Practical Web Development
Github

Presented by ARMS（THAILAND）Co., Ltd.

Index

1. Preparing a Github ID

1-1 What are Git and Github

- Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows
- As with most other distributed revision control systems, and unlike most client–server systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server
- See reference at "https://en.wikipedia.org/wiki/Git_(software)"

1. Preparing a Github ID

1-1 What are Git and Github

•GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

•Unlike Git, which is strictly a command-line tool, GitHub provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project

•See reference at https://en.wikipedia.org/wiki/GitHub

In a nutshell, Git is a distributed version management system, Github is a web service name based on Git.

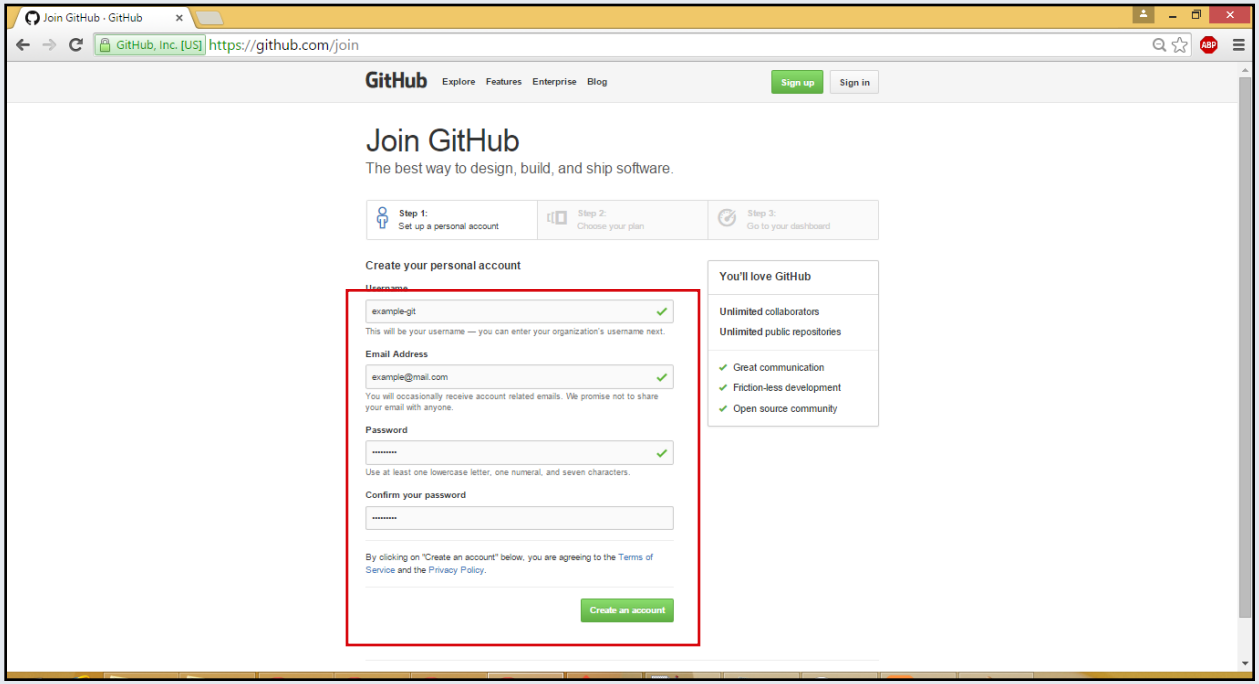
Git is developed by Linux development team.

