

# Lab 00

## How to Survive & Introduction to Git

Software Studio  
DataLab, CS, NTHU

# Notice

- These slides will focus on how to **submit you code** by using Git command line
- You can also use other Git GUI tool or built-in Git tool in other IDE/editor

# Teaching Assistants



Yu-Hsuan Chen

陳佑軒



Jou-Hsuan Yang

楊柔暄



Sheng-Wei Cheng

鄭勝偉

# How to Find Us?

- Office Hour (TAs)
  - Tue. 10:10am-12:00pm at Delta 729
- Email
  - Yu-Hsuan Chen : [yhch@datalab.cs.nthu.edu.tw](mailto:yhch@datalab.cs.nthu.edu.tw)
  - Jou-Hsuan Yang : [jhya@datalab.cs.nthu.edu.tw](mailto:jhya@datalab.cs.nthu.edu.tw)
  - Sheng-Wei Cheng: [swcheng@datalab.cs.nthu.edu.tw](mailto:swcheng@datalab.cs.nthu.edu.tw)
- Online Forum
  - eeclass

G o o g l e

The image shows the word "Google" in a playful, hand-drawn style. The first 'G' is blue. The second 'o' is a red circle with a smiling face and a purple and white striped scarf. The third 'o' is an orange circle with a smiling face and a teal scarf with white polka dots. The first 'g' is blue. The 'l' is a solid green vertical bar. The final 'e' is red. The letters have a slightly textured, brush-painted appearance.

# If I have Question?

- Always **Google first** !
  - Learn how to google is important.
- If you try your best but still can't catch it.
  - Feel free to ask us on eeclash or office hour.



# Today's exercise (1/2)

- Install Git command line tool in your computer.
  - Follow appendix “Git Command-line Tool Installation”.
- Follow following steps.

# Outline

- General Rule
- Introduction to Git
  - Version control
  - Git Basics
  - Try Git!
  - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References



# Outline

- **General Rule**
- Introduction to Git
  - Version control
  - Git Basics
  - Try Git!
  - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

# The Policy of Labs

- **All labs need to be submitted to GitLab.**
- Late submission will **not** be accepted.
- Plagiarism will not be tolerated.
  - If we find you copy someone's code, you will get **0 point** for that lab.
- Grading
  - Submission before lab ends gets 100% score
  - Submission before **11:59pm** gets 60% score

# Grading Example

- 4 problems, 25% each
- Solved 4 during the lab
  - 100
- Solved 3 during the lab, 1 before 11:59pm
  - $75 + 25 * 0.6 = 90$
- Solved 4 after the lab, before 11:59pm
  - $100 * 0.6 = 60$

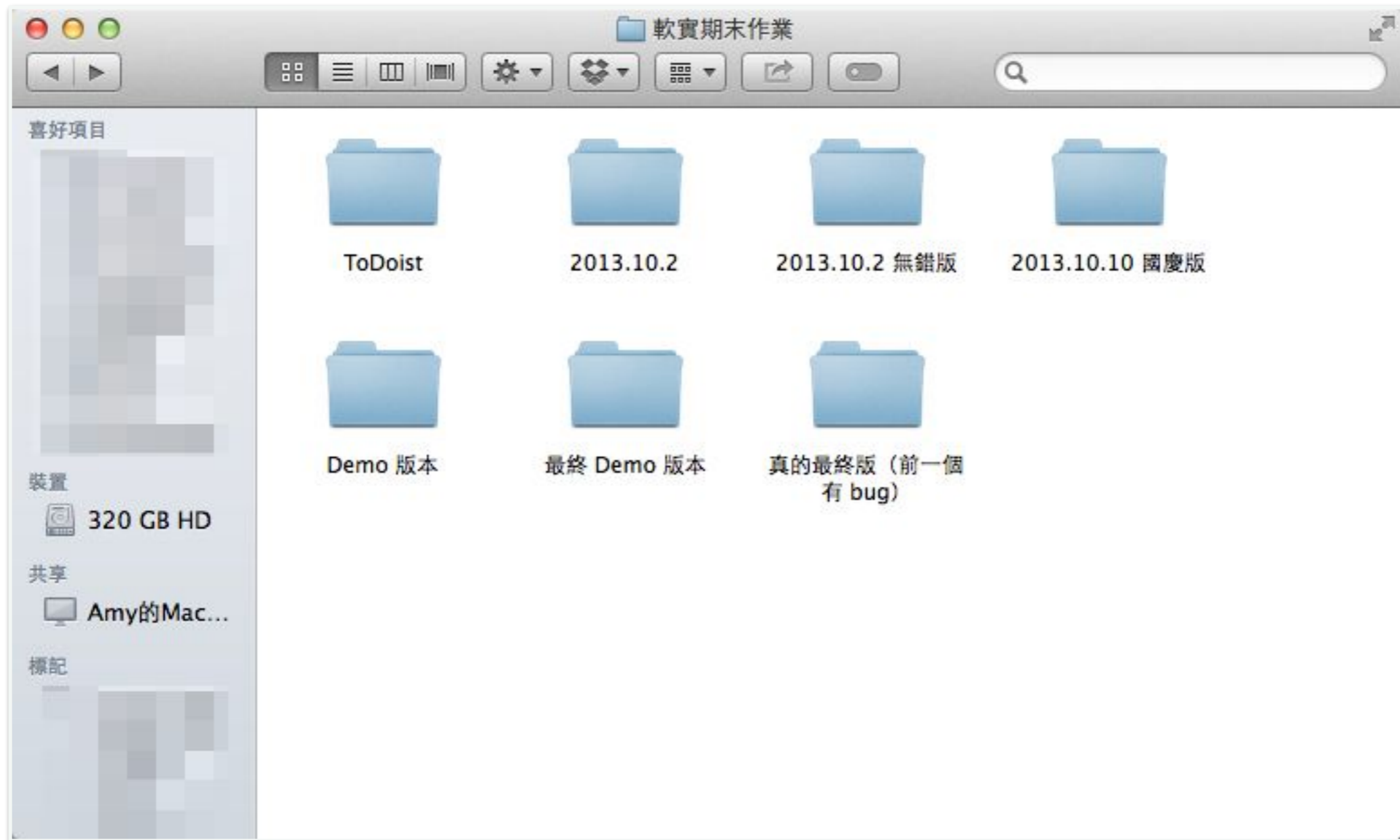
# Outline

- General Rule
- Introduction to Git
  - Version control
  - Git Basics
  - Try Git!
  - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

# Why use version control?

**We want to track what we  
did and when we did it.**

# Students' VCS



# Why use VCS?

- Managing your projects - tracking your files and modifications.
- Synchronization between modifications made by different developers.
- Revision history is still very helpful even if you work alone.



# Outline

- General Rule
- Introduction to Git
  - Version control
  - **Git Basics**
  - Try Git!
  - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

# Git



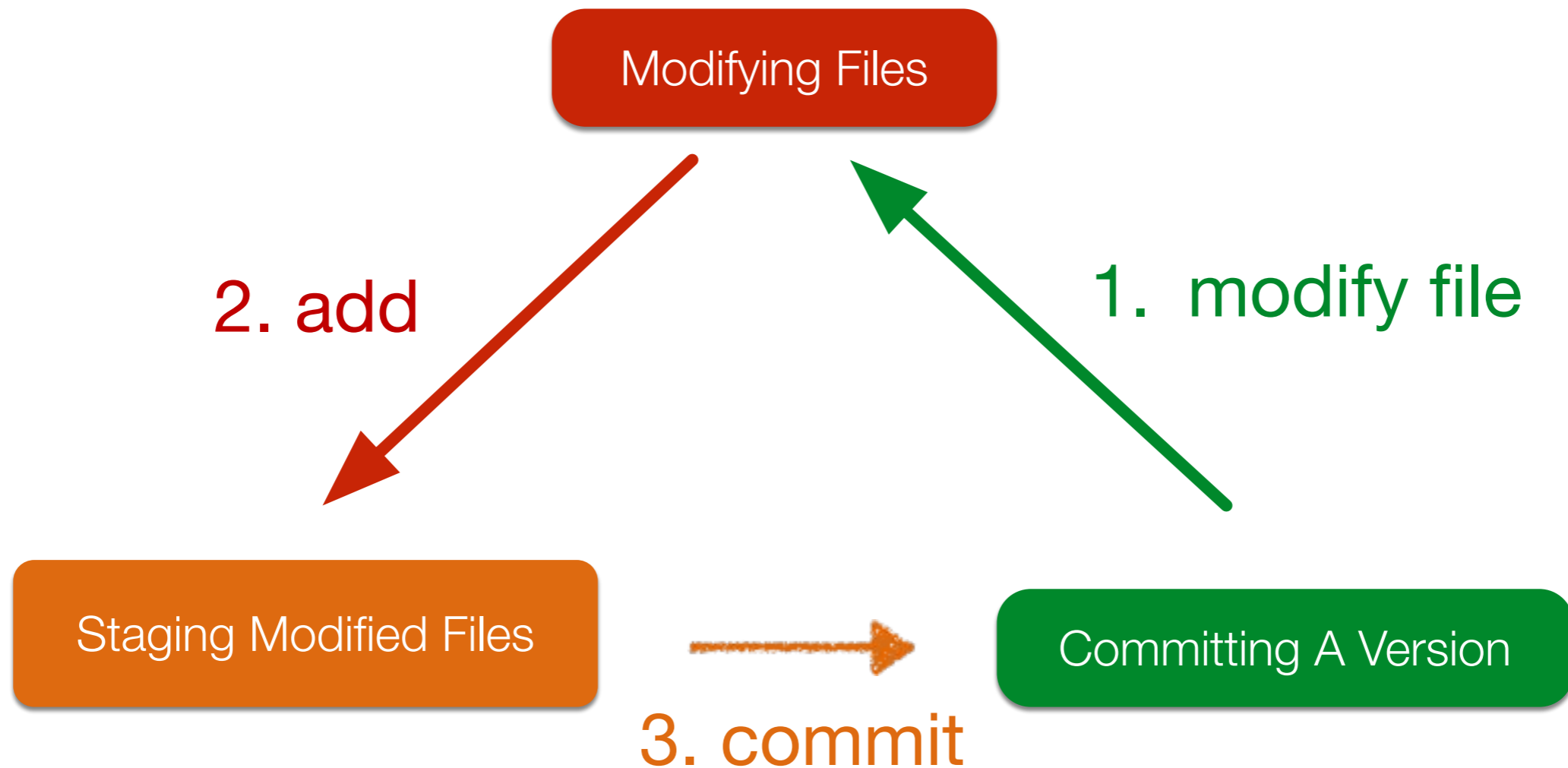
# Git

- Git is a popular version control system which is
  - Fast
  - Easy to use
  - Distributed
- A git repository is a mini database that tracks your files.

# Git Workflow (1/2)

- With a local repository in your computer, you'll need following operations to make git track your work:
  1. Create/modify files
  2. Let git monitor the files by *adding* them to staging files.
  3. *Commit* your changes to and git will create snapshots (versions) of the files for you.

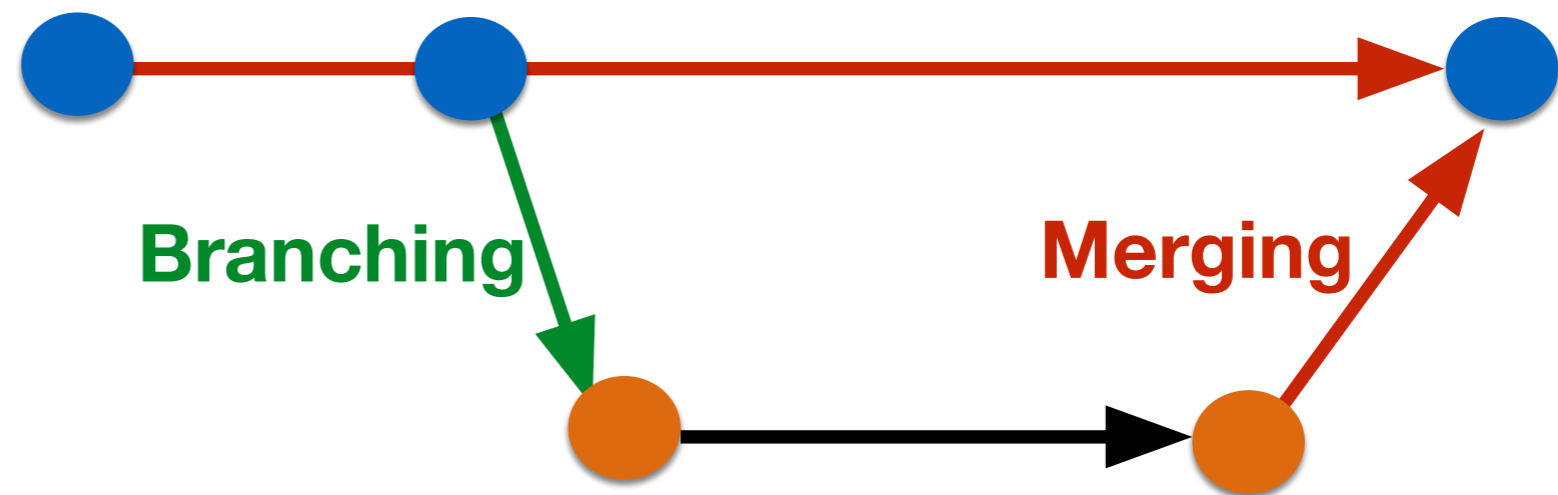
# Git Workflow (2/2)



# Git Branch

Master

Develop



# Outline

- General Rule
- Introduction to Git
  - Version control
  - Git Basics
  - **Try Git!**
  - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References

