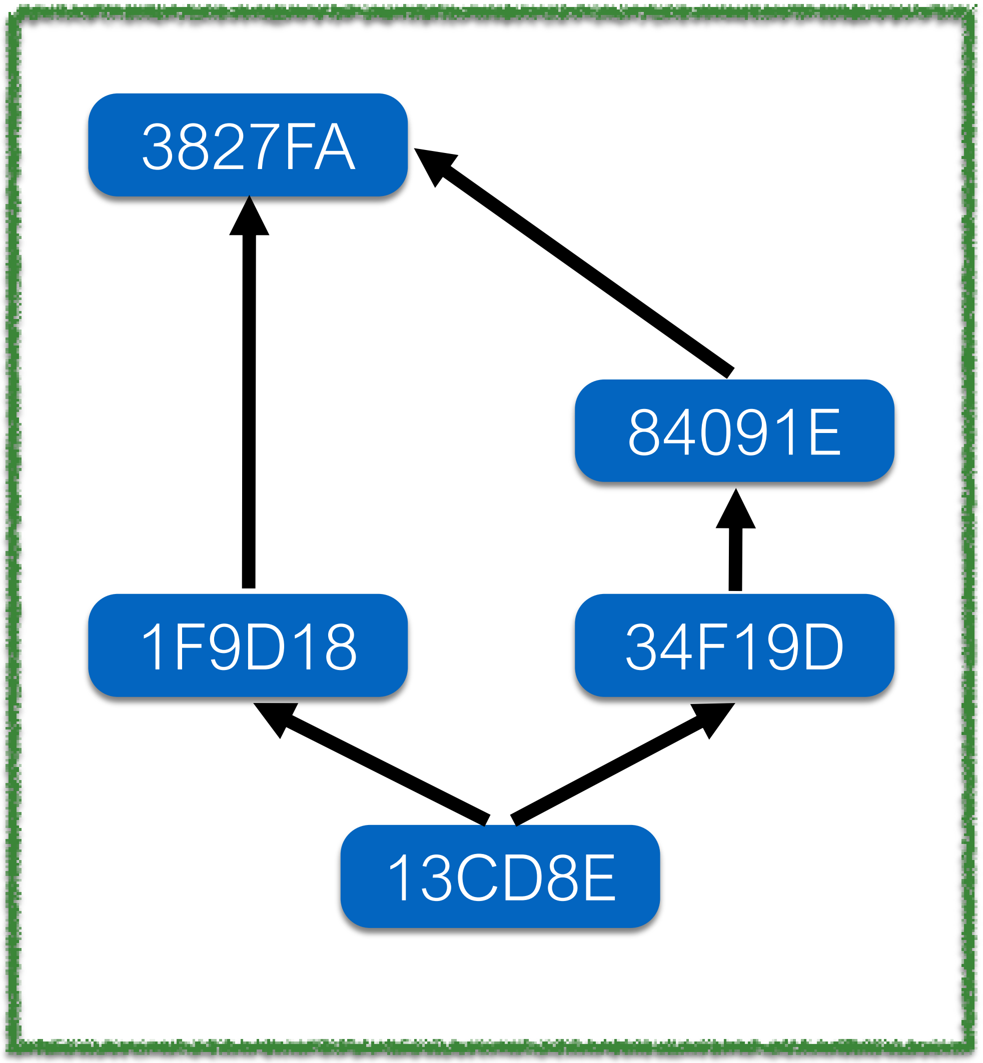
  ```
  git pull [Branch Name]
  ```

Fork

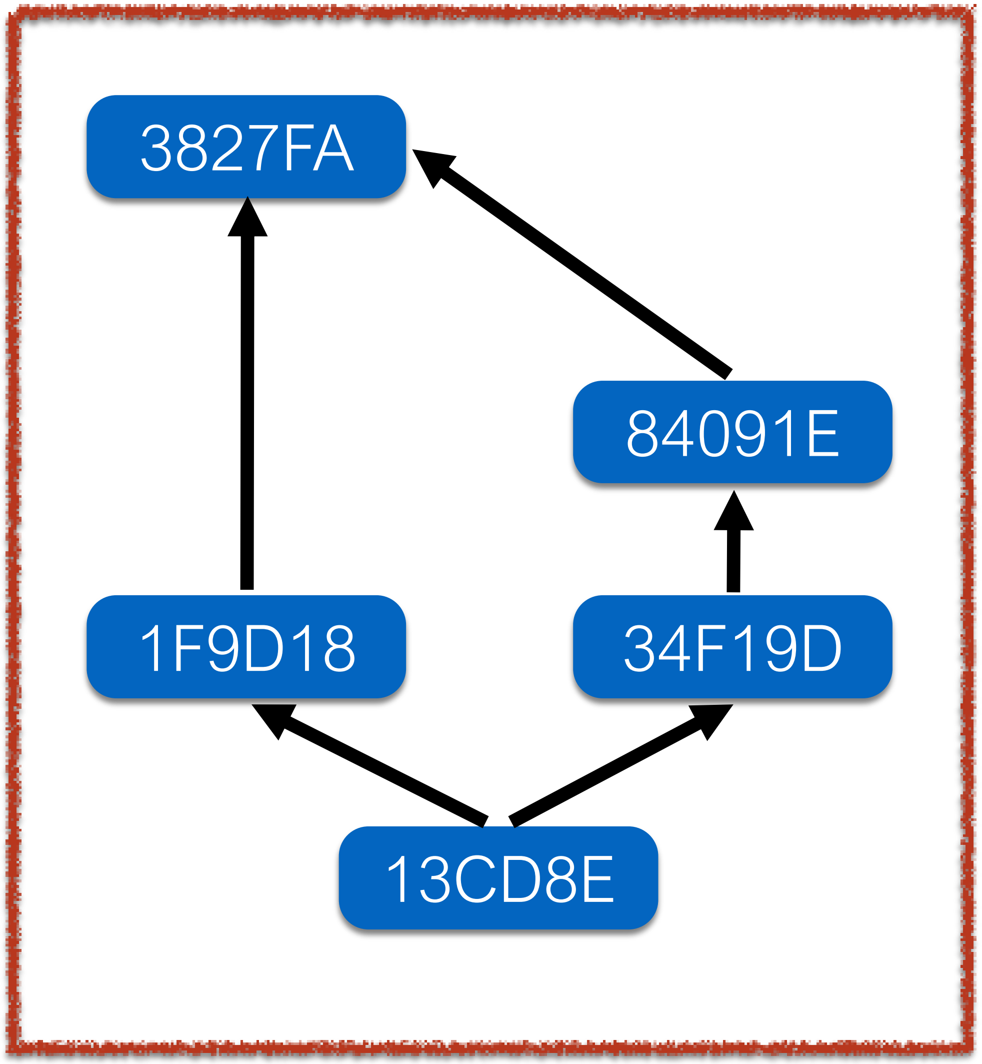


The Repo. Under TA’　s Account

Fork

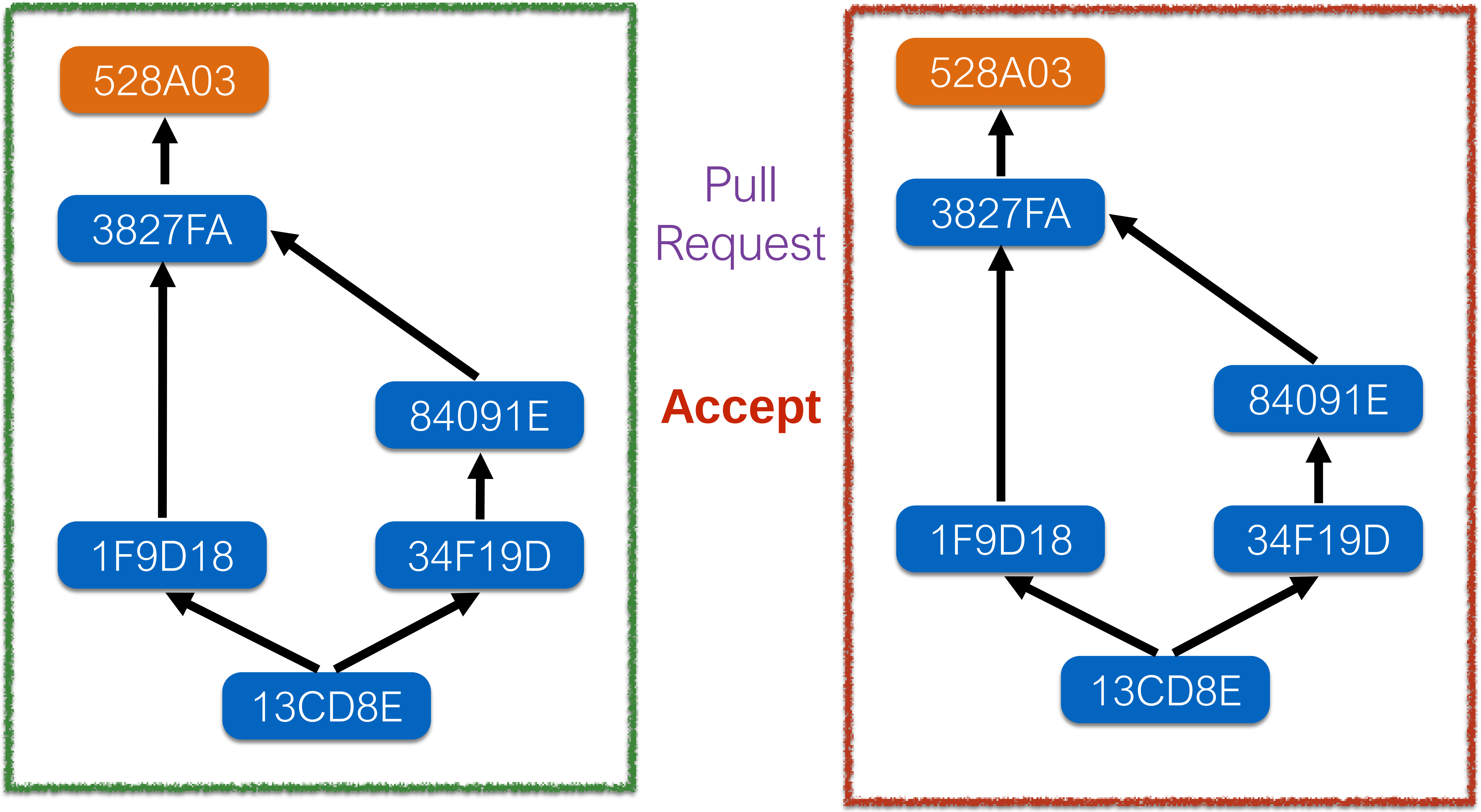


The Repo. Under TA’ s Account



The Repo. Under Your Account

Pull (Merge) Request



The Repo. Under TA’ s Account The Repo. Under Your Account

.gitignore File

- You can ignore some files that you don’ t want them to be tracked by editing the .gitignore file
- Remember to track and commit your .gitignore file
- Don’ t know what should be in .gitignore ?
    - https://github.com/github/gitignore
    - https://www.gitignore.io/

How to Submit Your Code to Gitlab

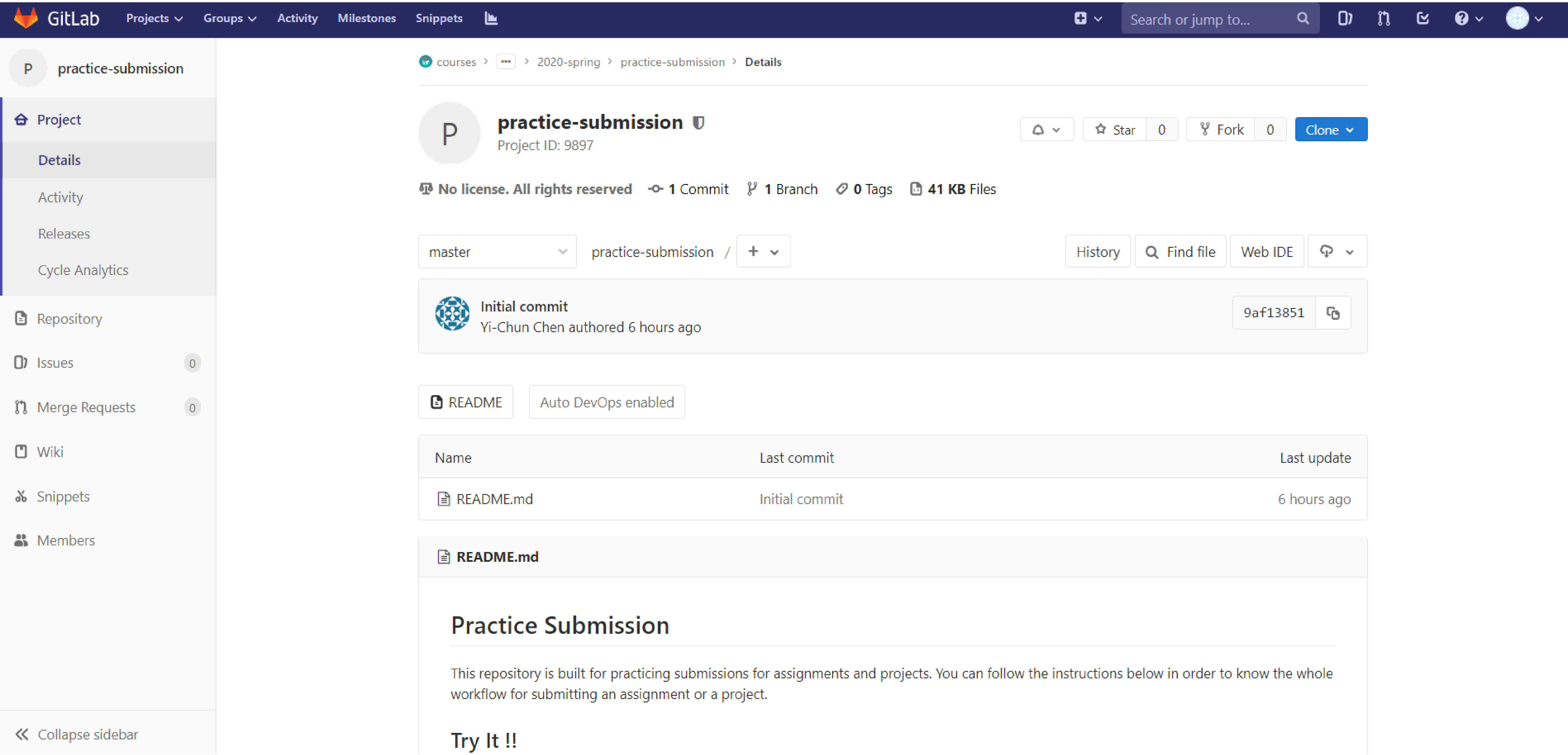
Workflow

- For each lab, you should follow the workflow below
    1. Fork our template repository on Gitlab