Github is run by Github company

1. Preparing a Github ID

1-2 Create a Github ID

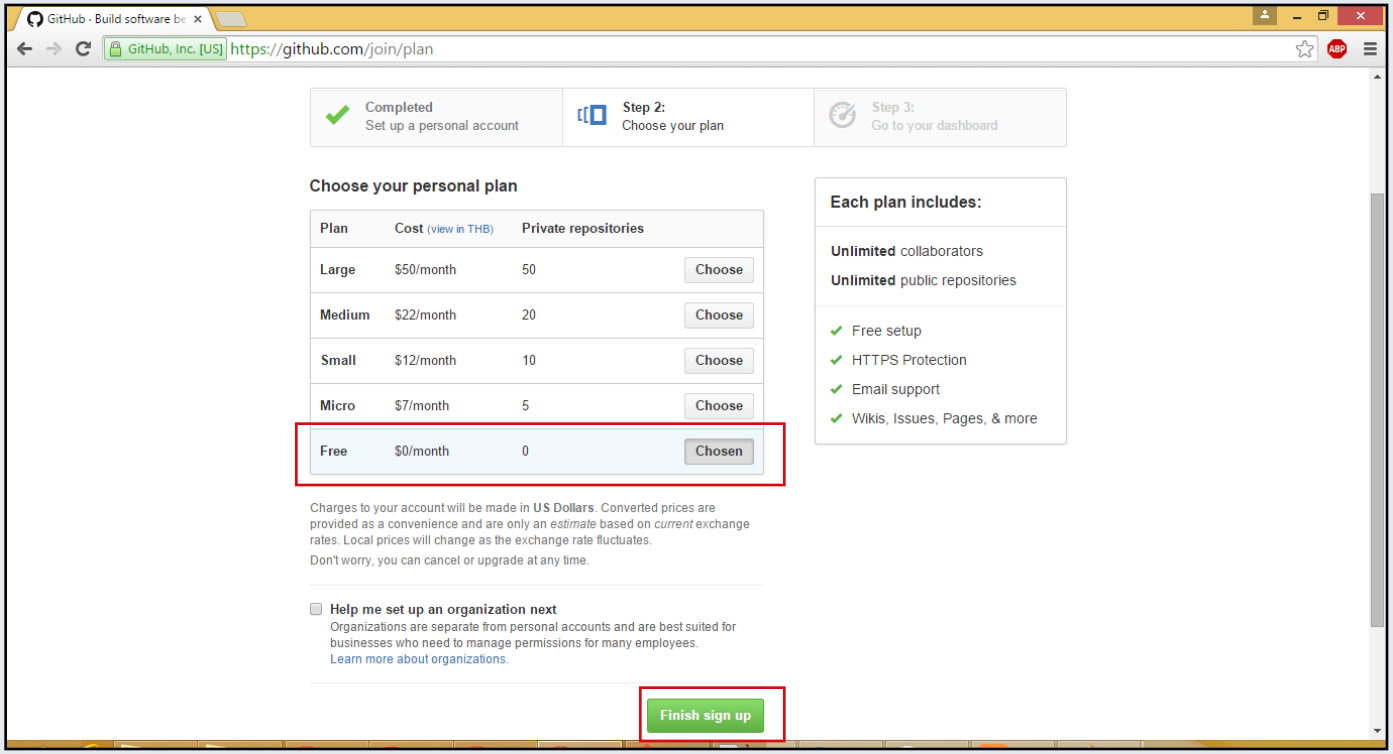
- Open "**https://github.com/join**" to create a new Github ID
- Insert these information and click "Create an account"