# Be Professional





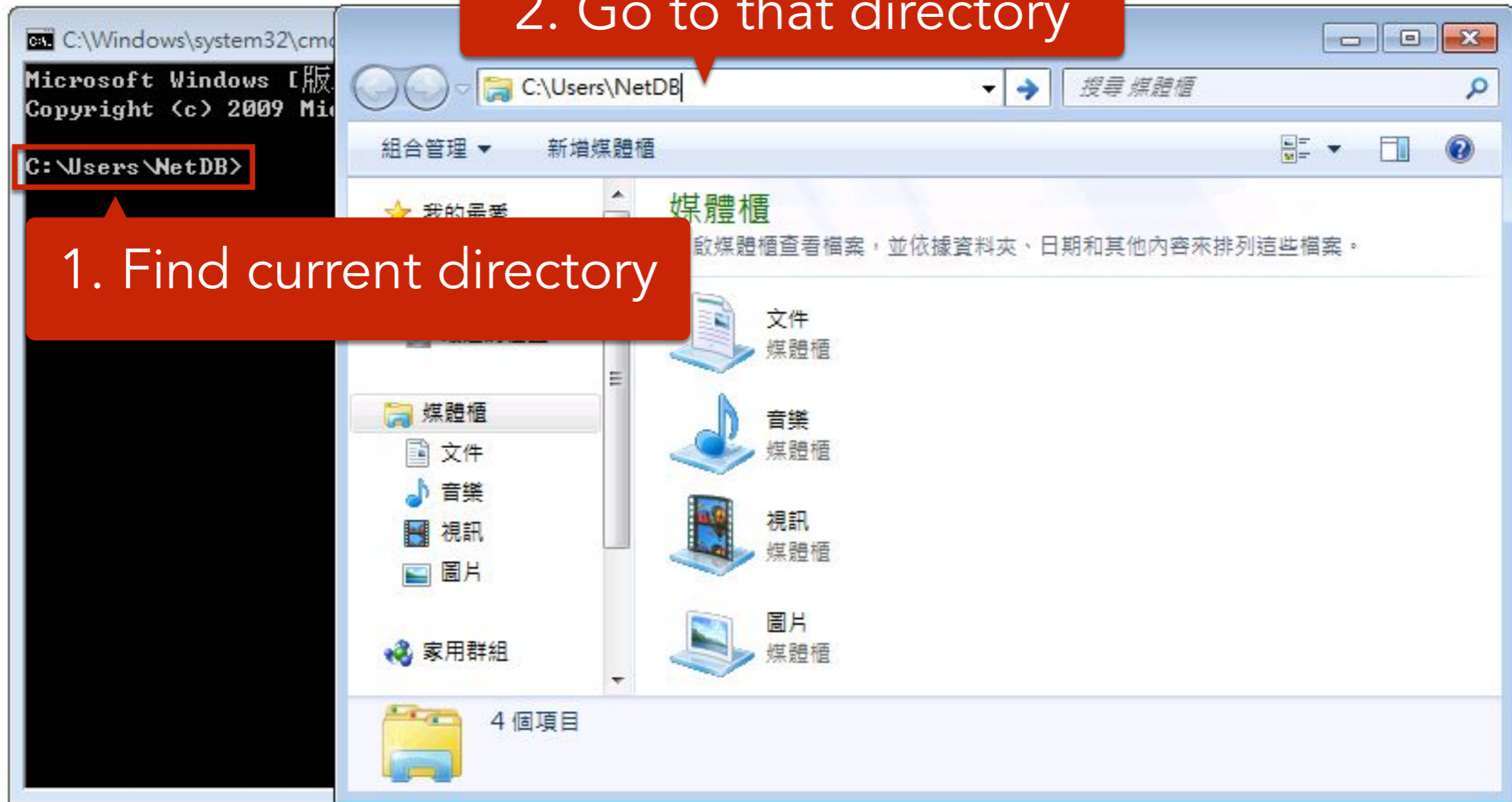
# Basic Git Commands

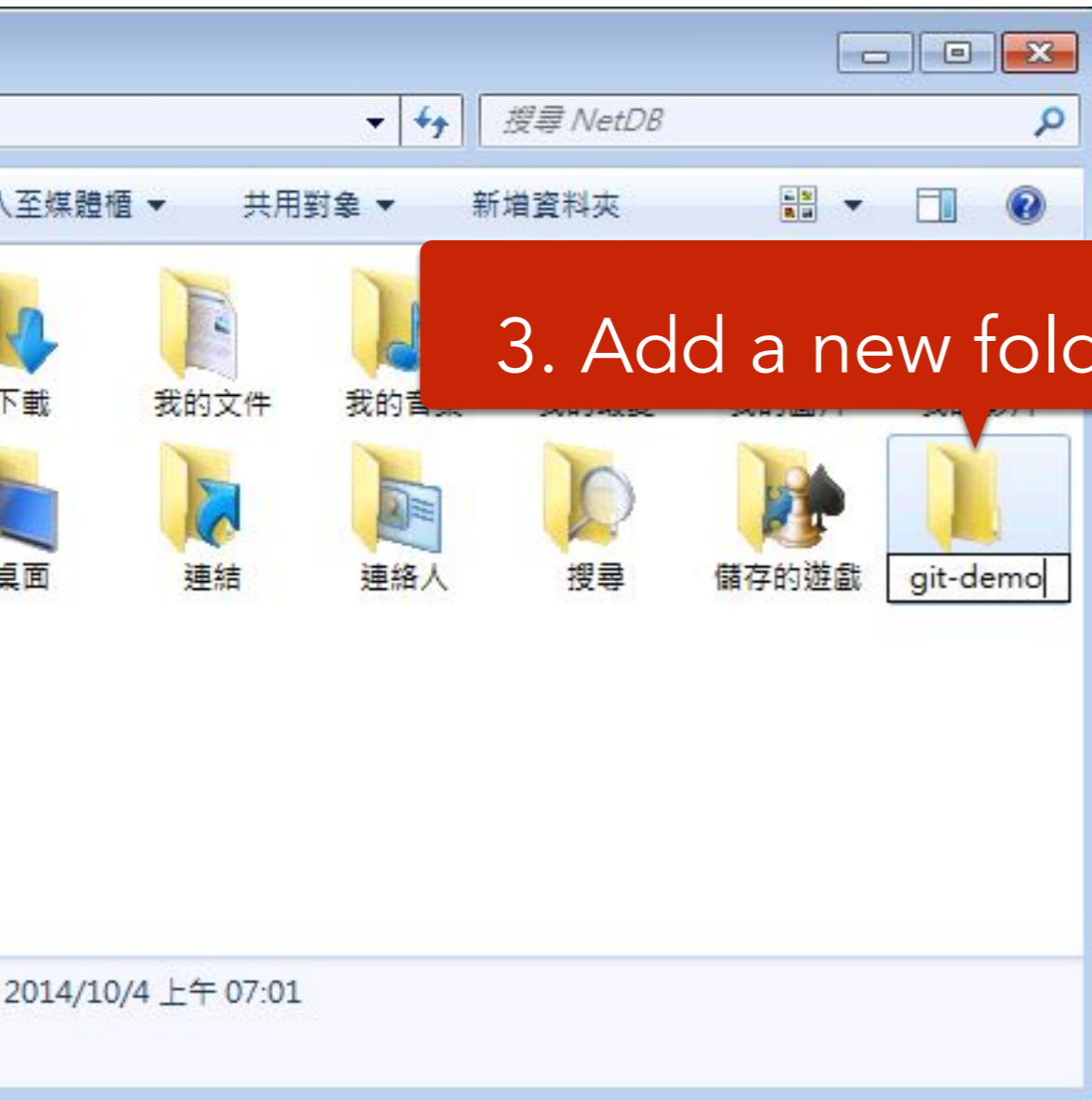
- **git init**
  - Initialize a repository at current directory.
- **git add [file\_name] ( \* git add . \* → means add all files )**
  - Add files to git repository and let git track them.
- **git commit -m "commit messages"**
  - Save the changes to the git repository and create snapshots of the files.
- **git checkout [version]**
  - Go to a specific version.

2. Go to that directory

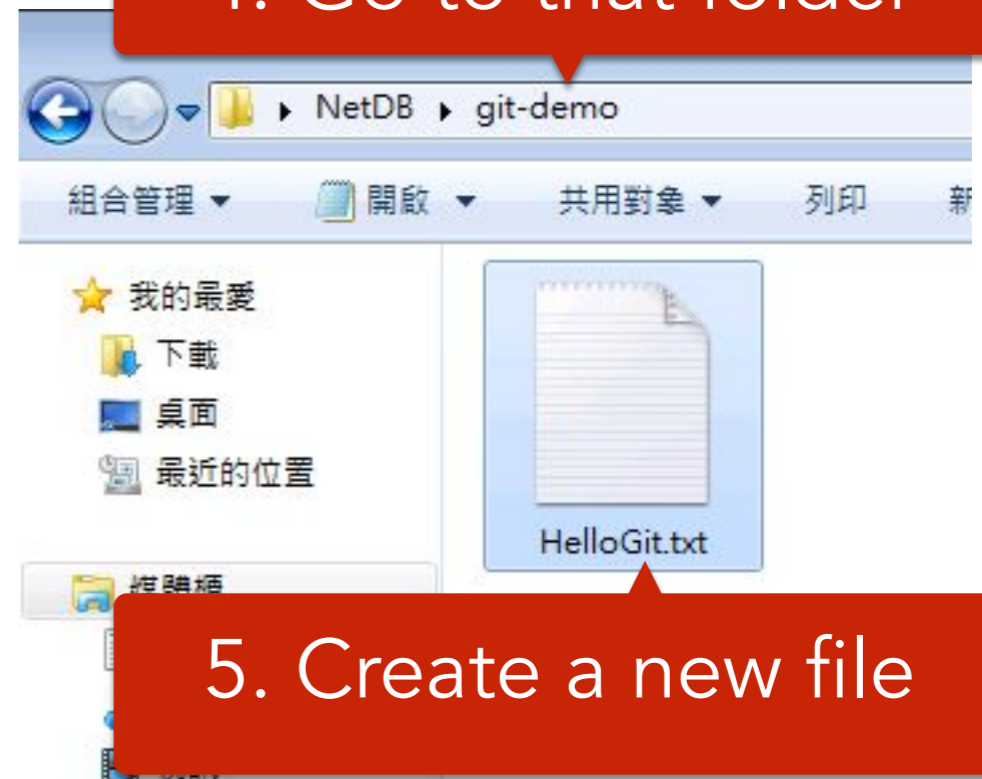
C:\Users\NetDB>

1. Find current directory

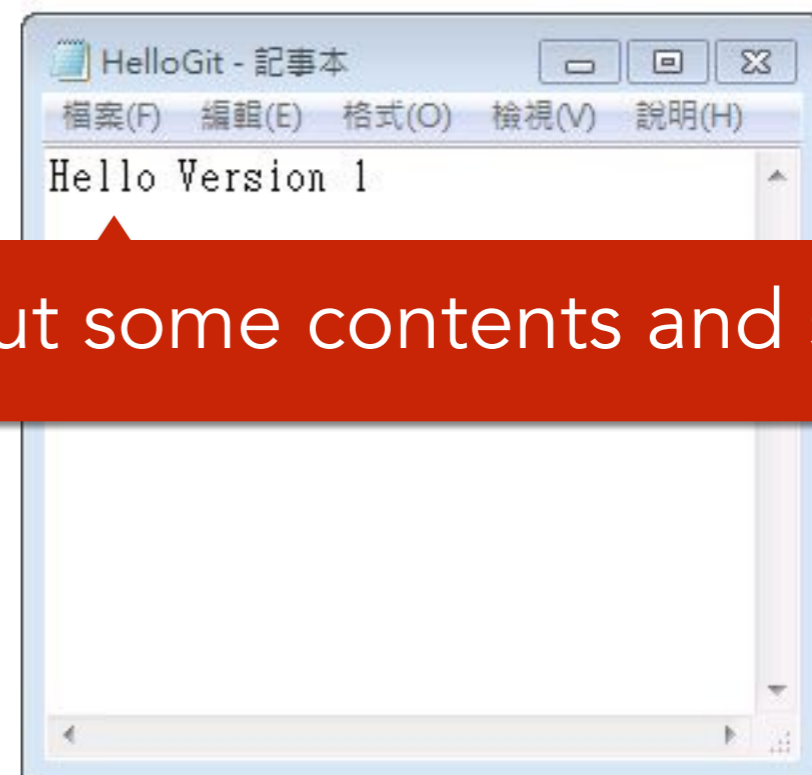




3. Add a new folder



5. Create a new file



6. Put some contents and save

```
命令提示字元
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

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

C:\Users\NetDI>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"
```

```
命令提示字元
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

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 07:12 <DIR> .
..
15 HelloGit.txt
15 位元組
5,944 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
```

```
命令提示字元
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\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

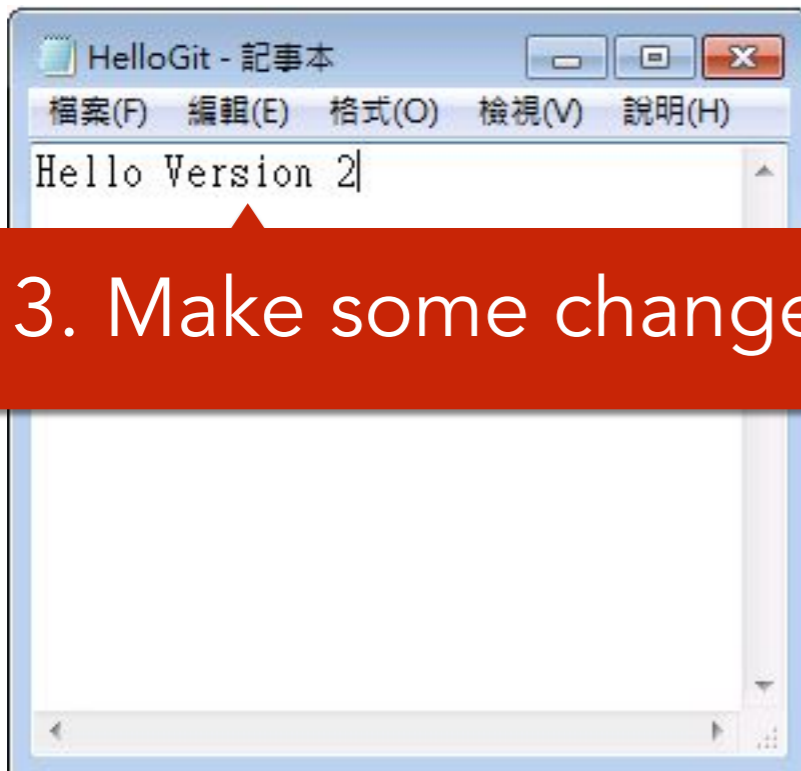
# where "version 1" is the commit message

\$ git commit -m "version 1"

14. Add it and commit again