    2. Clone the **forked** repository to your computer
    3. Finish your lab
    4. Commit in your computer
    5. Push to Gitlab
    6. Send merge request of your branch to our template repository

Workflow

- For each lab, you should follow the workflow below

  1. Fork our template repository on Gitlab
  2. Clone the **forked** repository to your computer
  3. Finish your lab
  4. Commit in your computer
  5. Push to Gitlab
  6. Send merge request of **your branch** to our template repository



practice-submission

Project ID: 9897

No license. All rights reserved 1 Commit 1 Branch 0 Tags 41 KB Files

Initial commit
Yi-Chun Chen authored 6 hours ago
9af13851

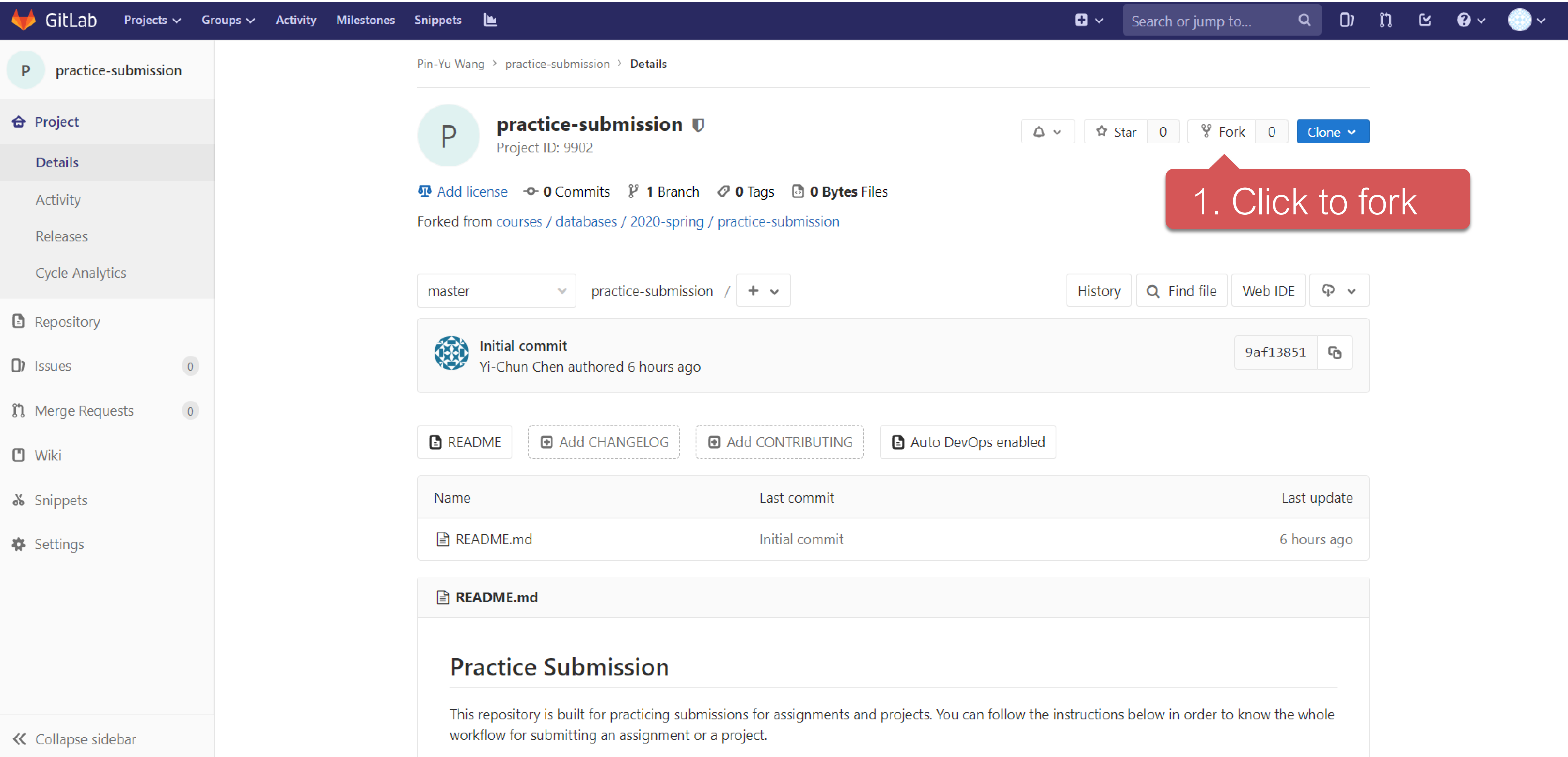
Name	Last commit	Last update
README.md	Initial commit	6 hours ago

README.md

Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting an assignment or a project.