1. Preparing a Github ID

1-2 Create a Github ID

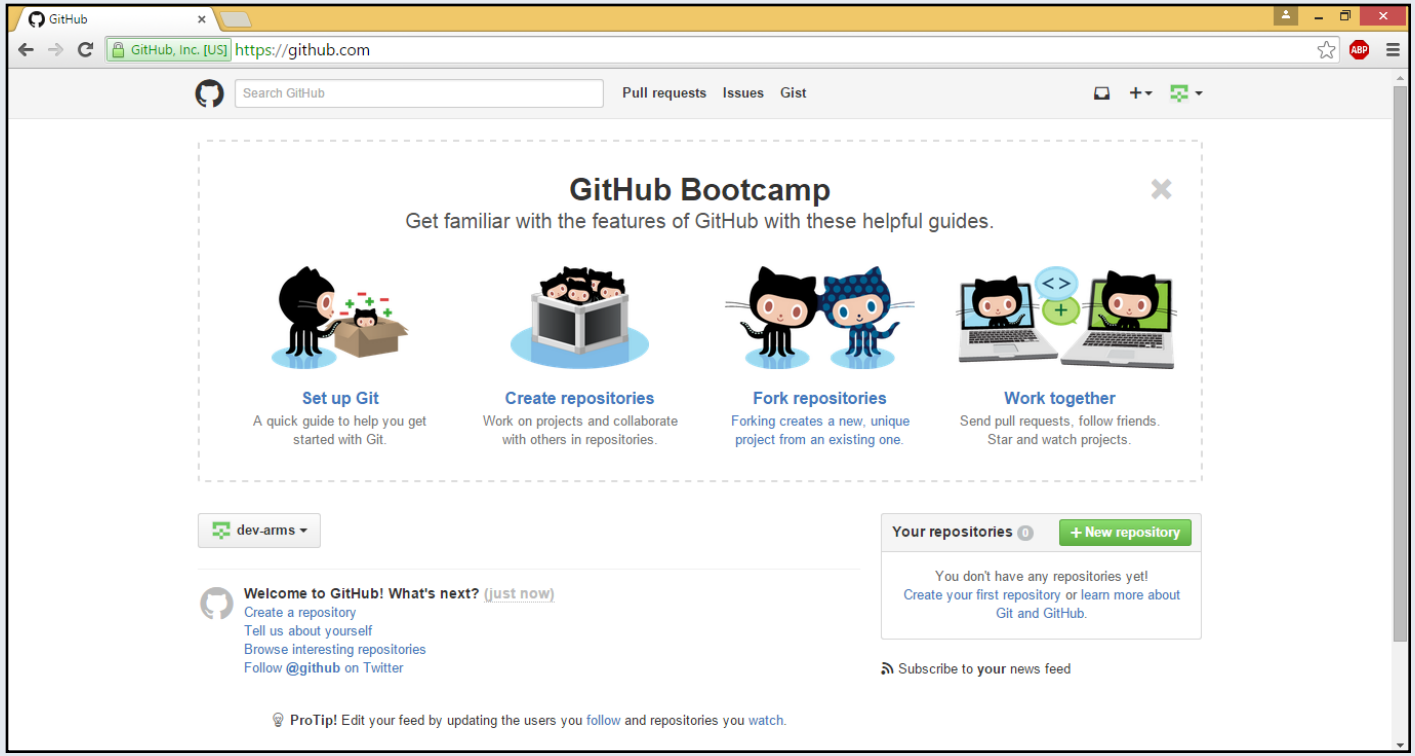
- Choose "Free" plan and click "Finish sign up"



1. Preparing a Github ID

1-2 Create a Github ID

- Now, you got a Github ID

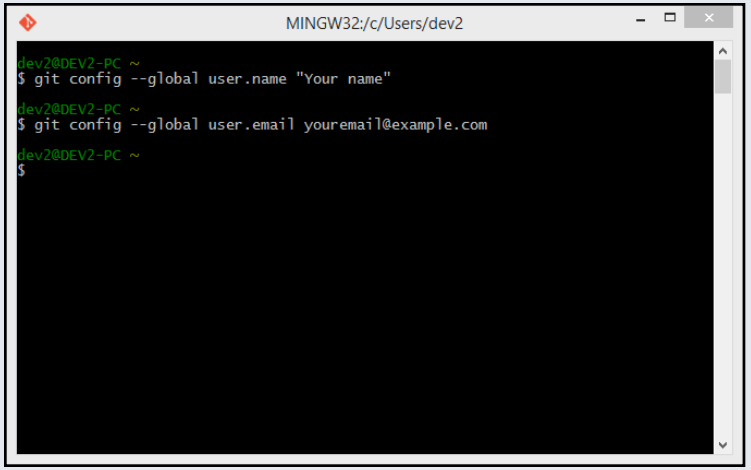


2. Preparing a Git

2-1 Configure

- The first thing you should do when Git installed is to set your user name and e-mail address
- This is important because Git uses this information when you commit
- Open Git Bash (there should be a shortcut on your desktop).