```
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(-)
```

13. Make some changes and save



## 15. View your versions

Version  
ID

```
C:\Users\NetDB\git-demo>git log
commit e134c845df593f1451c4e7e6c874ddef6df42a76
Author: cyhsu <cyhsu@netdb.cs.nthu.edu.tw>
Date: Sat Oct 4 08:09:55 2014 +0800

    version 2

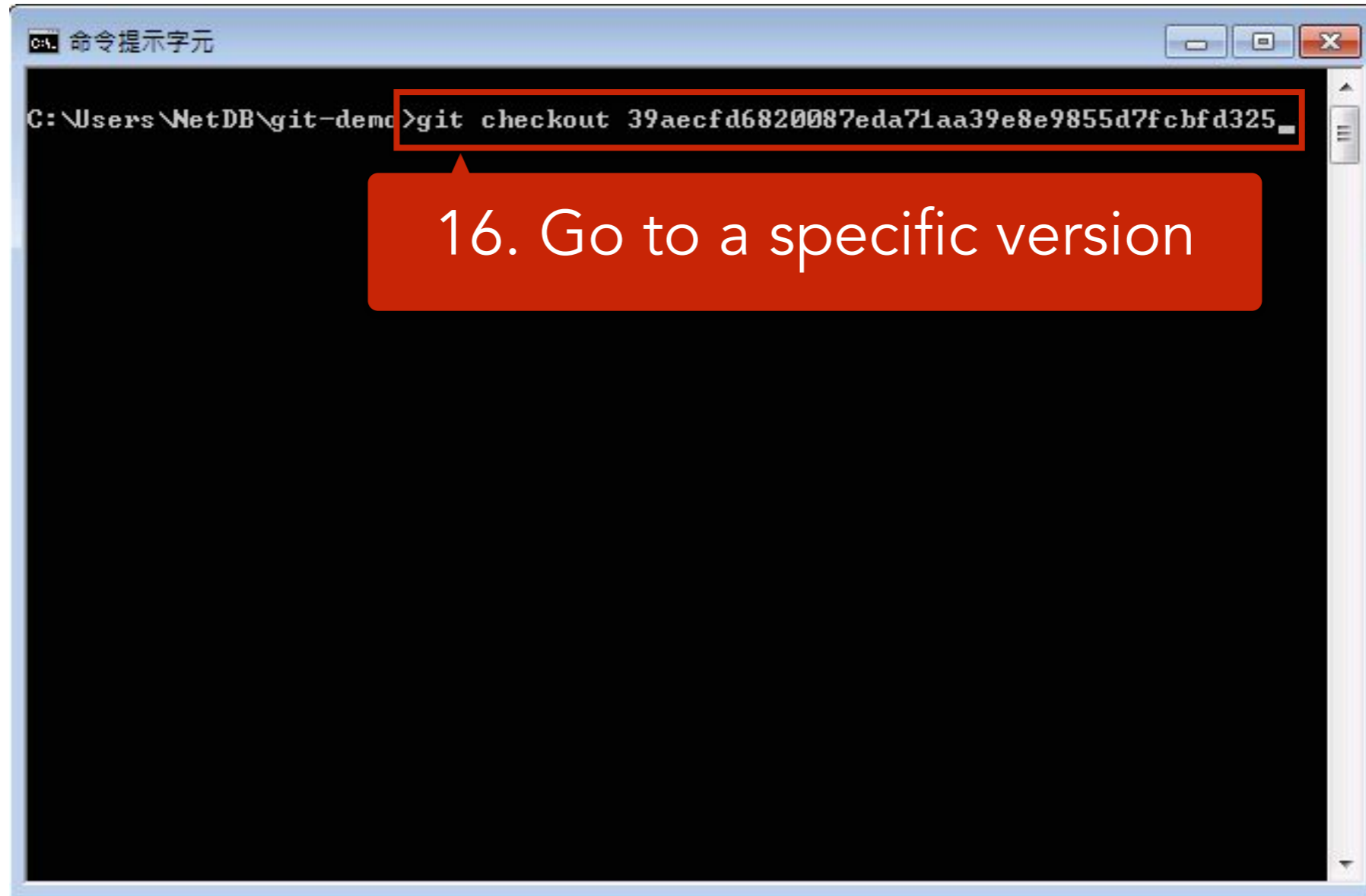
commit 39aecfd6820087eda71aa39e8e9855d7fcbfd325
Author: cyhsu <cyhsu@netdb.cs.nthu.edu.tw>
Date: Sat Oct 4 08:09:16 2014 +0800

    version 1
```

Commit  
messages

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





A screenshot of a Windows command prompt window. The title bar reads '命令提示字元'. The current directory is 'C:\Users\NetDB\git-demo'. The command entered is 'git checkout 39aecfd6820087eda71aa39e8e9855d7fcbfd325'. A red box highlights the command, and a red callout bubble points to it with the text '16. Go to a specific version'.

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

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

LIFE IS  
TOO SHORT  
TO TYPE  
THAT  
VERSION ID,

which is 40 characters long...

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!

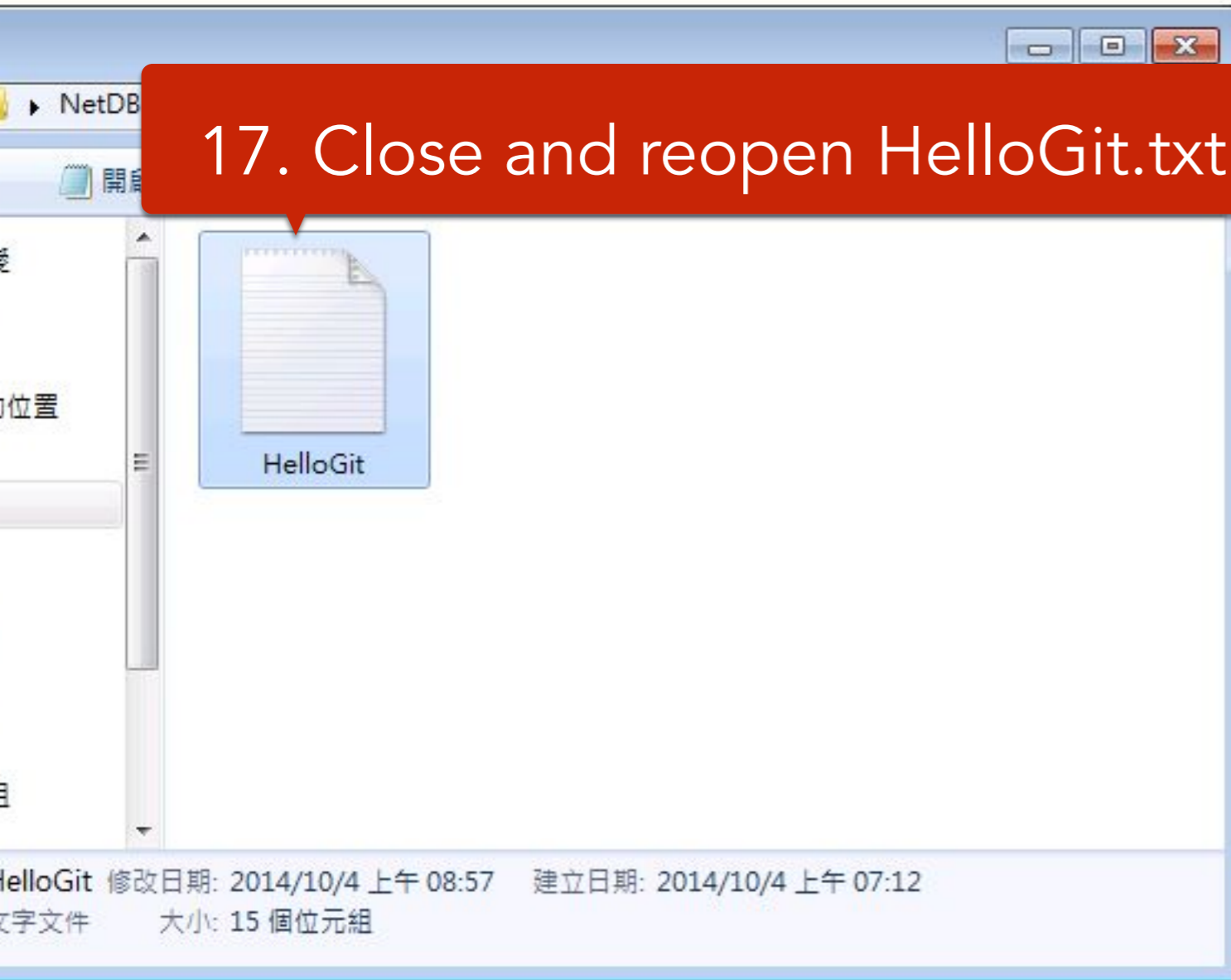
```
# Show versions with short version id
$ git log --oneline
```

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

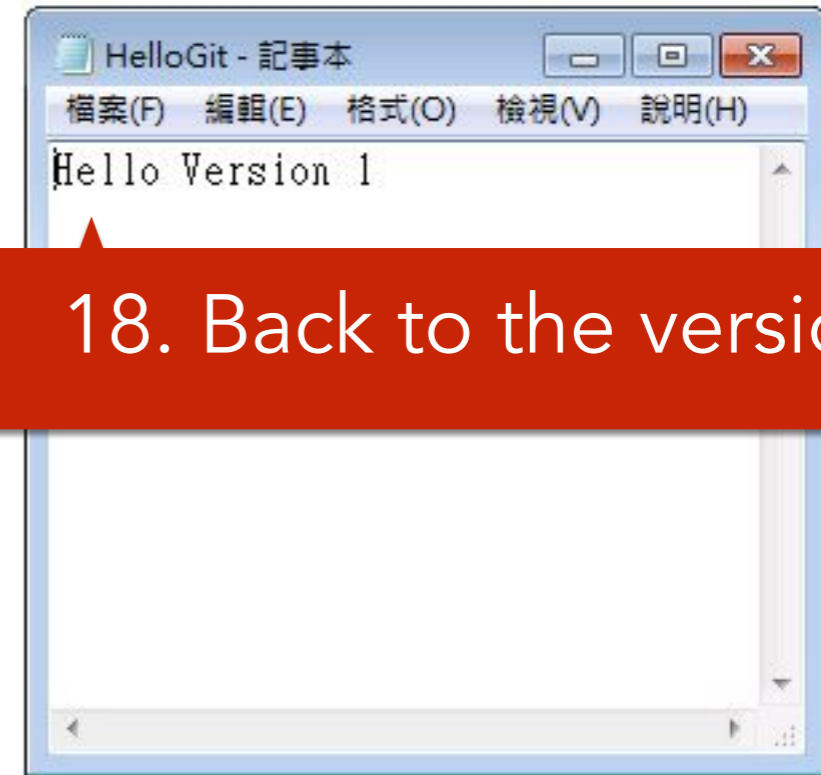
16. Go to a specific version

```
# 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



18. Back to the version 1!



# Try yourself (1/2)

- Branching steps
  - Creating a new branch

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

- Checking out the branch

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

# Try yourself (2/2)

- Merging steps
  - Checking out a branch to merge

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

- Merging another branch

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

# Outline

- General Rule
- Introduction to Git
  - Version control
  - Git Basics
  - Try Git!
  - Remote Repositories
- How to Submit Your Code to Gitlab
- Tools & References



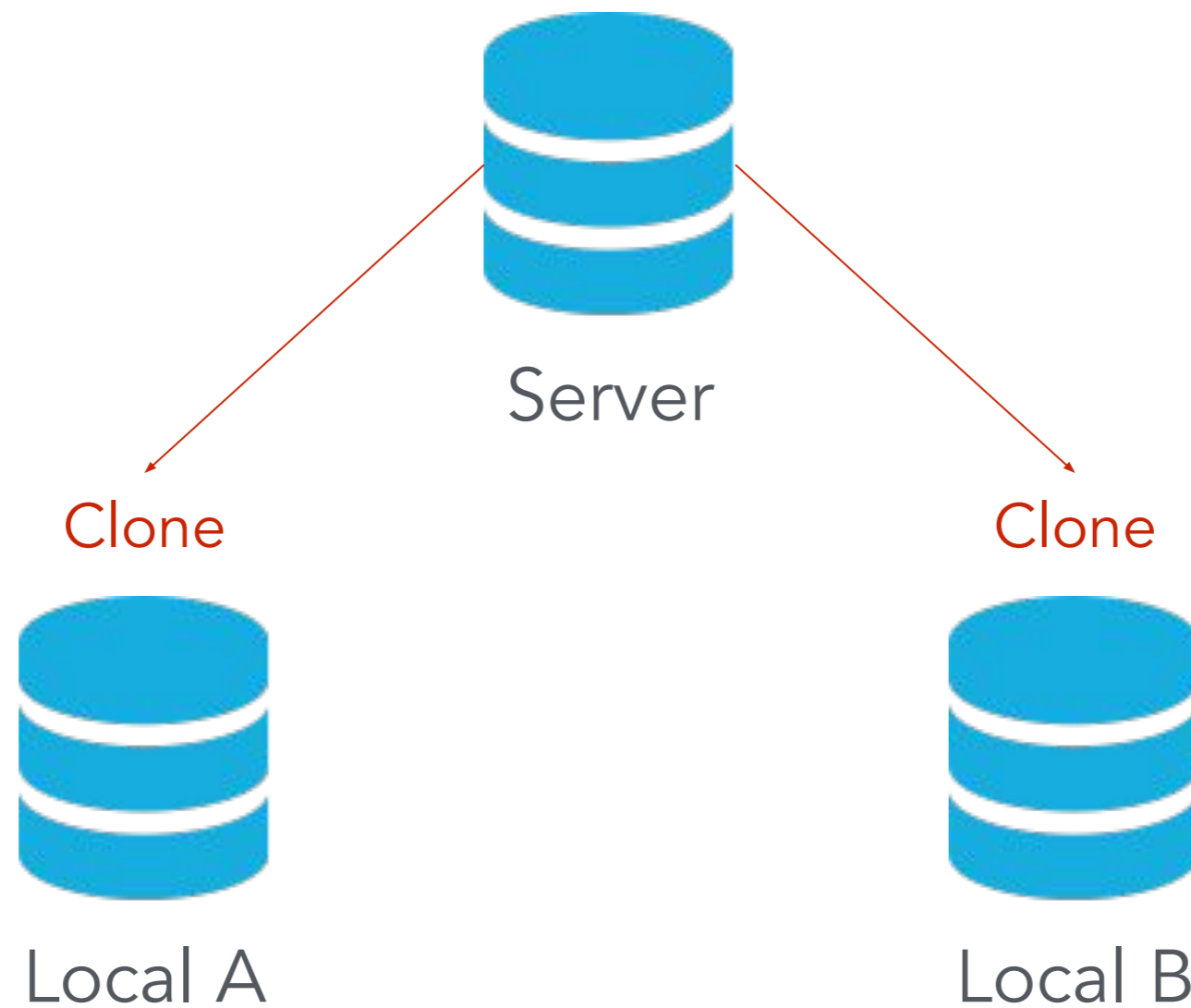
# Collaboration

- 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 to their computer and do their jobs independently.