Try It !!



1. Click to fork

Pin-Yu Wang › practice-submission › Details

practice-submission

Project ID: 9902

Add license · 0 Commits · 1 Branch · 0 Tags · 0 Bytes Files

Forked from courses / databases / 2020-spring / practice-submission

practice-submission /

Initial commit

Yi-Chun Chen authored 6 hours ago

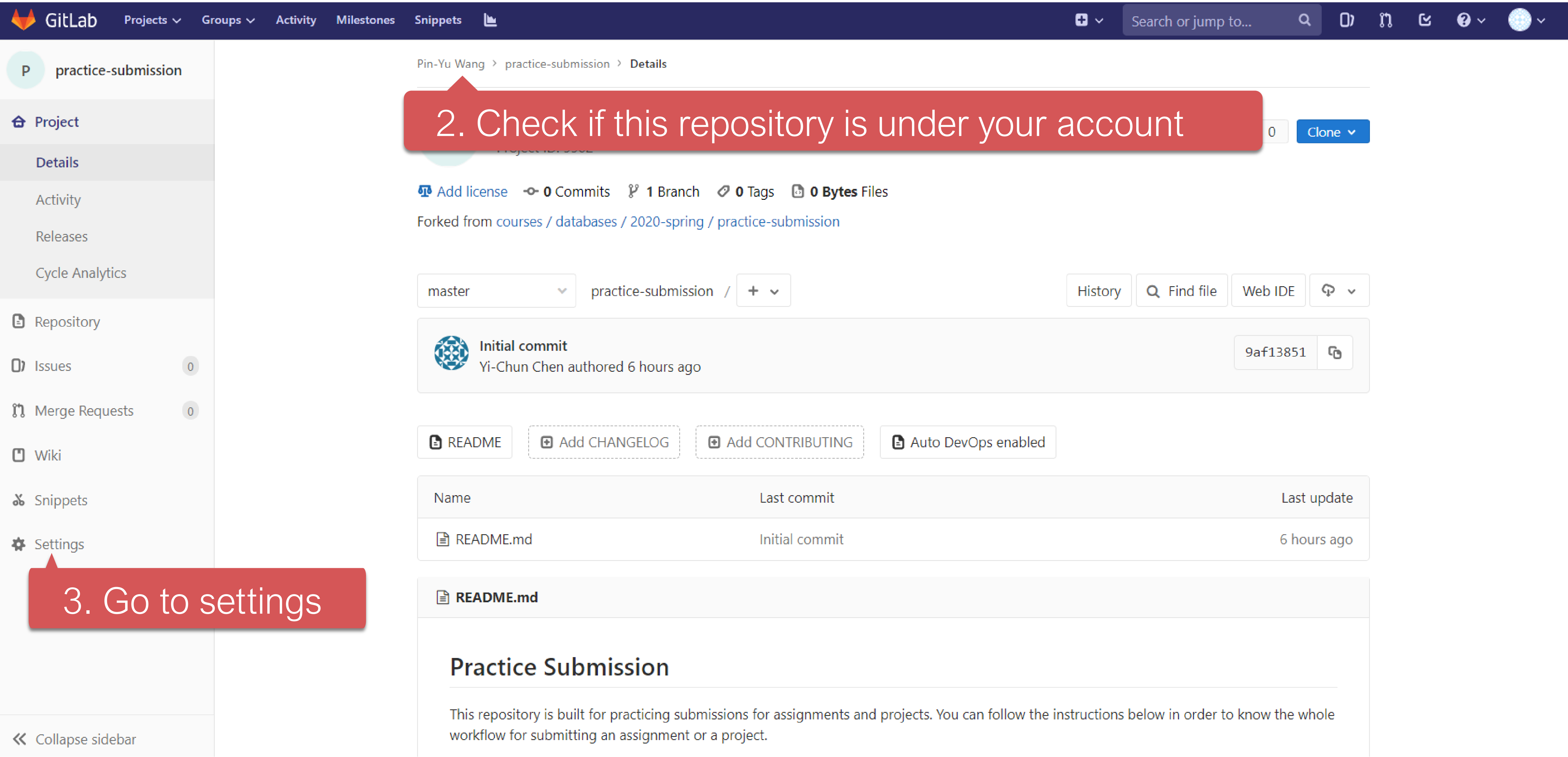
9af13851

Name	Last commit	Last update
README.md	Initial commit	6 hours ago

README.md

Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting an assignment or a project.



GitLab Projects Groups Activity Milestones Snippets Search or jump to...

practice-submission

- Project
  - Details
  - Activity
  - Releases
  - Cycle Analytics
- Repository
- Issues 0
- Merge Requests 0
- Wiki
- Snippets
- Settings
- Collapse sidebar

Pin-Yu Wang › practice-submission › Details

2. Check if this repository is under your account

0 Clone

Add license 0 Commits 1 Branch 0 Tags 0 Bytes Files

Forked from courses / databases / 2020-spring / practice-submission

master practice-submission / + History Find file Web IDE

Initial commit
Yi-Chun Chen authored 6 hours ago
9af13851

README Add CHANGELOG Add CONTRIBUTING Auto DevOps enabled

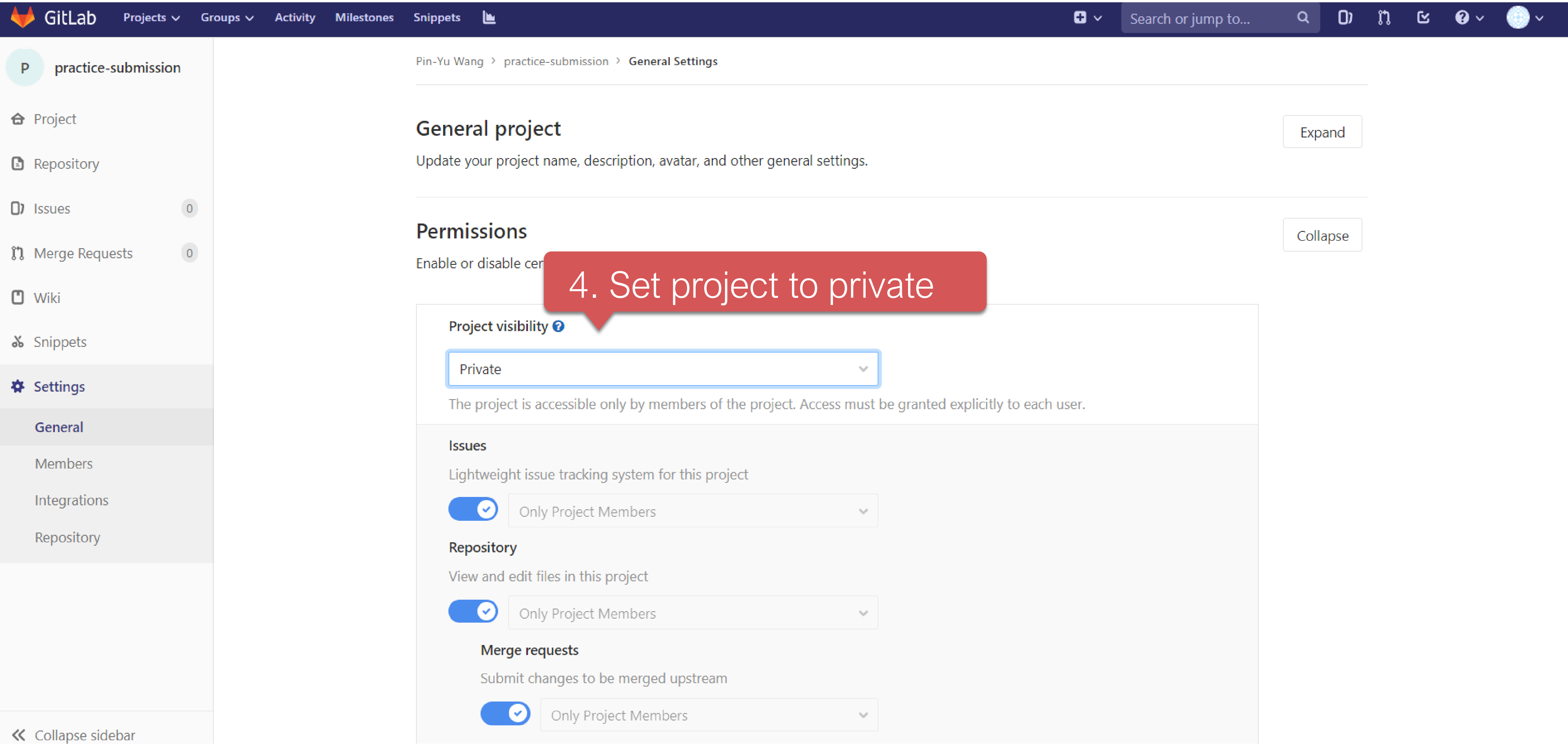
Name	Last commit	Last update
README.md	Initial commit	6 hours ago

README.md

Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting an assignment or a project.