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



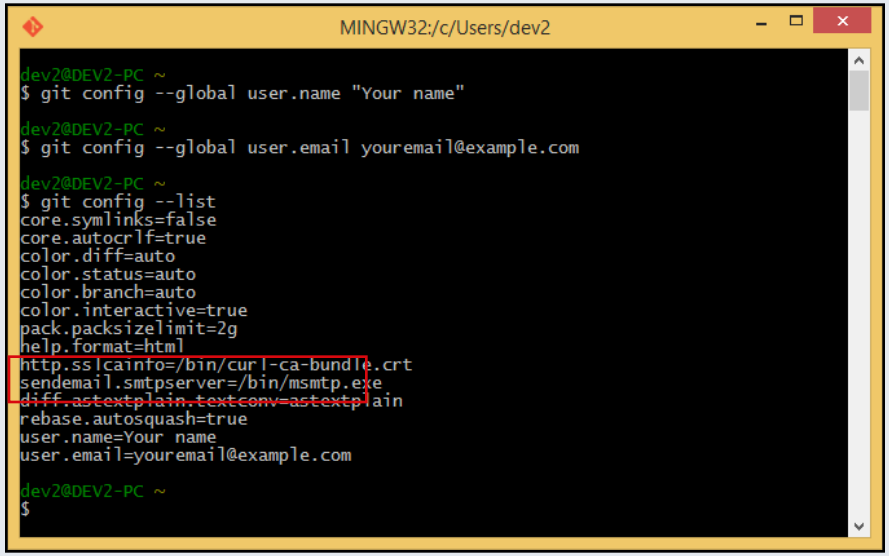
2. Preparing a Git

2-1 Configure

- Check your settings
- Use the git config --list command to list all the settings

```
git config --list
```

- Make sure that **user.name** and **user.email** have been properly added to the setting



You can check a specific key value like below

```
git config {key}
```

```
git config user.email
```

2. Preparing a Git

2-1 Configure

•If you set a wrong user.name or user.email, use the following commands to edit the wrong setting.

`git config --global --edit` or `vi .gitconfig`

To edit the .gitconfig, you have to use vi editor.
It has a command mode and edit mode,

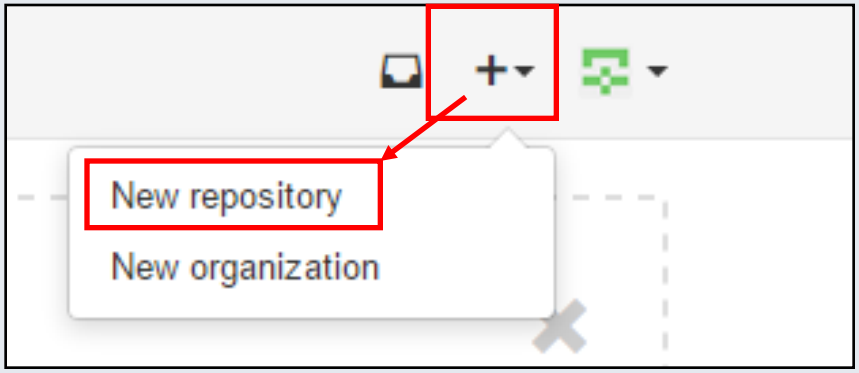
1. To enter into an edit mode, press either “a” “i” “o” key.
2. To finish the edit mode, press “esc” key.
3. Then type : wq (colon : write the change and quit)

If you don’t want to write the change,
:q! (colon : quit without saving)

3. Getting started with Git and GitHub

3-1 Create a new repository on GitHub

•Open Github page, click + icon on the upper-right corner , and then click New repository