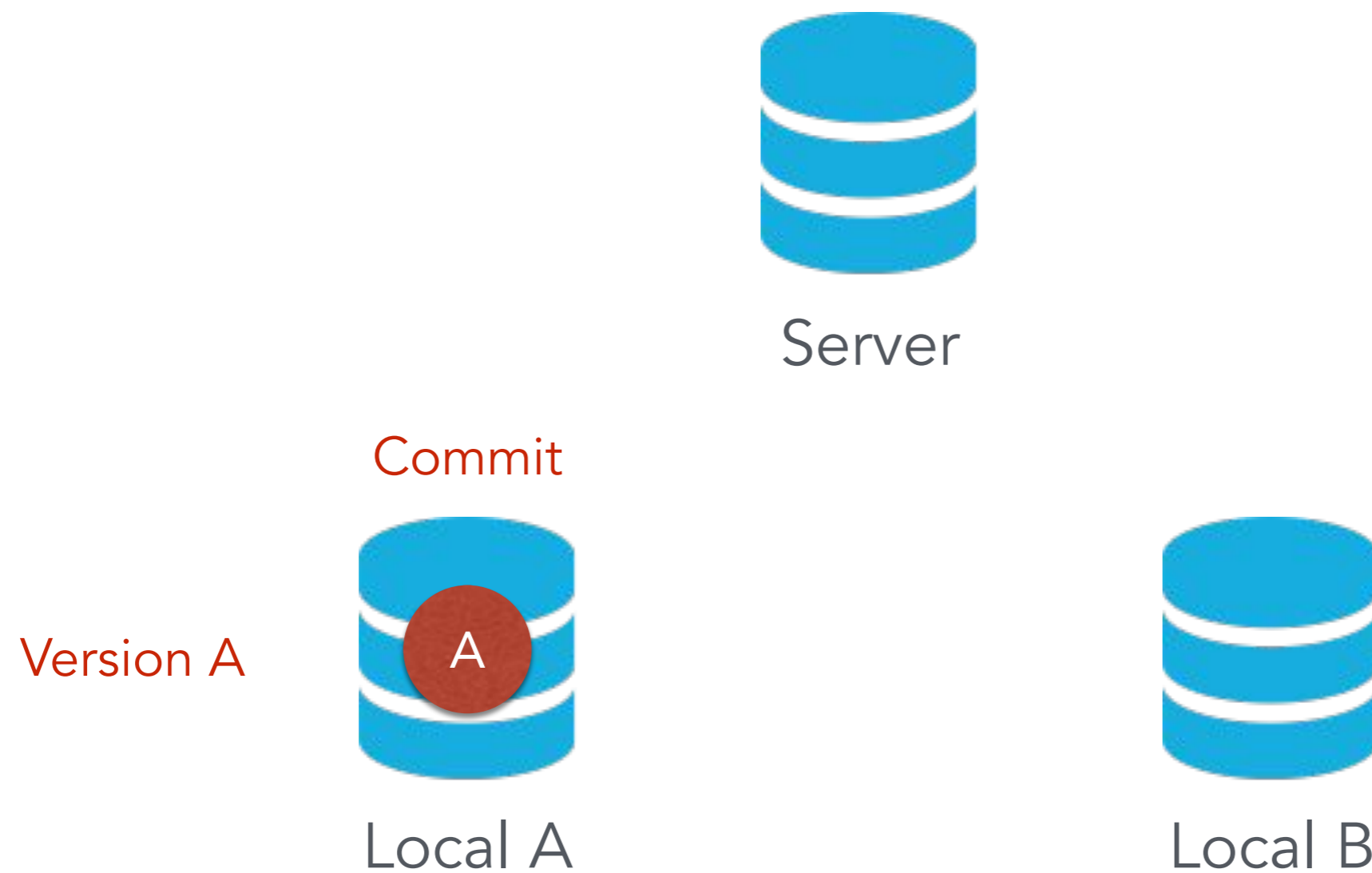
# Collaboration Workflows (1/2)

1. If you don't have the project, *clone* (download) the repository from the server.
2. Do your work and commit the changes at local. Once done, *push* (upload) the repository to the server.
3. If someone else modified the project, you can *pull* (sync) the repository to get the updated project.
4. Repeat 2 and 3.

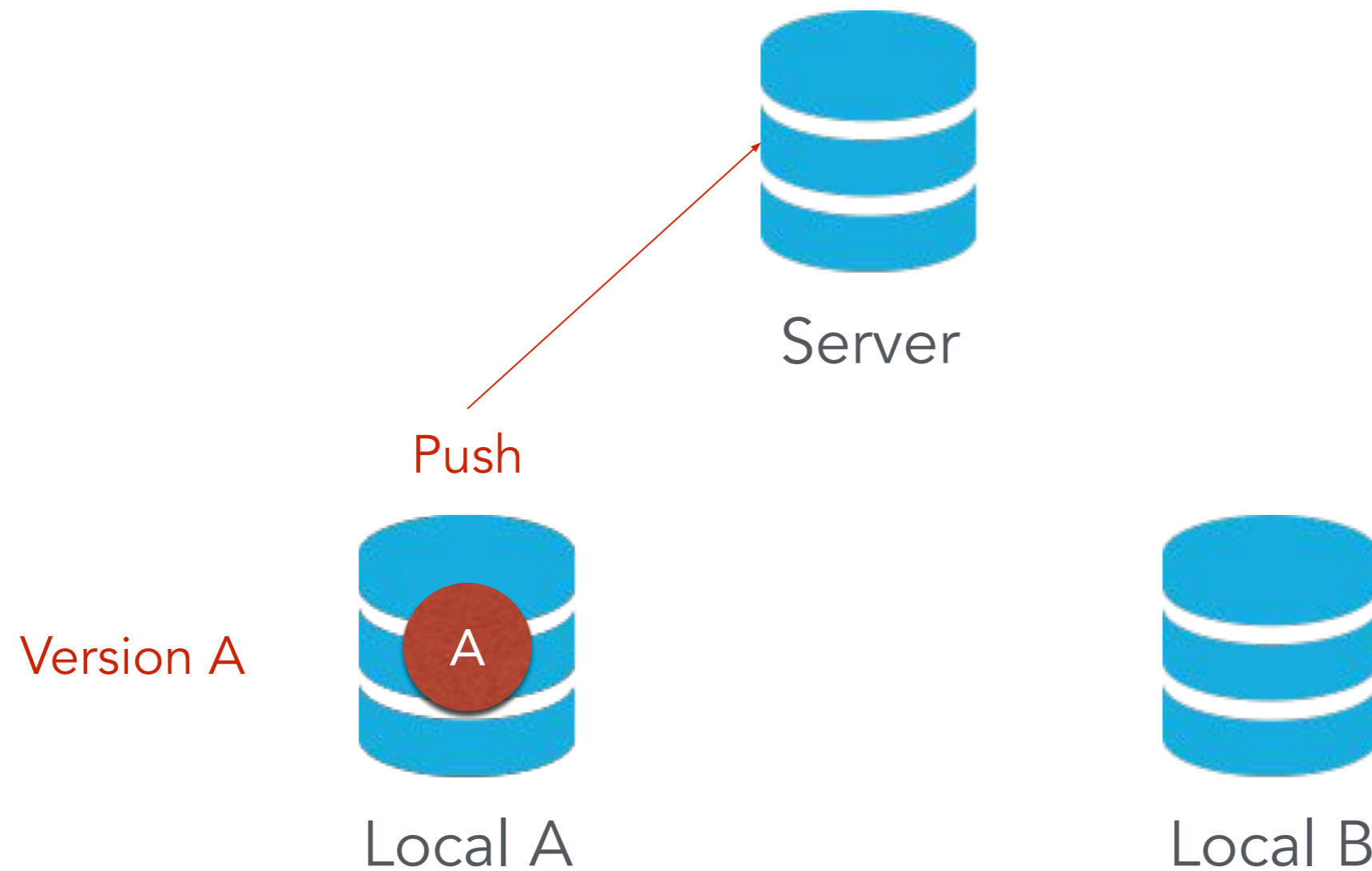
# Collaboration Workflows (2/2)



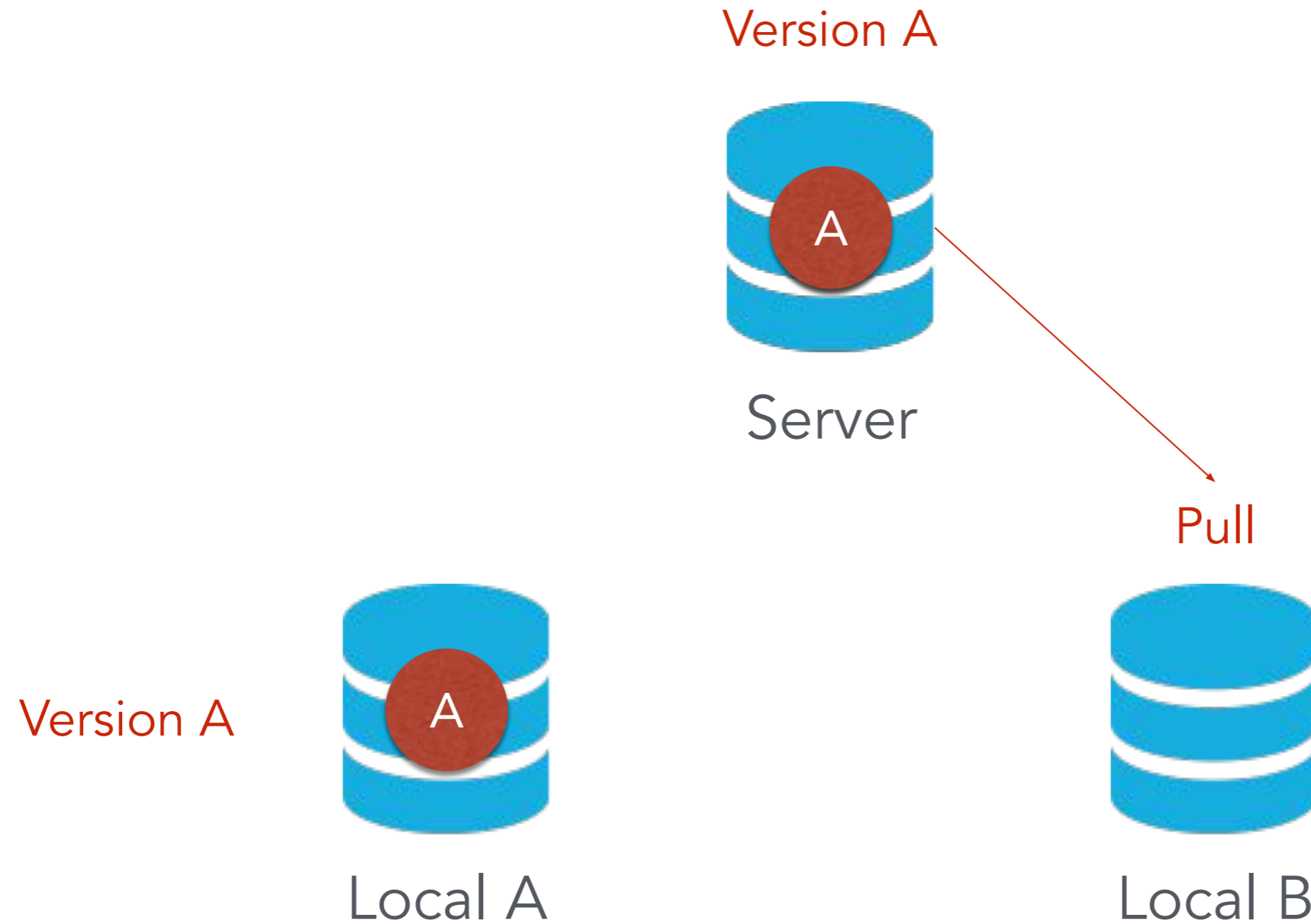
# Collaboration Workflows (2/2)