3. Go to settings



4. Set project to private

General project

Update your project name, description, avatar, and other general settings.

Permissions

Enable or disable cer

Project visibility: Private

The project is accessible only by members of the project. Access must be granted explicitly to each user.

Issues

Lightweight issue tracking system for this project

Only Project Members

Repository

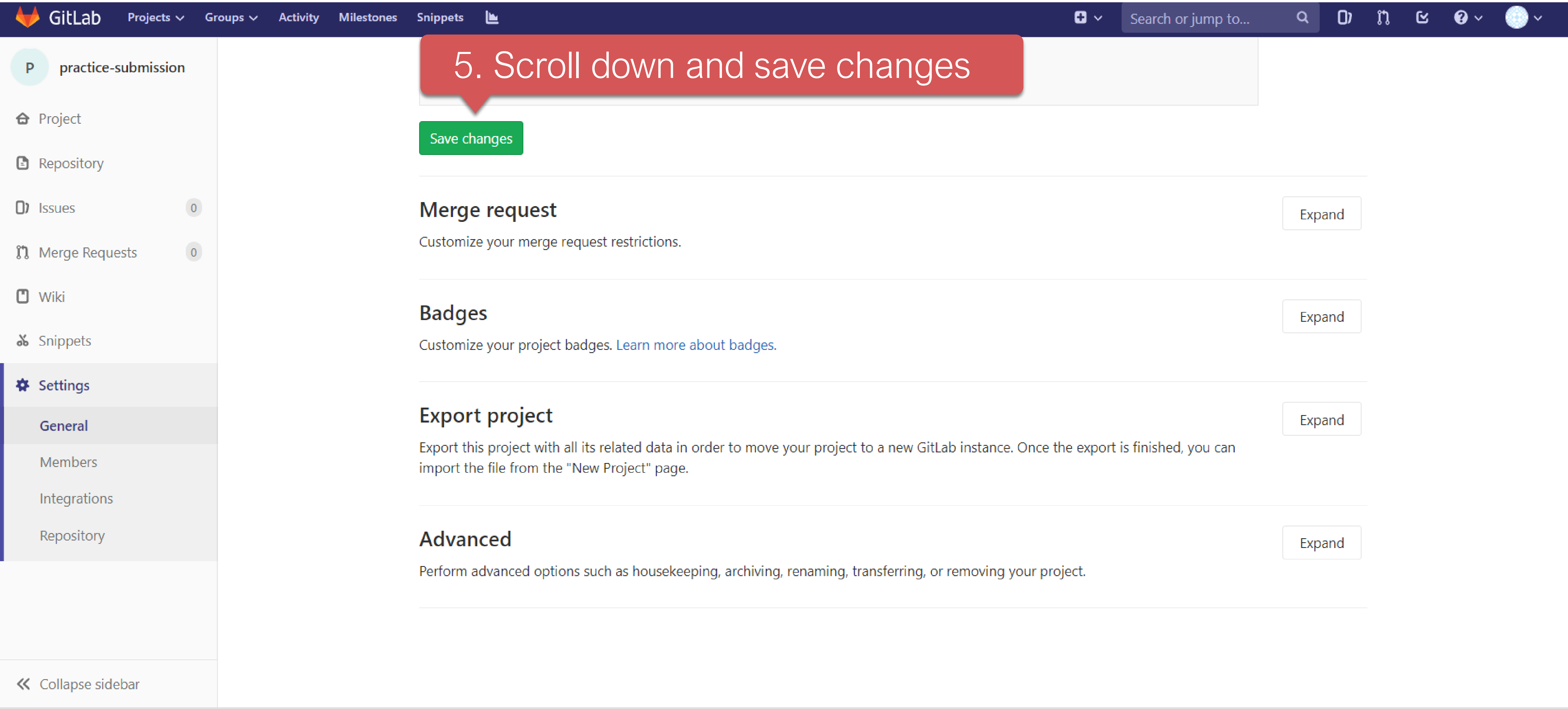
View and edit files in this project

Only Project Members

Merge requests

Submit changes to be merged upstream

Only Project Members



5. Scroll down and save changes

Merge request

Customize your merge request restrictions.

Badges

Customize your project badges. Learn more about badges.

Export project

Export this project with all its related data in order to move your project to a new GitLab instance. Once the export is finished, you can import the file from the "New Project" page.

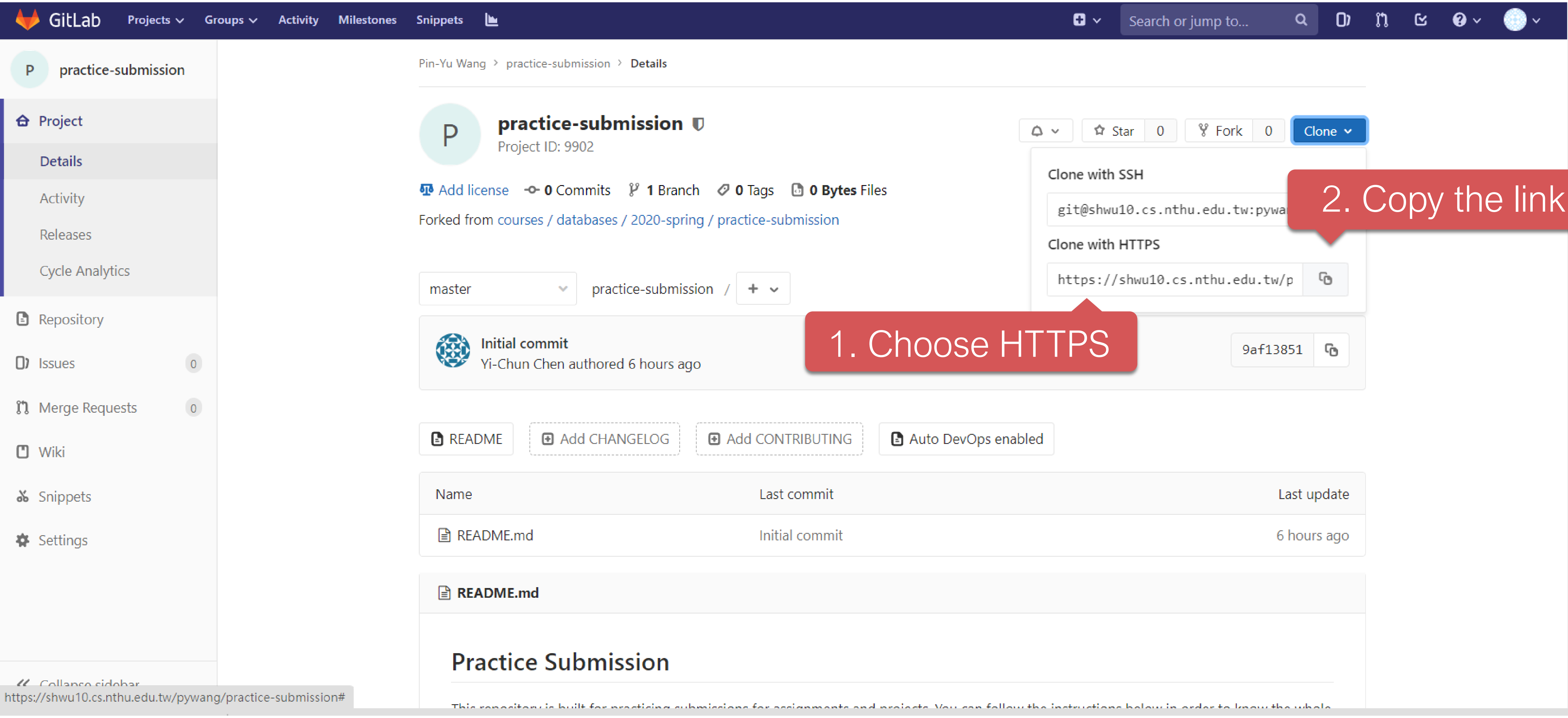
Advanced

Perform advanced options such as housekeeping, archiving, renaming, transferring, or removing your project.

Workflow

- For each lab, you should follow the workflow below

1. Fork our template repository on Gitlab
2. Clone the **forked** repository to your computer
3. Finish your lab
4. Commit in your computer
5. Push to Gitlab
6. Send merge request of **your branch** to our template repository



Pin-Yu Wang › practice-submission › Details

practice-submission

Project ID: 9902

Add license · 0 Commits · 1 Branch · 0 Tags · 0 Bytes Files

Forked from courses / databases / 2020-spring / practice-submission

Clone with SSH

`git@shwu10.cs.nthu.edu.tw:pywa`

Clone with HTTPS

`https://shwu10.cs.nthu.edu.tw/p`

2. Copy the link

1. Choose HTTPS

Initial commit

Yi-Chun Chen authored 6 hours ago

9af13851

Name	Last commit	Last update
README.md	Initial commit	6 hours ago

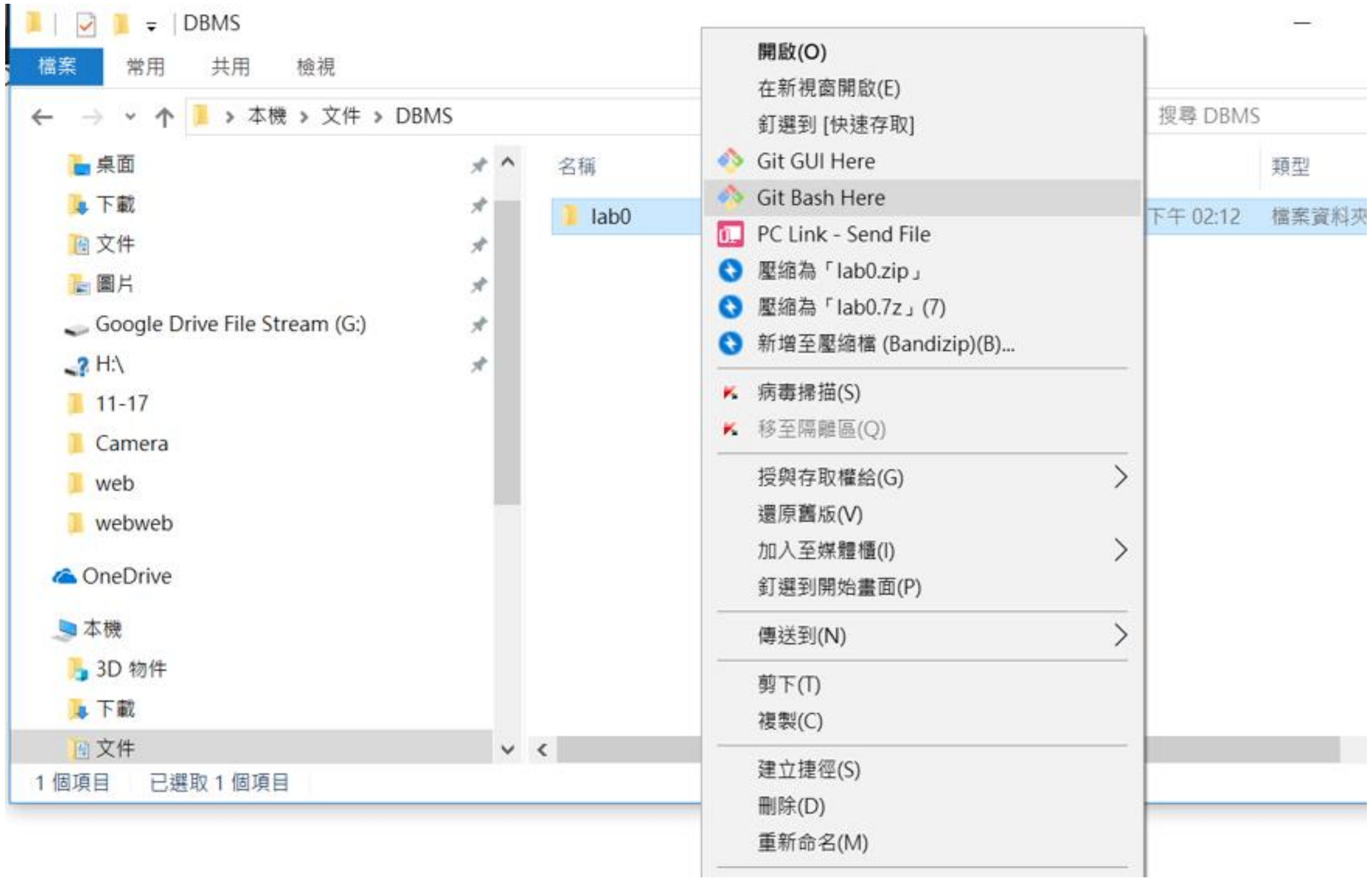
README.md

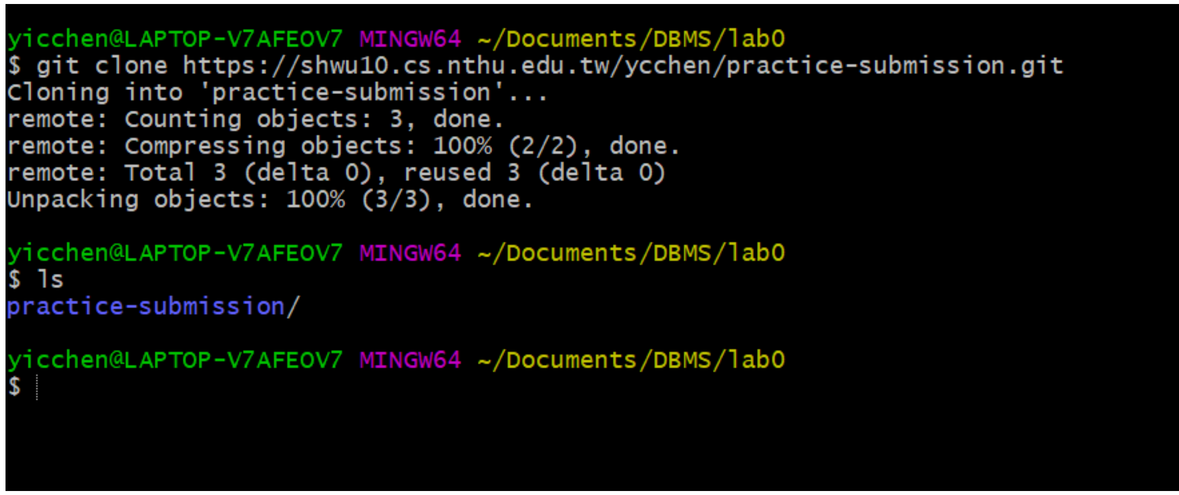
Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole

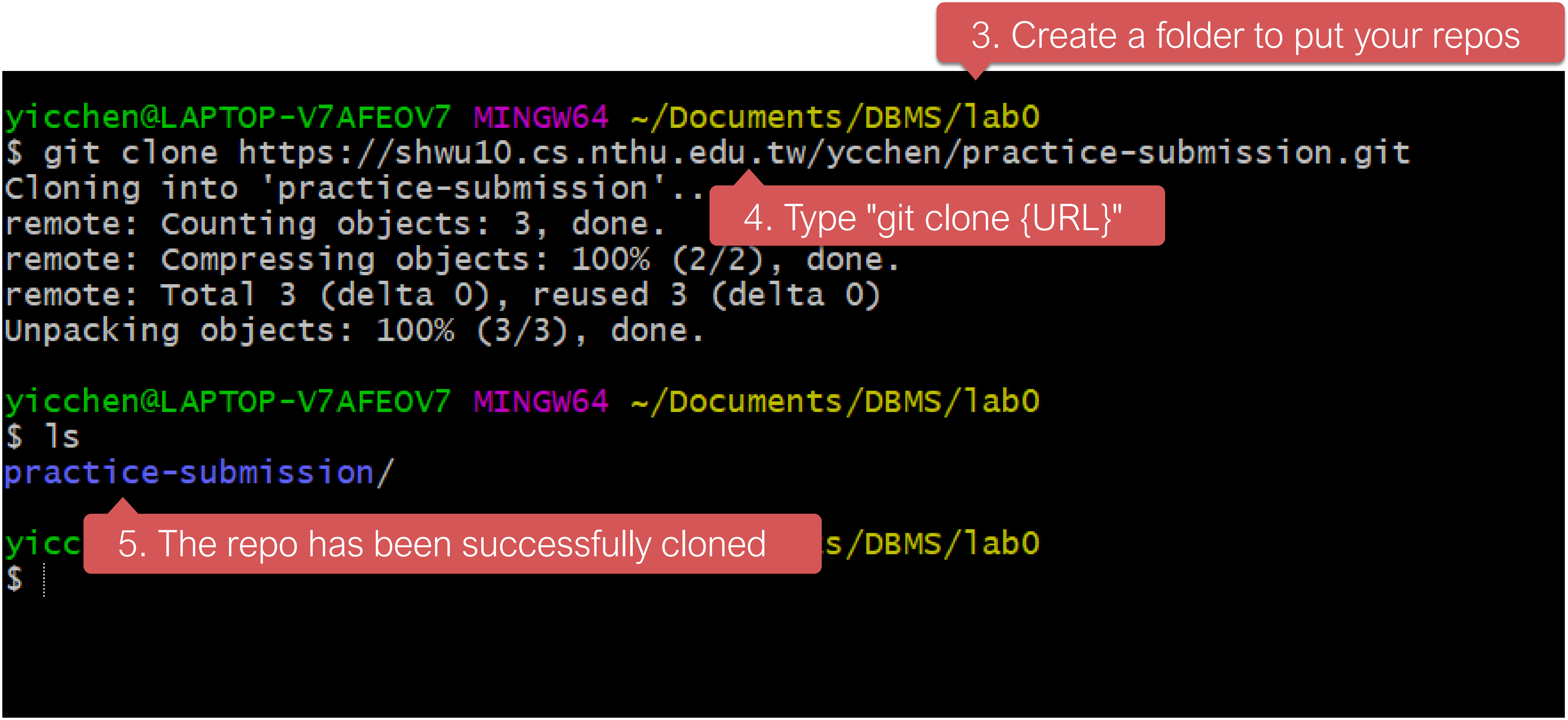
https://shwu10.cs.nthu.edu.tw/pywang/practice-submission#