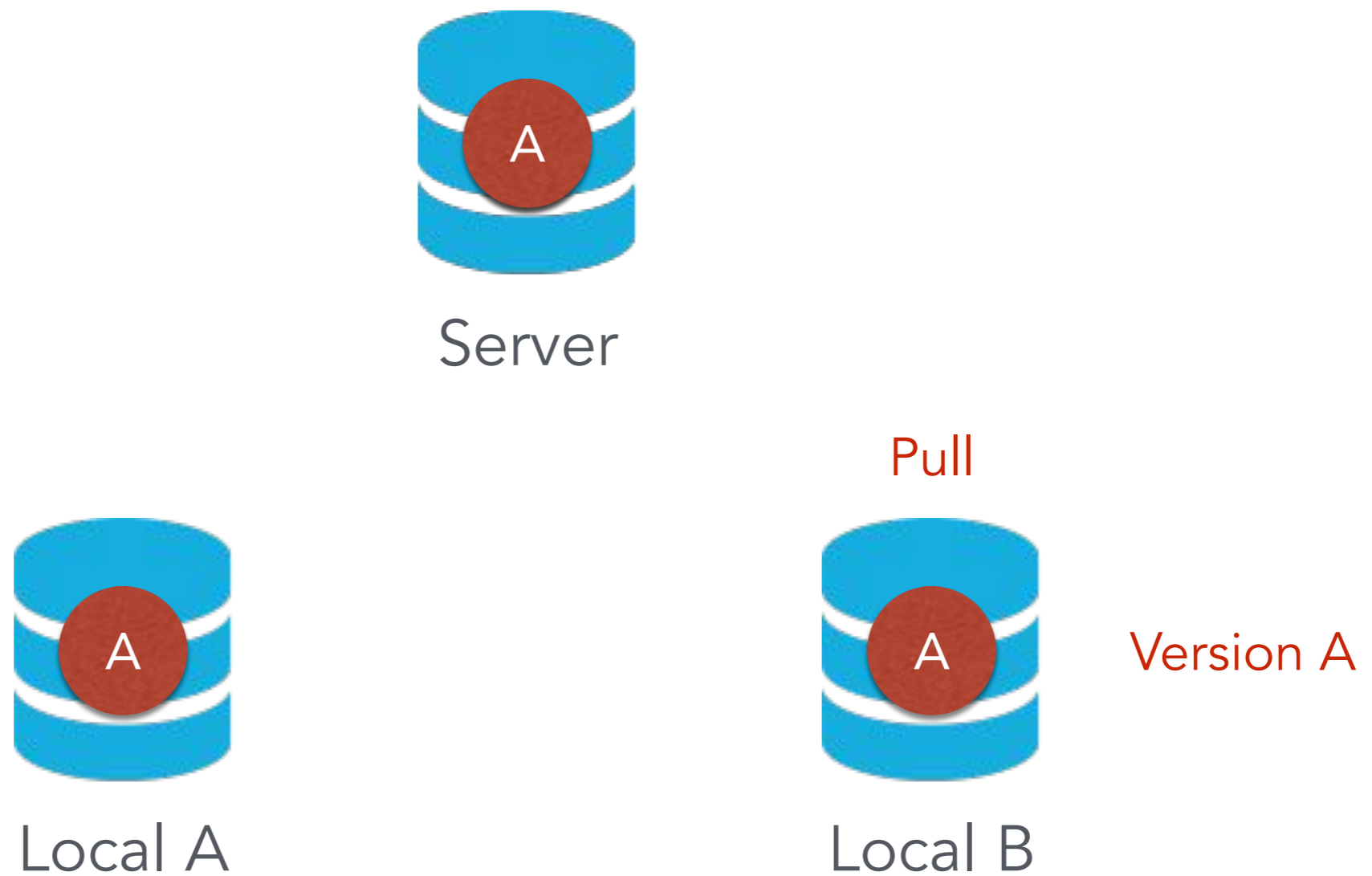
# Collaboration Workflows (2/2)



# Collaboration Workflows (2/2)



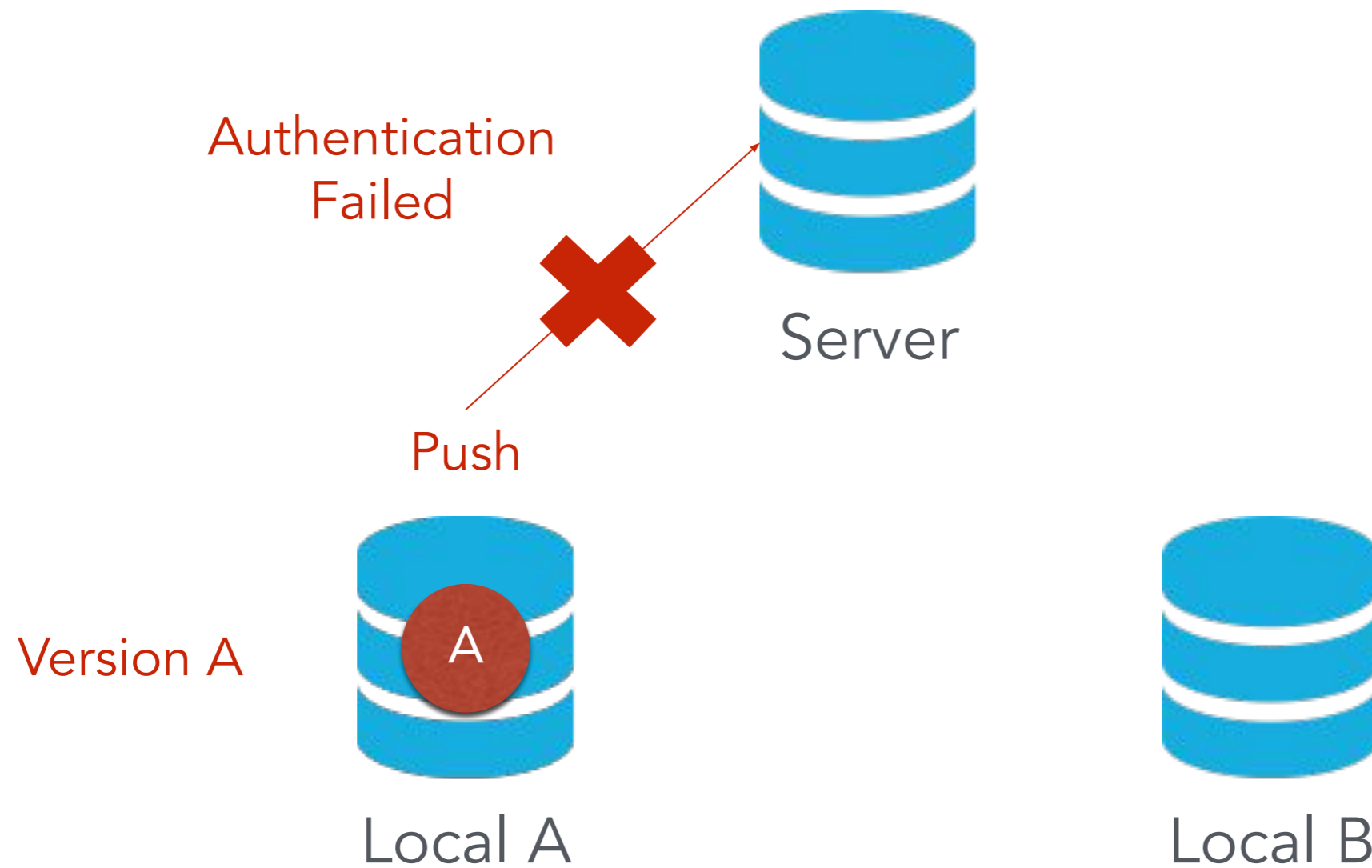
# Collaboration Workflows (2/2)



Something went wrong.



# Authentication Failed



# Why Authentication Failed?

# Collaboration Workflow

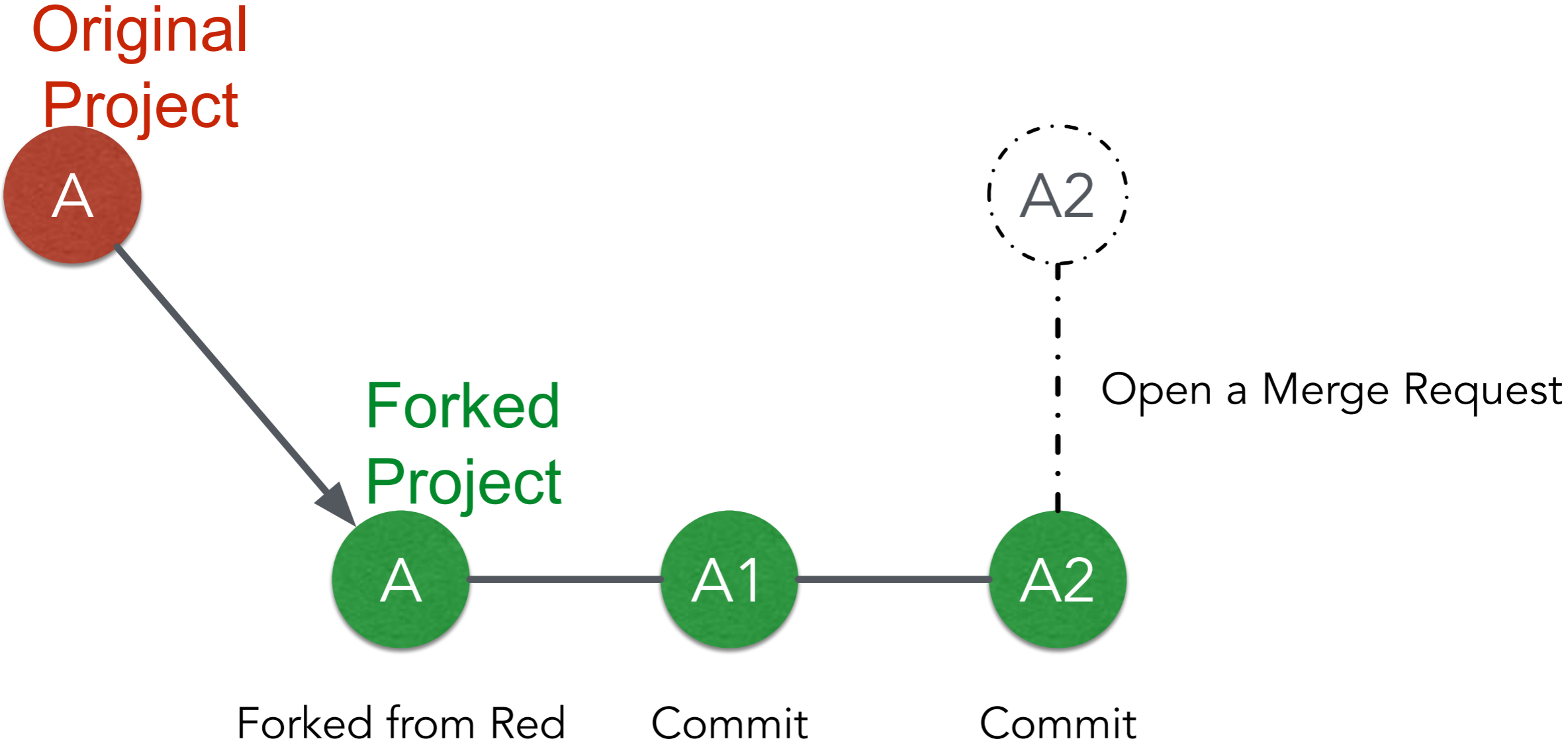
- If you tried to clone the code template from a server and want to **push** the modified file.
  - You will get authentication failed.
  - It's because it was a **project of others**, which means you are not able to save the changes back to the server.
- So, how can I copy a project from others on a open source platform like Github?