If You use Windows





```
yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0
$ git clone https://shwu10.cs.nthu.edu.tw/ycchen/practice-submission.git
Cloning into 'practice-submission'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0)
Unpacking objects: 100% (3/3), done.

yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0
$ ls
practice-submission/

yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0
$
```



3. Create a folder to put your repos

```
yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0
$ git clone https://shwu10.cs.nthu.edu.tw/ycchen/practice-submission.git
Cloning into 'practice-submission'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0)
Unpacking objects: 100% (3/3), done.

yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0
$ ls
practice-submission/

yicc                                         s/DBMS/lab0
$
```

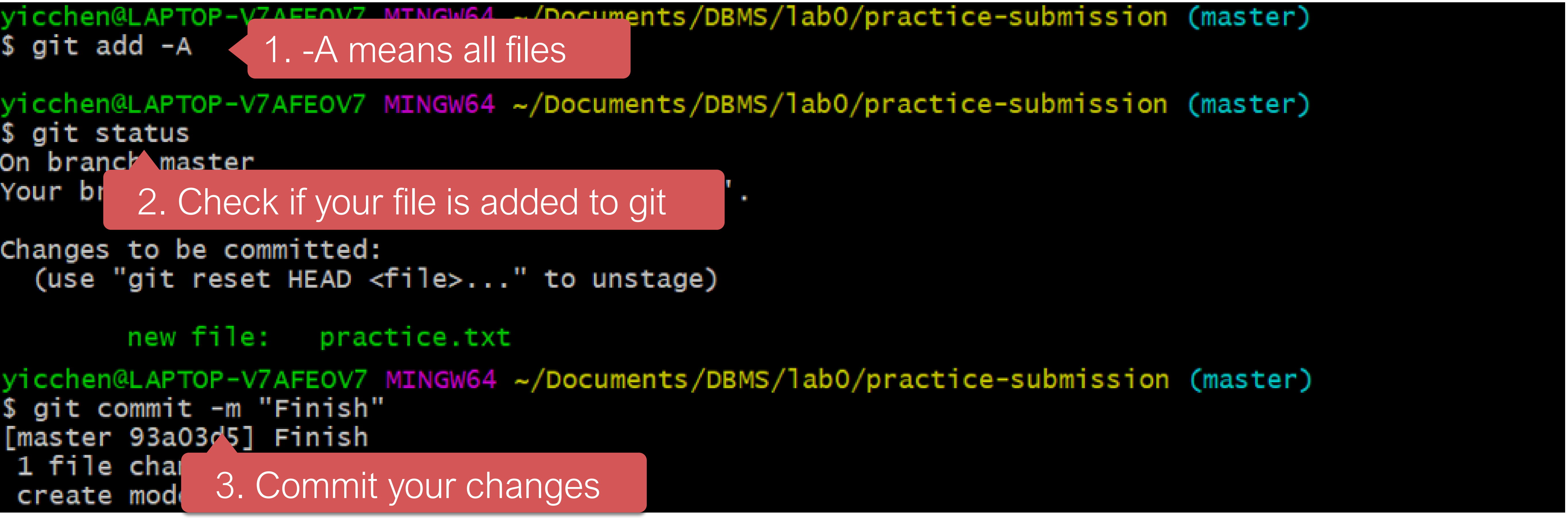
4. Type "git clone {URL}"

5. The repo has been successfully cloned

Workflow

- For each lab, you should follow the workflow below

1. Fork our template repository on Gitlab
2. Clone the **forked** repository to your computer
3. Finish your lab
4. Commit in your computer
5. Push to Gitlab
6. Send merge request of **your branch** to our template repository

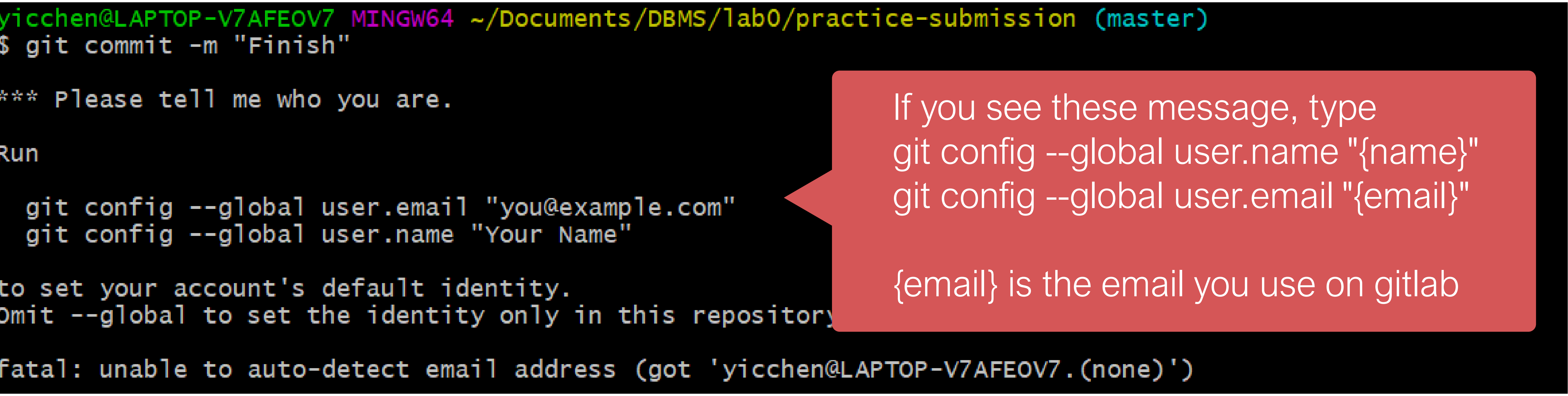


```
yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git add -A

yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git status
On branch master
Your br                                            '.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   practice.txt

yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git commit -m "Finish"
[master 93a03d5] Finish
 1 file char
 create mod
```

1. -A means all files

2. Check if your file is added to git

3. Commit your changes



```
yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git commit -m "Finish"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository

fatal: unable to auto-detect email address (got 'yicchen@LAPTOP-V7AFEOV7.(none)')
```

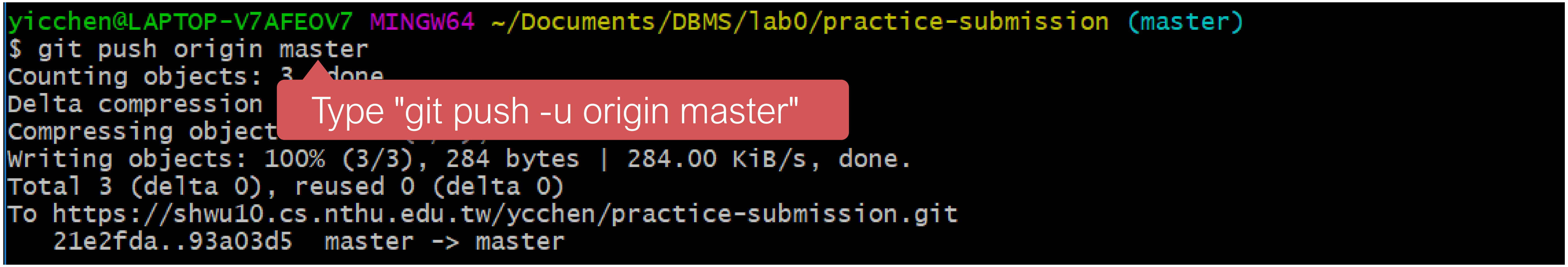
If you see these message, type
git config --global user.name "{name}"
git config --global user.email "{email}"

{email} is the email you use on gitlab

Workflow

- For each lab, you should follow the workflow below

1. Fork our template repository on Gitlab
2. Clone the **forked** repository to your computer
3. Finish your lab
4. Commit in your computer
5. Push to Gitlab
6. Send merge request of **your branch** to our template repository



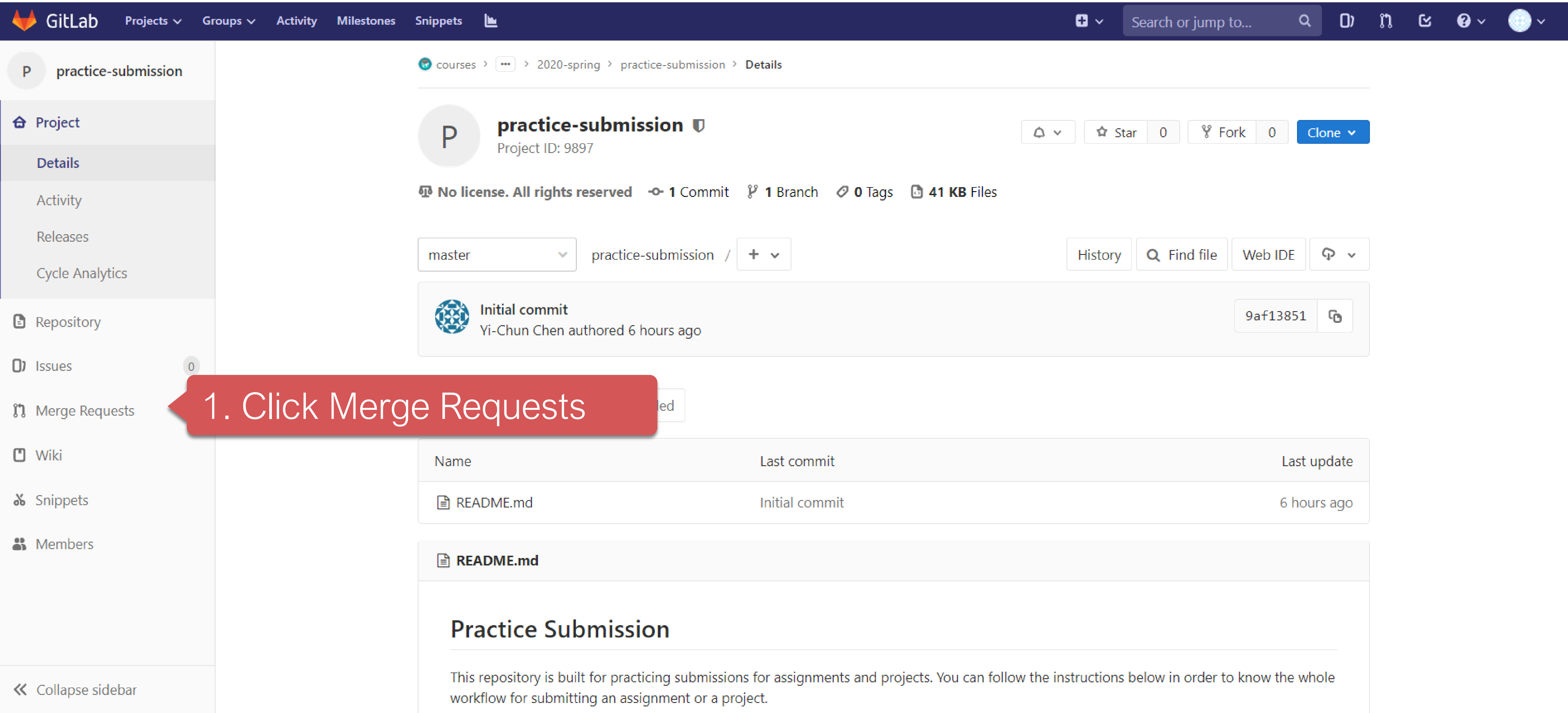
```
yicchen@LAPTOP-V7AFEOV7 MINGW64 ~/Documents/DBMS/lab0/practice-submission (master)
$ git push origin master
Counting objects: 3  done
Delta compression
Compressing object
Writing objects: 100% (3/3), 284 bytes | 284.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://shwu10.cs.nthu.edu.tw/ycchen/practice-submission.git
   21e2fda..93a03d5  master -> master
```

Type "git push -u origin master"

Workflow

- For each lab, you should follow the workflow below

1. Fork our template repository on Gitlab
2. Clone the **forked** repository to your computer
3. Finish your lab
4. Commit in your computer
5. Push to Gitlab
6. Send merge request of your branch to our template repository



practice-submission

Project ID: 9897

No license. All rights reserved 1 Commit 1 Branch 0 Tags 41 KB Files

Initial commit

Yi-Chun Chen authored 6 hours ago

9af13851

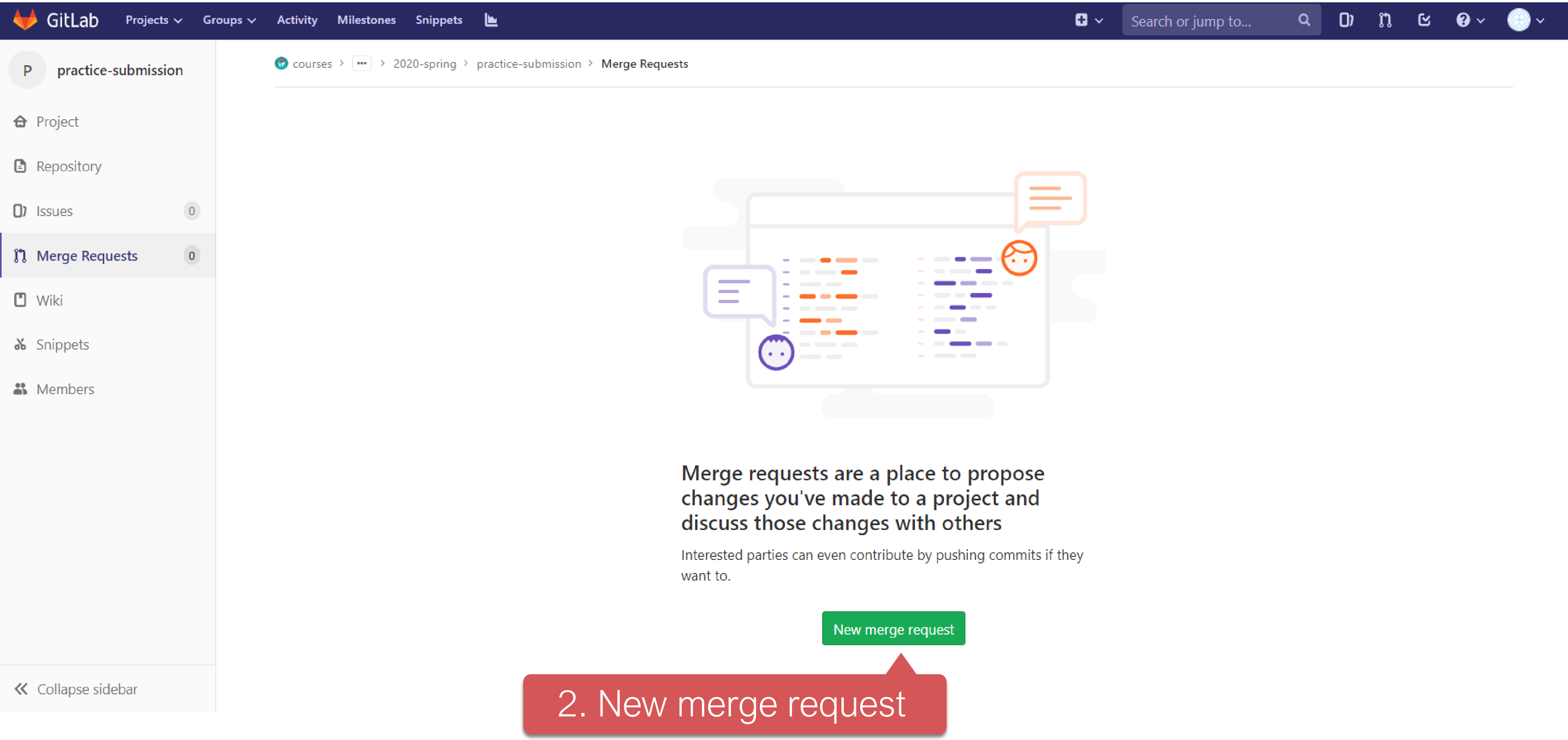
Name	Last commit	Last update
README.md	Initial commit	6 hours ago

README.md

Practice Submission

This repository is built for practicing submissions for assignments and projects. You can follow the instructions below in order to know the whole workflow for submitting an assignment or a project.

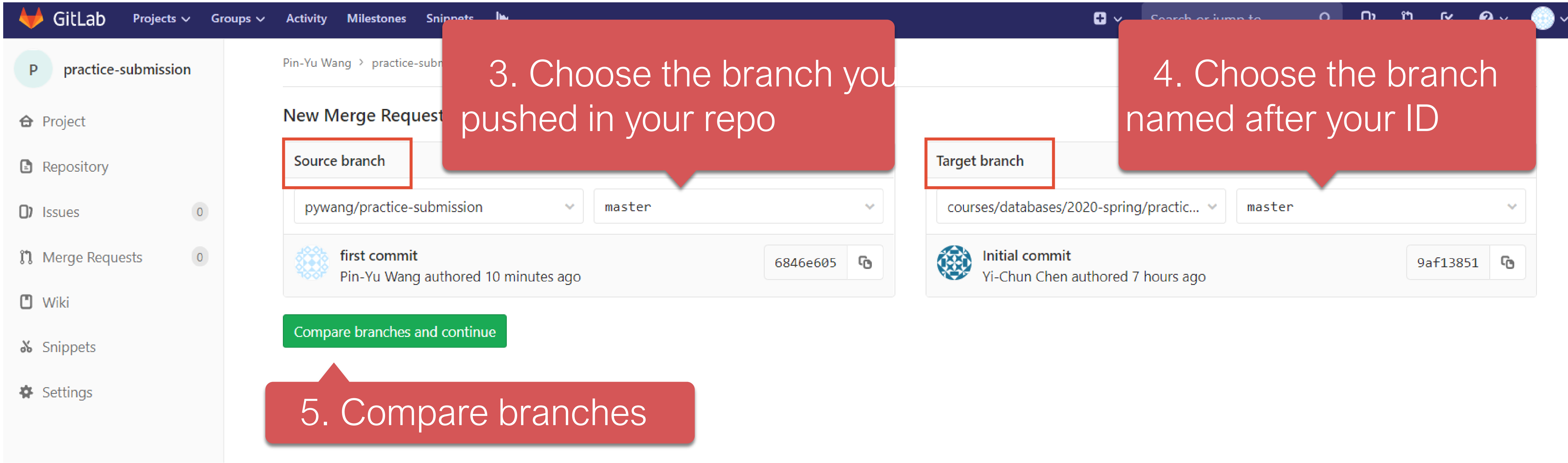
1. Click Merge Requests



Merge requests are a place to propose changes you've made to a project and discuss those changes with others

Interested parties can even contribute by pushing commits if they want to.