Introducing

# Fork

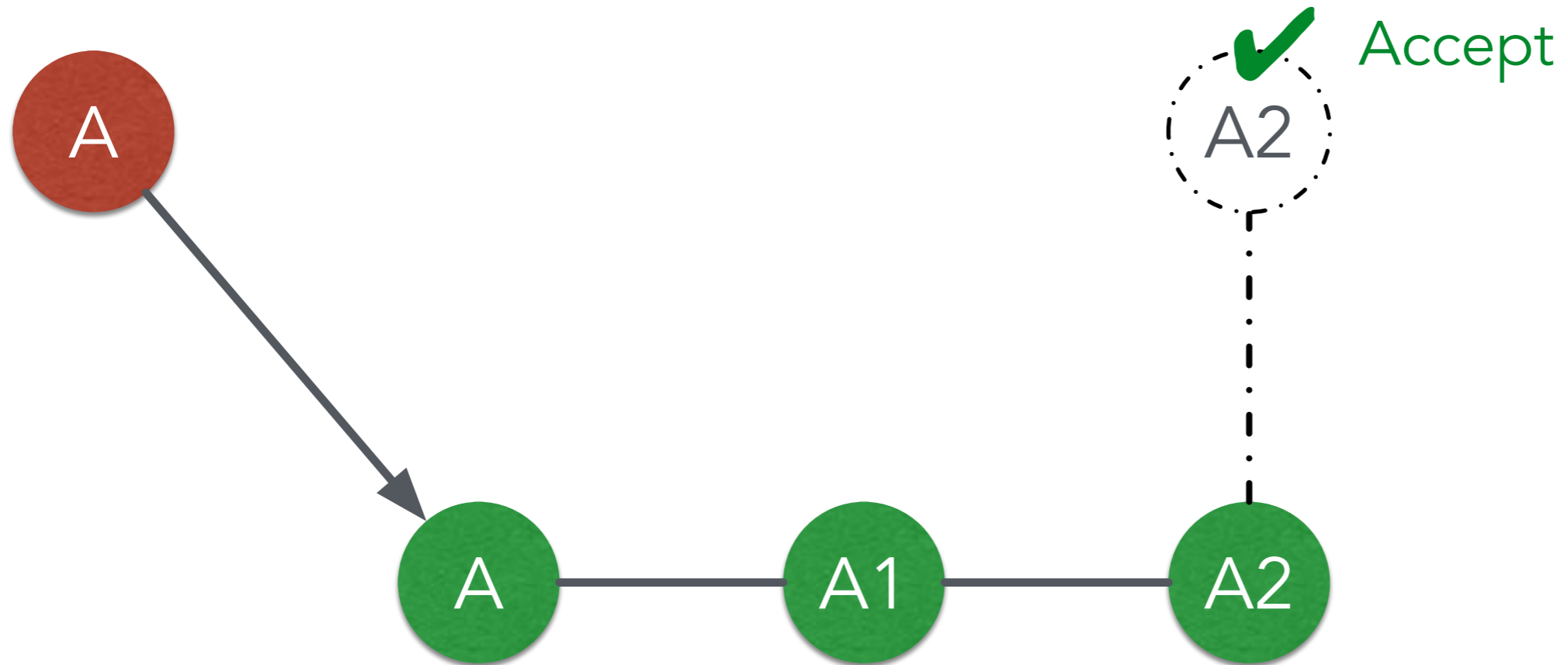


# Author



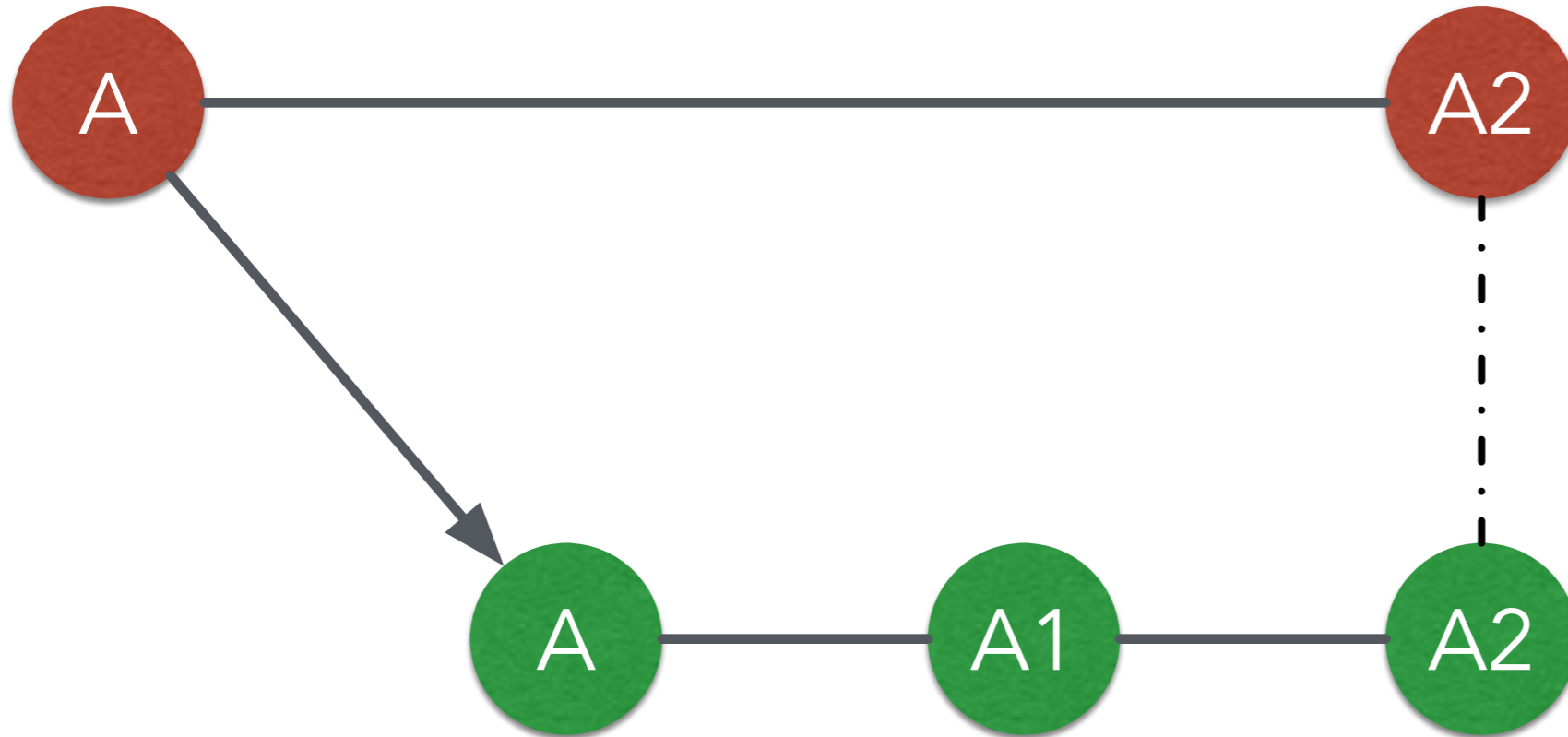
# Editor

Author

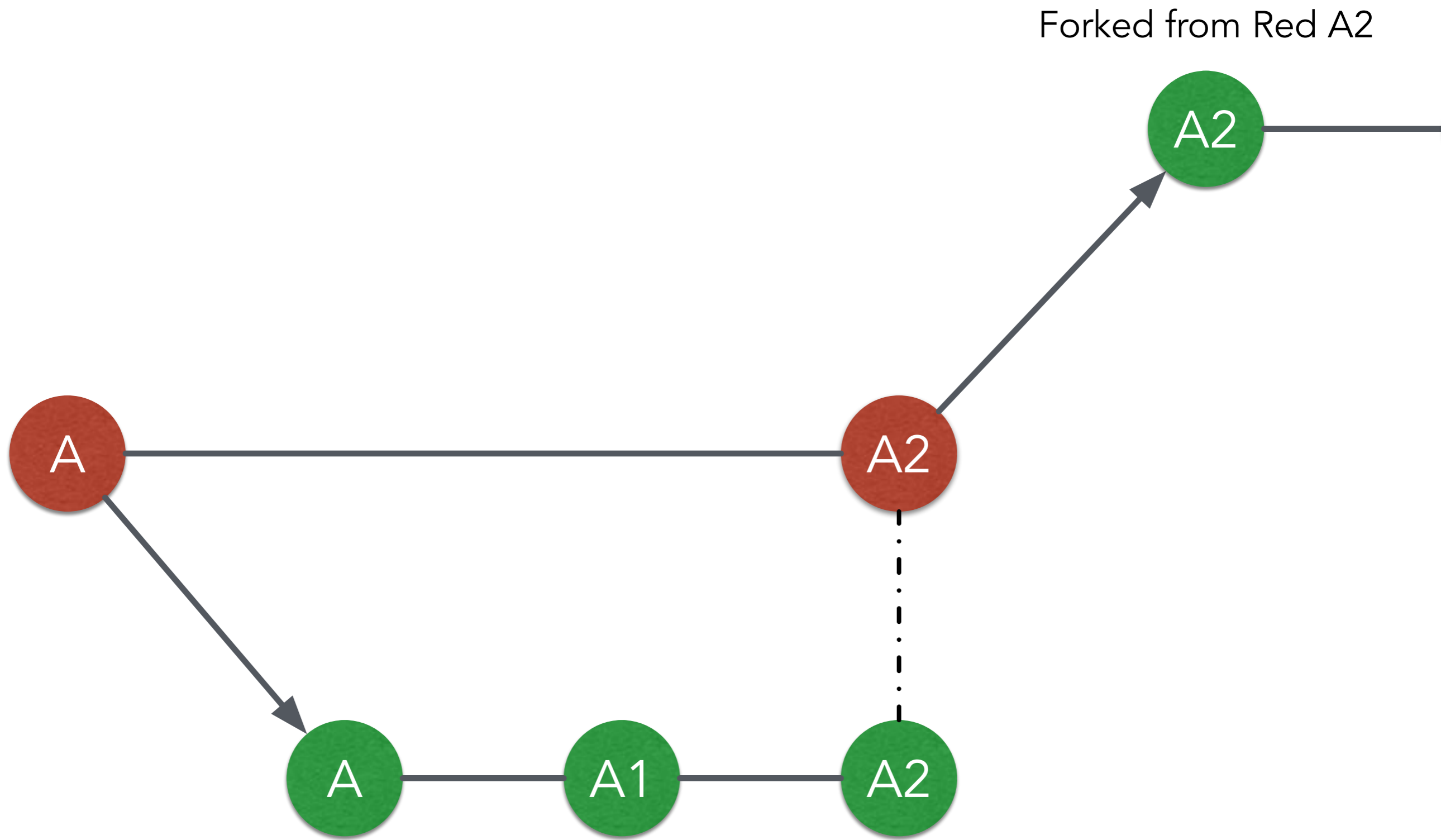


Editor

Author



Editor





# Git Collaboration Workflow

1. *Fork* a repository to make a copy of it.
2. *Clone* the repository you forked to your workspace.
3. Do your work and *commit* the changes in your workspace.
4. *Push* the repository to the server to synchronize them.
5. Open a *merge request* to origin repository .

# Basic Git Commands (2/2)

- **git clone [url]**
  - Clone a repository from remote server
- **git push [url] [branch-name]**
  - Push committed file to remote server

# Outline

- General Rule
- Introduction to Git
  - Version control
  - Git Basics
  - Try Git!
  - Remote Repositories
- **How to Submit Your Code to Gitlab**
- Tools & References



# Gitlab

- We have created account for you
- Account: student ID (e.g. 106012345)
- Password: student ID (e.g. 106012345)

# Gitlab

The image shows a Gmail interface. At the top left, there is a hamburger menu icon, the Gmail logo, and the word "Gmail". To the right is a search bar with the text "搜尋郵件". Below the search bar is a row of action icons: a minus sign, a plus sign, a warning sign, a trash can, an envelope, a clock, a right arrow, a speech bubble, and a vertical ellipsis. On the left side, there is a "撰寫" (Compose) button with a plus sign icon. Below it is the "收件匣" (Inbox) section, which is highlighted in red and shows a count of "9". Other sections include "已加星號" (Starred) and "已延後" (Deferred). The main inbox area shows a list of emails. The first email is from "GitLab" with the subject "Access to the courses / software-studio / 2020-spring". The second email is also from "GitLab" with the subject "Confirmation instructions - Welcome, TA\_ACOUNT! T". Above the second email, there is a notification for "社交網路" (Social network) with "33 個新對話" (33 new conversations) and "Twitter". To the right, there is a "促銷內容" (Promotional content) section.

≡ Gmail

搜尋郵件

撰寫

收件匣 9

已加星號

已延後

主要

社交網路 33 個新對話  
Twitter

促銷內容

☆ GitLab Access to the courses / software-studio / 2020-spring

☆ GitLab Confirmation instructions - Welcome, TA\_ACOUNT! T

# Gitlab

Confirmation instructions 收件匣 ×



GitLab <gitlab@shwu10.cs.nthu.edu.tw>

下午8:11 (2分鐘前)



寄給我 ▾

英文 ▾ > 中文(繁體) ▾ [翻譯郵件](#)

[關閉下列語言的翻譯功能：英文 ×](#)



Welcome, TA\_ACCOUNT!

To get started, click the link below to confirm your account.

[Confirm your account](#)



GitLab

You're receiving this email because of your account on [shwu10.cs.nthu.edu.tw](https://shwu10.cs.nthu.edu.tw). [Manage all notifications](#) · [Help](#)

# Gitlab



Your email address has been successfully confirmed. Please sign in.

## DataLab

Welcome to GitLab for DataLab.

Sign in

Username or email

Password

Remember me [Forgot your password?](#)

Sign in



# 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

You can access course projects in [this group](#)

courses >  > 2024-spring

2

**2024-spring** 

Group ID: 3517  [Leave group](#)



New subgroup

New project

**Subgroups and projects**

Shared projects


Archived projects

Search by name

Last created





S

submission-exercise 

★ 0

3 days ago

**S** **submission-exercise** 

Project ID: 6444 



   Star 0  Fork 0


1. Click to fork

0 Commits  1 Branch  0 Tags  20 KB Files  20 KB Storage

master  submission-exercise /  

History Find file Web IDE   **Clone** 


 **Initial commit** 21779350 

 Chen Yu-Hsuan authored 3 days ago

 Upload File  Auto DevOps enabled  Add README  Add LICENSE  Add CHANGELOG  Add CONTRIBUTING

 Configure Integrations

Name	Last commit	Last update
 README.md	Initial commit	3 days ago

 **README.md**

## Lab 1 - 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 a lab or project.

### Try it!

1. Fork this project.
2. Clone the **forked** project from Gitlab to your local environment.
3. Add a new file and write something.
4. Commit your work.
5. Push the repository to the server.
6. Send a merge request of **your branch** to the origin repository.



## Fork project

A fork is a copy of a project.  
Forking a repository allows you to make changes without affecting the original project.

### Project name

submission-exercise

### Project URL

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

Select a namespace

### Project slug

submission-exercise

Want to house several dependent projects u

2. Select your name

### Project description (optional)

### Visibility level [?](#)

Private

Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

Internal

The project can be accessed by members of the group.

Public

The project can be accessed without any authentication.

3. Click Private

Fork project

Cancel

4. Click to fork

S submission-exercise

- Project information
- Repository
- Issues 0
- Merge requests 0
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Packages & Registries
- Analytics
- Wiki
- Snippets
- Settings

Chen Yu-Hsuan > submission-exercise

5. Check if this repository is under your account

Project ID: 3798

🔑 🔔 ⭐ Star 0 🍴 Fork 0

🔗 3 Commits 🌿 99 Branches 🏷️ 0 Tags 📄 102 KB Files 🗄️ 113 KB Storage

Forked from [courses / software-studio / 2023-spring / submission-exercise](#)

master submission-exercise / + History Find file Web IDE ↓ Clone



finish lab 1

Chen Yu-Hsuan authored 1 year ago

1654ad5c



📄 Upload File

📄 README

⚙️ Auto DevOps enabled

⊕ Add LICENSE

⊕ Add CHANGELOG

⊕ Add CONTRIBUTING

⚙️ Configure Integrations

Name

Last commit

Last update

📄 README.md

finish lab 1

1 year ago

6. Go to settings

- S submission-exercise
- Project
- Repository
- Issues 0
- Merge Requests 0
- Wiki
- Snippets
- Settings
  - General
  - Members
  - Integrations
  - Repository

alan0313 > submission-exercise > General Settings

### General project

Expand

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

### Permissions

Collapse

Enable or disable certain project features and choose access levels.