2. New merge request



New Merge Request

Source branch

pywang/practice-submission | master

first commit
Pin-Yu Wang authored 10 minutes ago
6846e605

Target branch

courses/databases/2020-spring/practic... | master

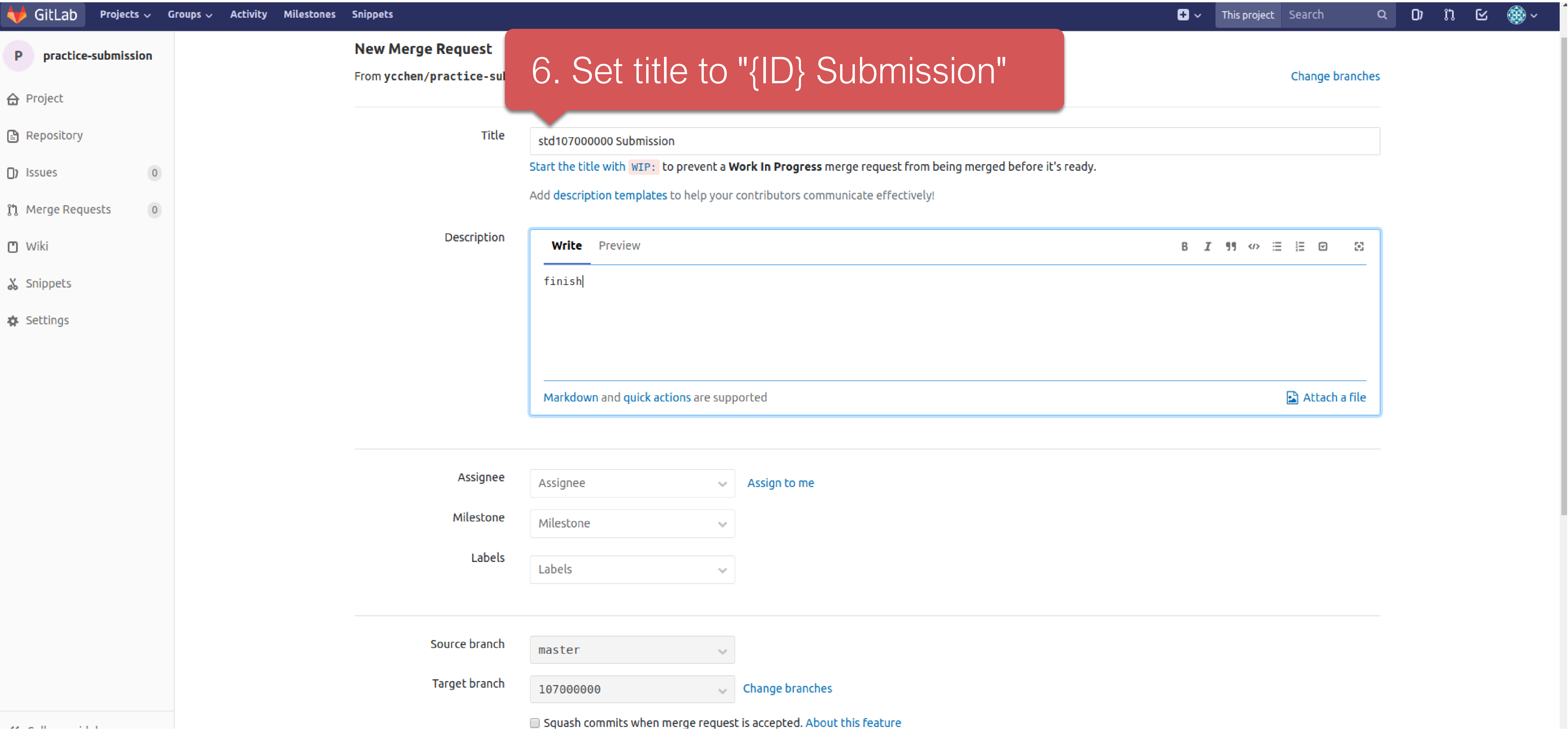
Initial commit
Yi-Chun Chen authored 7 hours ago
9af13851

Compare branches and continue

3. Choose the branch you pushed in your repo

4. Choose the branch named after your ID

5. Compare branches



6. Set title to "{ID} Submission"

New Merge Request

From ycchen/practice-sub

Change branches

Title: std107000000 Submission

Start the title with `WIP:` to prevent a **Work In Progress** merge request from being merged before it's ready.

Add description templates to help your contributors communicate effectively!

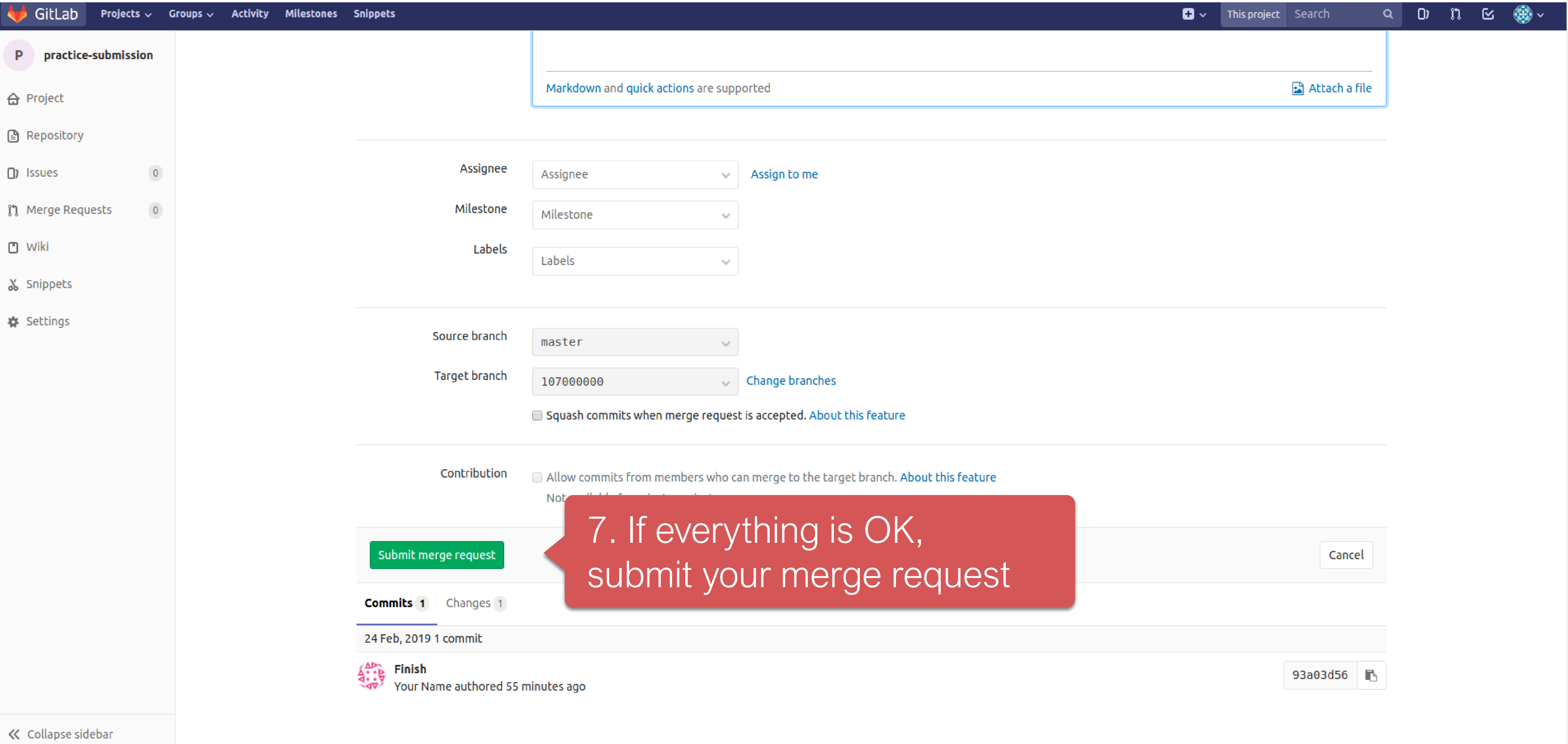
Description:

```
finish
```

Markdown and quick actions are supported

Assignee	Assignee
Milestone	Milestone
Labels	Labels
Source branch	master
Target branch	107000000

□ Squash commits when merge request is accepted. About this feature



practice-submission

Markdown and quick actions are supported

Assignee: Assignee

Milestone: Milestone

Labels: Labels

Source branch: `master`

Target branch: `107000000`

□ Squash commits when merge request is accepted. About this feature

Contribution: □ Allow commits from members who can merge to the target branch. About this feature

Not

7. If everything is OK, submit your merge request

24 Feb, 2019 1 commit

Finish
Your Name authored 55 minutes ago
`93a03d56`

In case of fire

1. git commit
2. git push
3. leave building

Reference

- Learn Git branching (interactive)
  - http://pcottle.github.io/learnGitBranching/
- Pro Git
  - http://git-scm.com/book/
- 寫給大家的 Git 教學
  - http://www.slideshare.net/littlebtc/git-5528339