7. Set project to private

#### 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

#### Pipelines

Build, test, and deploy your changes

Enable feature to choose access level

#### Git Large File Storage

Manages large files such as audio, video, and graphics files

<< Collapse sidebar

- S submission-exercise
- Project
- Repository
- Issues 0
- Merge Requests 0
- Wiki
- Snippets
- Settings
- General
- Members
- Integrations
- Repository

**Wiki**  
Pages for project documentation

Only Project Members

**Snippets**  
Share code pastes with others out of Git repository

Only Project Members

Save changes

8. Scroll down and save changes

**Merge request** Expand  
Customize your merge request restrictions.

**Badges** Expand  
Customize your project badges. [Learn more about badges.](#)

**Export project** Expand  
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.



General

**Advanced** Expand  
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


**S** **submission-exercise**   
Project ID: 6444 







   Star 0  Fork 0


0 Commits 1 Branch 0 Tags 20 KB Files 20 KB Storage

master submission-exercise / +

History Find file Web IDE  Clone

 **Initial commit**  
Chen Yu-Hsuan authored 3 days ago

 Upload File  Auto DevOps enabled  Add README  Add LICENSE  Add  Configure Integrations

Name	Last commit
 README.md	Initial commit


## README.md

### Lab 1 - 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 a lab or project.

#### Try it!

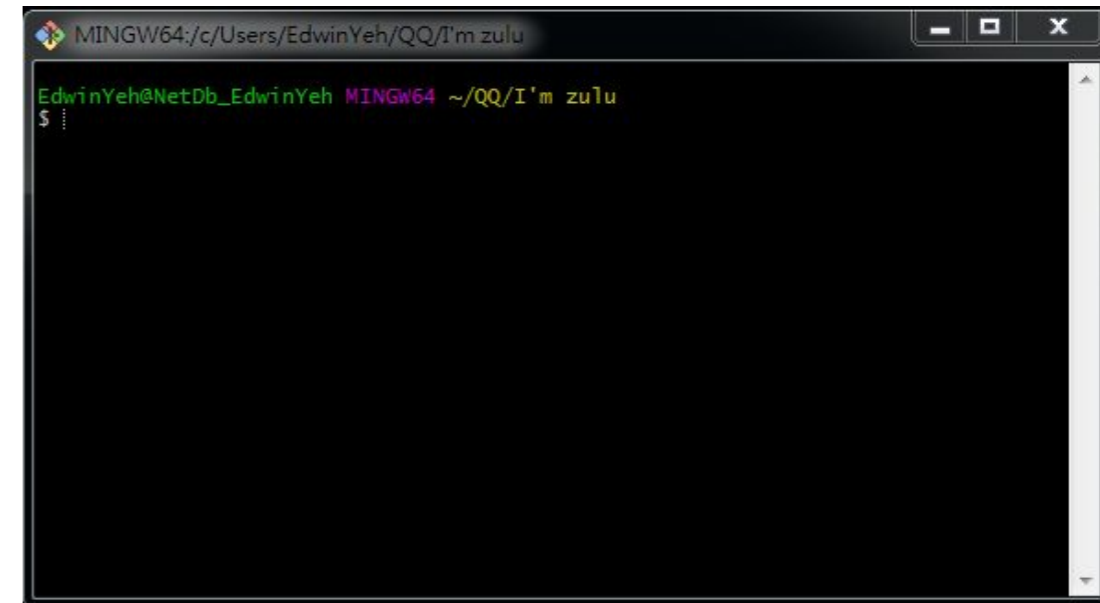
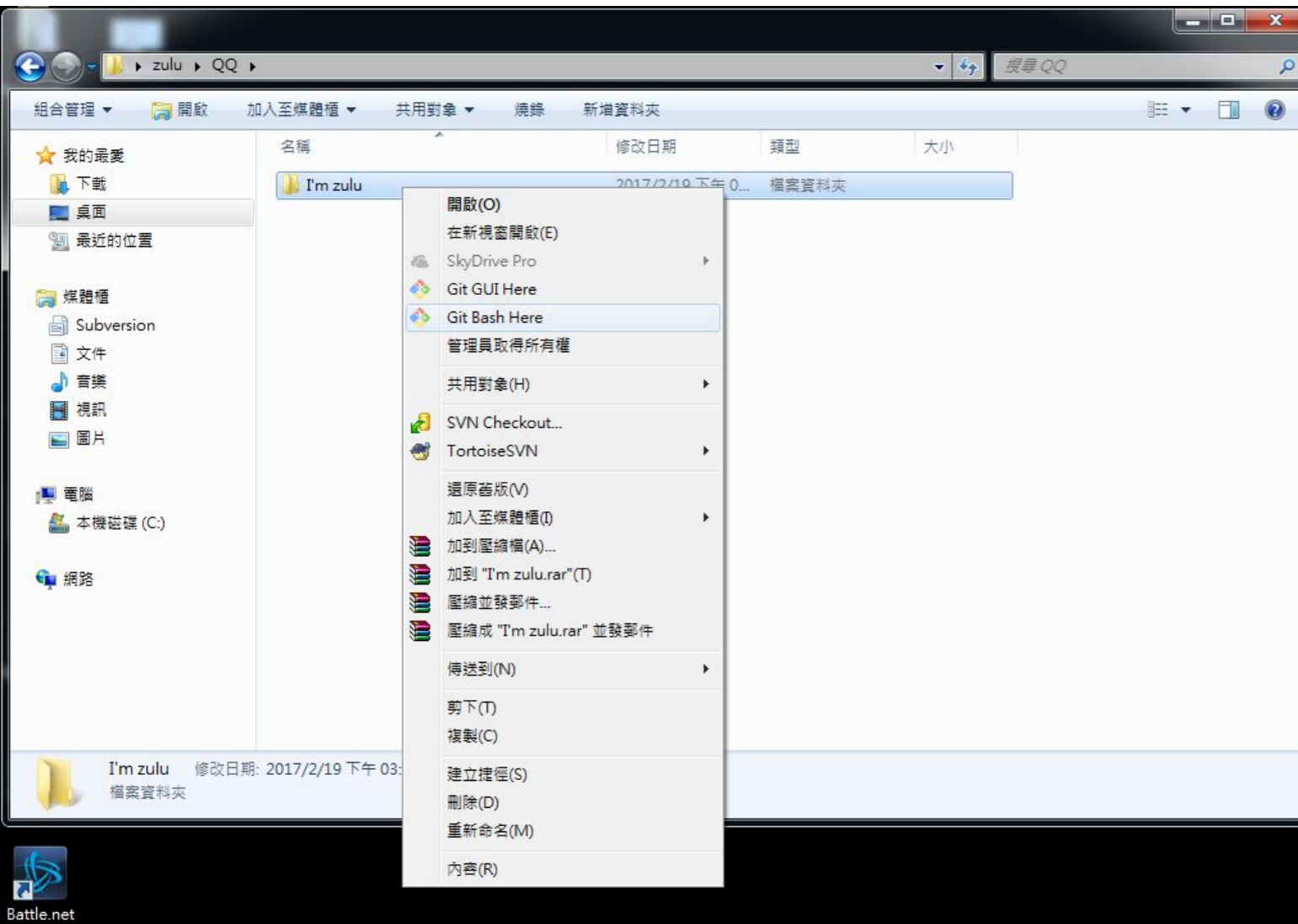
1. Fork this project.
2. Clone the **forked** project from Gitlab to your local environment.
3. Add a new file and write something.
4. Commit your work.
5. Push the repository to the server.
6. Send a merge request of **your branch** to the origin repository.

Clone with SSH  
git@shwu10.cs.nthu.edu.tw:  
Clone with HTTPS  
https://shwu10.cs.nthu.edu.tw/cour 

2. Copy the link

1. Choose HTTPS

# If You use Windows



### 3. Create a folder to put your repos

```
~/SS-Projects ➤ git clone https://shwu10.cs.nthu.edu.tw/ss-student/submission-exercise.git
Cloning into 'submission-exercise'...
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.
~/SS-Projects ➤ ls
submission-exercise
~/SS-Projects ➤
```

### 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

```
~/SS-Projects/submission-exercise master vim lab1.js
~/SS-Projects/submission-exercise master git add -A
~/SS-Projects/submission-exercise master + git status
```

1. -A means all files

```
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   lab1.js
```

2. Check if your file is added to git

```
~/SS-Projects/submission-exercise master + git commit -m "Finish lab1"
[master c1acaf4] Finish lab1
1 file changed, 1 insertion(+)
create mode 100644 lab1.js
~/SS-Projects/submission-exercise master
```

3. Commit your changes

```
~/SS-Projects/submission-exercise master vim lab1.html
~/SS-Projects/submission-exercise master git add -A
~/SS-Projects/submission-exercise master + git commit -m "Finish lab1"
```

```
[master 8a603d9] Finish lab1
```

```
Committer: Real Wei <realwei@Realweis-MBP.local>
```

```
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:
```

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

```
After doing this, you may fix the identity used for this commit with
```

```
git commit --amend --reset-author
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 lab1.html
```

```
~/SS-Projects/submission-exercise master
```

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



```
~/SS-Projects/submission-exercise ? master git push -u origin master
```

```
Counting objects: 6, done.
```

```
Delta compression using up to 4 threads.
```

```
Compressing objects: 100% (4/4), done.
```

```
Writing objects: 100% (6/6), 497 bytes | 0 bytes/s, done.
```

```
Total 6 (delta 1), reused 0 (delta 0)
```

```
To https://shwu10.cs.nthu.edu.tw/ss-student/submission-exercise.git
```

```
 b1e0571..8a603d9  master -> master
```

```
Branch master set up to track remote branch master from origin.
```

```
~/SS-Projects/submission-exercise ? 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

- S submission-exercise
- Project information
- Repository
- Issues 0
- Merge requests 0
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Packages & Registries
- Analytics
- Wiki
- Snippets
- Settings

Chen Yu-Hsuan > submission-exercise

**S submission-exercise**

Project ID: 3798

Star 0 Fork 0

**1. Click Merge Requests**

KB Files 113 KB Storage

Forked from [courses / software-studio / 2023-spring / submission-exercise](#)

master submission-exercise / +

History Find file Web IDE Clone

**finish lab 1** 1654ad5c

Chen Yu-Hsuan authored 1 year ago

Upload File
 README
 Auto DevOps enabled
 Add LICENSE
 Add CHANGELOG
 Add CONTRIBUTING

Configure Integrations

Name	Last commit	Last update
README.md	finish lab 1	1 year ago

- S submission-exercise
- Project
- Repository
- Issues 0
- Merge Requests 0
- Wiki
- Snippets
- Settings



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.

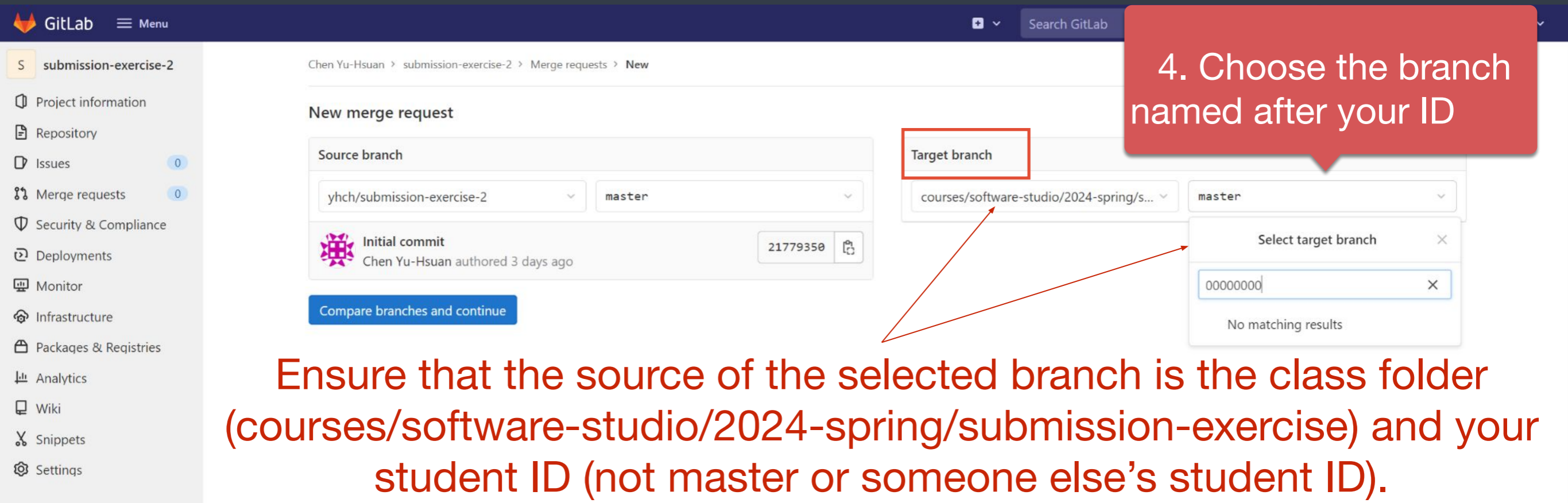
New merge request

New merge request

## 2. New merge request

3. Choose the branch you pushed in your repo

Ensure that the commit in the version record of the selected branch matches the commit name of the final version you have in mind.



4. Choose the branch named after your ID

Ensure that the source of the selected branch is the class folder (courses/software-studio/2024-spring/submission-exercise) and your student ID (not master or someone else's student ID).


- submission-exercise-2
- Project information
- Repository
- Issues 0
- Merge requests 0
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Packages & Registries
- Analytics
- Wiki
- Snippets
- Settings


Chen Yu-Hsuan > submission-exercise-2 > Merge requests > New

### New merge request

**Source branch**

yhch/submission-exercise-2 master

 **Initial commit**  
Chen Yu-Hsuan authored 3 days ago

21779350 

Compare branches and continue

**Target branch**

courses/software-studio/2024-spring/s... master

Select target branch

00000000

No matching results

5. Compare branches

You can also confirm or change whether the selected branch is correct in this section. For instance, in the picture, the course branch selected is master, which is incorrect. The correct one should be `courses/software-studio/2024-spring/submission-exercise: {your student ID}`.

GitLab Menu

Chen Yu-Hsuan > submission-exercise-2 > Merge requests > New

### New merge request

From `yhch/submission-exercise-2:master` into `courses/software-studio/2024-spring/submission-exercise:master` [Change branches](#)

**6. Set title to "{ID} Submission"**

**Title**

Start the title with `Draft:` to prevent a merge request draft from merging before it's ready.  
Add [description templates](#) to help your contributors to communicate effectively!

**Description**

**Write** Preview

Describe the goal of the changes and what reviewers should be aware of.

Markdown and quick actions are supported [Attach a file](#)



Description

**Write** Preview



Describe the goal of the changes and what reviewers should be aware of.

Markdown and quick actions are supported

Attach a file

Assignee

Unassigned

[Assign to me](#)

Reviewer

Unassigned

Milestone

Milestone

Labels

Labels

Merge options

Squash commits when merge request is accepted. [?](#)

Contribution

Allow commits from members who can merge to the target branch. [About this feature](#)

Not available for private projects

Create merge request

7. If everything is OK,  
submit your merge request

# Notice - Don't do this

Google 搜尋

好手氣

# Notice - Don't do this



The image shows a Google search interface with the search term 'gitlab' entered. The search results are displayed below the search bar, showing several entries related to GitLab. The first result is for 'GitLab' with a URL 'https://gitlab.com/' and a note that the search results are limited due to the website's robots.txt. The second result is for 'GitLab.com | GitLab' with a URL 'https://about.gitlab.com/gitlab-com/' and a description of the service. The third result is for 'GitLab介紹— Practical guide for git users 0.1 文档' with a URL 'git-tutorial.readthedocs.io/zh/latest/gitlab.html' and a description of the guide. The fourth result is for 'GitHub - gitlabhq/gitlabhq: GitLab CE | Please open new issues in our ...' with a URL 'https://github.com/gitlabhq/gitlabhq' and a description of the repository. The fifth result is for 'Gitlab - 維基百科，自由的百科全書 - Wikipedia' with a URL 'https://zh.wikipedia.org/zh-tw/Gitlab' and a description of the open-source application.

Google

gitlab

全部 圖片 影片 新聞 書籍 更多 設定 工具

約有 8,300,000 項結果 (搜尋時間：0.45 秒)

**GitLab**  
<https://gitlab.com/> ▾ 翻譯這個網頁  
這項網站搜尋結果說明因為網站的 robots.txt 而無法提供瞭解詳情

**GitLab.com | GitLab**  
<https://about.gitlab.com/gitlab-com/> ▾ 翻譯這個網頁  
GitLab.com. unlimited free repositories and collaborators. Sign Up. Free public & private repositories and unlimited collaborators. Runs GitLab Enterprise Edition ...

**GitLab介紹— Practical guide for git users 0.1 文档**  
[git-tutorial.readthedocs.io/zh/latest/gitlab.html](https://git-tutorial.readthedocs.io/zh/latest/gitlab.html) ▾  
GitLab介紹¶. 目前最流行的線上Git專案管理系統可以說是非GitHub 莫屬，對於一般OpenSource的專案選擇使用GitHub做為線上Git專案管理系統即可，也免收任何 ...

**GitHub - gitlabhq/gitlabhq: GitLab CE | Please open new issues in our ...**  
<https://github.com/gitlabhq/gitlabhq> ▾ 翻譯這個網頁  
README.md. GitLab. Build status CE coverage report Code Climate Core Infrastructure Initiative Best Practices. Canonical source. The canonical source of ...

**Gitlab - 維基百科，自由的百科全書 - Wikipedia**  
<https://zh.wikipedia.org/zh-tw/Gitlab> ▾  
GitLab是一個利用Ruby on Rails開發的開源應用程式，實現一個自代管的Git專案倉庫，可通過Web介面進行存取公開的或者私人專案。它擁有與GitHub類似的功能， ...

# Notice - Don't do this



## GitLab.com

GitLab.com offers free unlimited (private) repositories and unlimited collaborators.

- [Explore projects on GitLab.com](#) (no login needed)
- [More information about GitLab.com](#)
- [GitLab.com Support Forum](#)

By signing up for and by signing in to this service you accept our:

- [Privacy policy](#)
- [GitLab.com Terms.](#)

<b>Sign in</b>	Register
Username or email	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	<a href="#">Forgot your password?</a>
<input type="button" value="Sign in"/>	

Didn't receive a confirmation email? [Request a new one.](#)

Sign in with    

# The GitLab setup in our laboratory is accessible through the link.

Shan-Hung Wu   Description   Announcement   Curriculum ▾   Resources

## Resources

Here are some course materials and resources related to this course. For code and its details (such as assigned reading, project links, quiz, etc.) please refer to the GitLab. For online forum please refer to the iLMS system.



Here!!!!

# Outline

- General Rule
- Introduction to Git
  - Version control
  - Git Basics
  - Try Git!
  - Remote Repositories
- How to Submit Your Code to Gitlab
- **Tools & References**

# Tools

- Git GUI
  - GitKraken
- Editor / IDE
  - Visual Studio Code
  - Atom
  - Sublime Text
  - Brackets
  - Notepad++
  - Webstorm

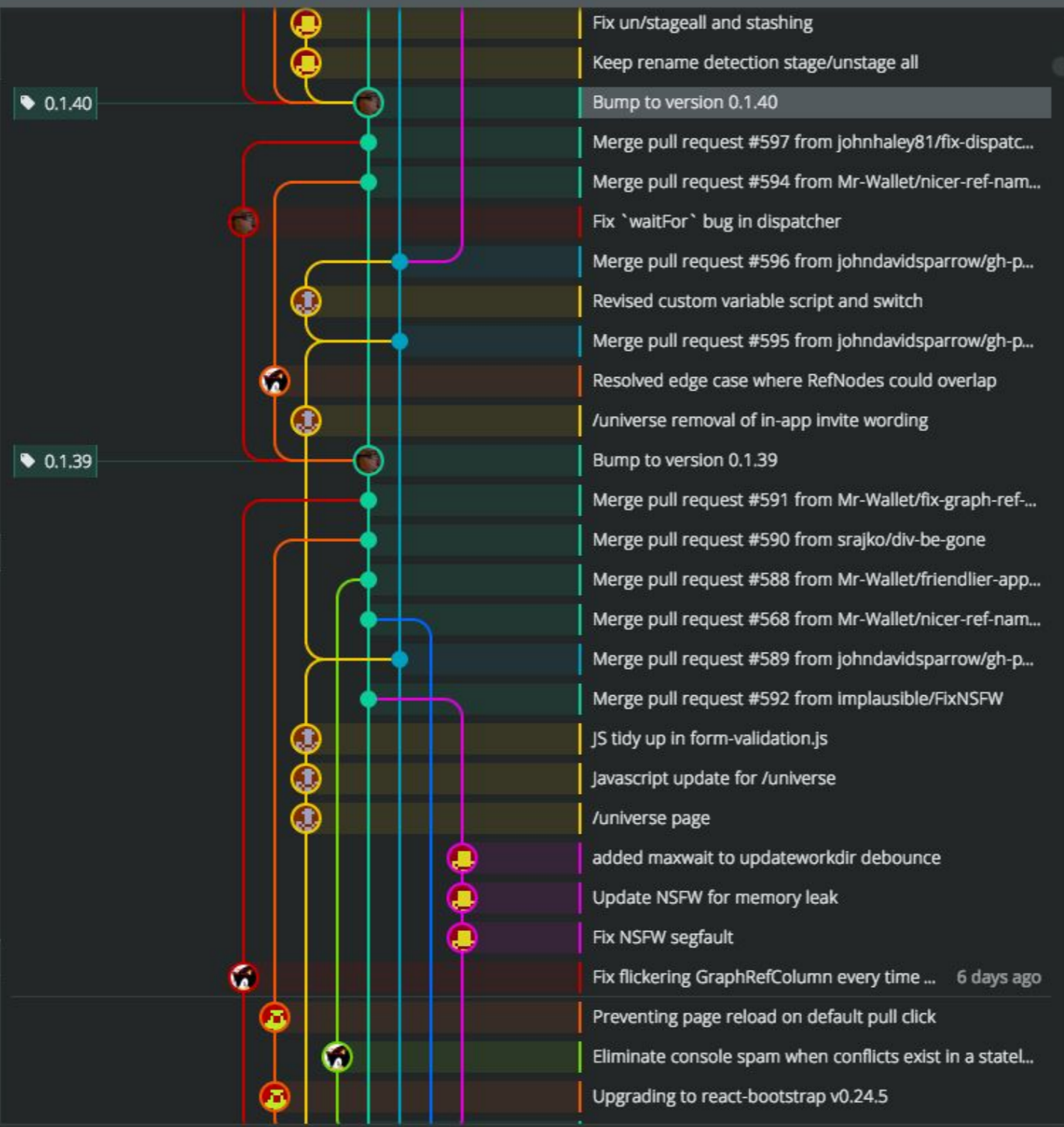


axosoft

# GitKraken



- Viewing 112/151 Show All
- LOCAL (7/11)
- fancier-refbar-changes
  - fancy-responsive-refbar-it... 42 ↗ 99+
  - graph-color-test
  - hopscotch 24 ↗ 99+
  - init-repo-gitignore-typeahead
  - invite-system 6 ↗ 99+
  - jars-view-file-history
  - master 5 ↘
  - remote-panel-redesign 15 ↗ 13+
  - settings-theme-styling
  - view-file-history 24 ↗ 99+
- REMOTE (6/41)
- Jeff-Schinella (0/1)
  - Jordan-Wallet (0/7)
  - Justin-GK (0/1)
  - Ken-Price (0/2)
  - Kyle-Smith (2/8)
  - Max-Korp (0/2)
  - Sjegan-Rajko (0/8)
  - ayresa (0/3)
  - cbargren (0/5)
  - origin (4/4)
- TAGS (99/99)



Commit: cca151e6b9e32c3f9209c25131706740050

Parent: 8efe30a11761983173f844900fa5ec5c6be2

Author: John Haley <johnh@axosoft.com>

Author Date: September 30th 2015, 2:54 pm

Bump to version 0.1.40

0 added 0 deleted 2 modified

npm-shrinkwrap.json

```
@@ -1,6 +1,6 @@
1 1 {
2 2   "name": "gitkraken",
-3  -3   "version": "0.1.39",
+3  +3   "version": "0.1.40",
4  4   "dependencies": {
5  5     "atom-keymap": {
6  6       "version": "5.1.11",
```

package.json

```
@@ -1,7 +1,7 @@
1 1 {
2 2   "name": "gitkraken",
3 3   "productName": "GitKraken",
-4  -4   "version": "0.1.39",
+4  +4   "version": "0.1.40",
5  5   "description": "An intuitive git cli
6  6   "main": "./src/appBootstrap/main.js"
7  7
```



VS Code



EXPLORE

WORKING FILES

03.jpg img

TBL-STYLES

css

img

js

- hoverIntent.js
- jquery.dropdown.js
- jquery.more.js
- jquery.more.min.js
- jquery.plugin.js
- jquery.plugin.min.js
- mapper.js
- maputil.js
- navigation.js**
- smoothscroll.js
- tabs.js

navigation.js js

```
1 var scriptbase = _spPageContextInfo.webServerRelativeUrl + "/_layouts/15/";
2
3 $(document).ready(function () {
4     $.getScript(scriptbase + "SP.Runtime.js", function () {
5         $.getScript(scriptbase + "SP.js", function () {
6             $.getScript(scriptbase + "SP.Taxonomy.js", function () {
7                 context = SP.ClientContext.get_current();
8                 //Call your code here.
9                 console.log("Navigation - ready to rock.");
10
11                 // Get default termstore
12
13                 session = SP.Taxonomy.TaxonomySession.getTaxonomySession(context);
14                 termStore = session.getDefaultSiteCollectionTermStore();
15                 context.load(session);
16                 context.load(termStore);
17                 context.executeQueryAsync(
18                     function () {
19                         console.log('Got default term store');
20                     },
21                     function(sender, args) {
22                         console.log('Could not get default term store. ' + args.get_message());
23                     }
24                 );
25
26
27             });
28         });
29     });
30 });
31
32 var topnavbar;
33
34 topnavbar += '<div class="tbl-site-navigation">';
35 topnavbar += '<ul class="dropdown">';
36 topnavbar += '<li class=""><a href="#">The Brand Code - a</a></li>';
37 topnavbar += '<li class="dropdown1">';
38 topnavbar += '<ul class="sub_menu" style="visibility: hidden;">';
39 topnavbar += '<li class="large">';
40 topnavbar += '<div class="dropdownbox">';
41 topnavbar += '<div class="dropdownbox-title">Welcome to the Brand Code</div>';
42 topnavbar += '<ol>';
43 topnavbar += '<li><a href="">The importance of Brand Building</a></li>';
44 topnavbar += '<li><a href="">Introduction to the Brand Code</a></li>';
45 topnavbar += '<li><a href="">You and the Brand Code</a></li>';
46 topnavbar += '</ol>';
47 topnavbar += '</div>';
```





# ATOM

A hackable text editor  
for the 21st Century

```
atom.coffee  Settings
18
19 # Essential: Atom global for dealing with packages, themes, menus, and the win
20 #
21 # An instance of this class is always available as the `atom` global.
22 module.exports =
23 class Atom extends Model
24   @version: 1 # Increment this when the serialization format changes
25
26   # Load or create the Atom environment in the given mode.
27   #
28   # Returns an Atom instance, fully initialized.
29   @loadOrCreate: (mode) ->
30     startTime = Date.now()
```

# 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>

# Today's exercise

- Install Git command line tool in your computer.
  - Follow appendix “Git Command-line Tool Installation”.
- Try to submit in GitLab.