3. Getting started with Git and GitHub

3-1 Create a new repository on GitHub

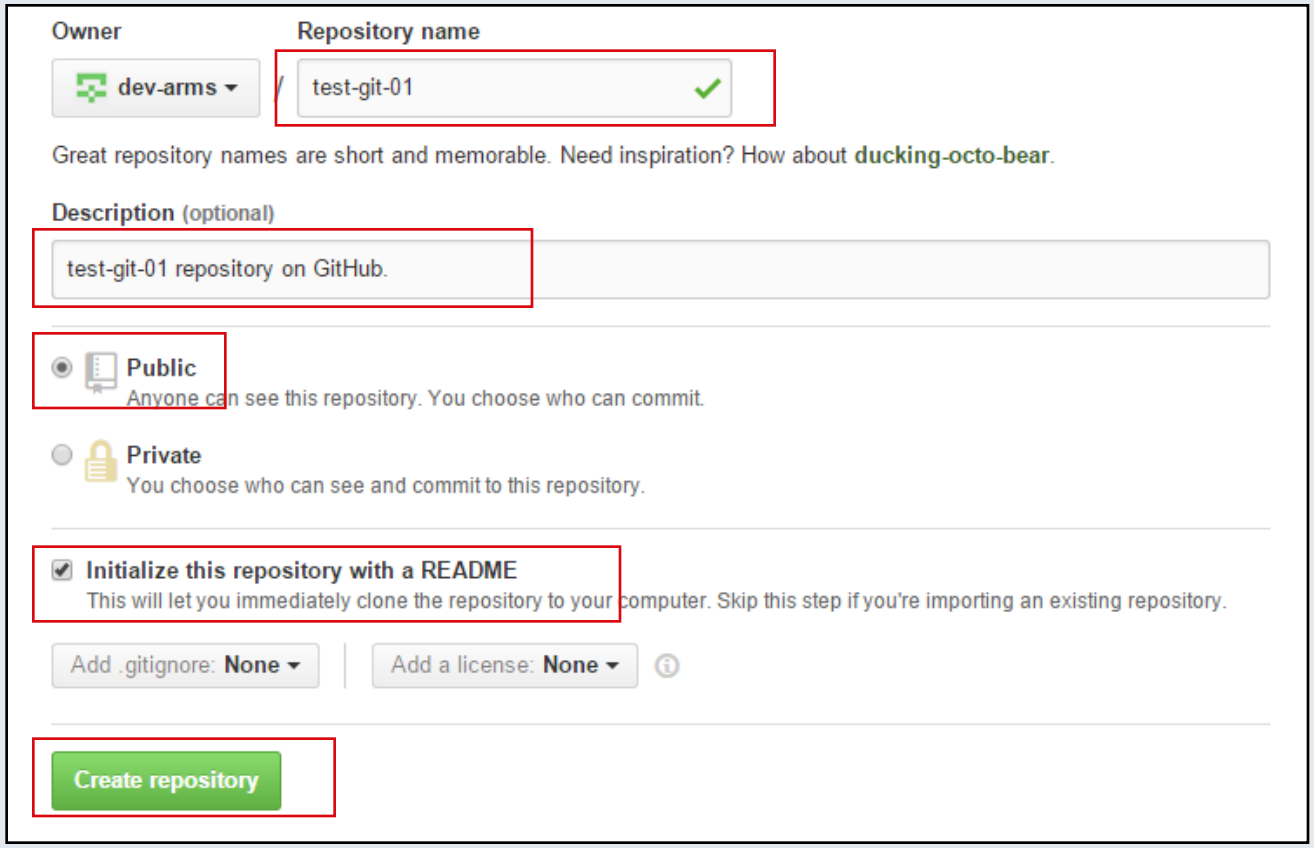
- Create Repository name

  Repository name : "test-git-01"
- Optionally, add a description of your repository.

  Desciption : "test-git-01 repository on GitHub."
- Choose ″**Public**″ repository
  - **Public repository** is a great choice for getting started! They're visible to any user on GitHub, so you can benefit from a collaborative community
  - **Private repository** requires a little more setup. They're only available to you, the repository owner, as well as any collaborators you choose to share with. Private repository is only available for a paid account.
- Select ″**Initialize this repository with a README**″
- Click ″**Create repository**″

3. Getting started with Git and GitHub

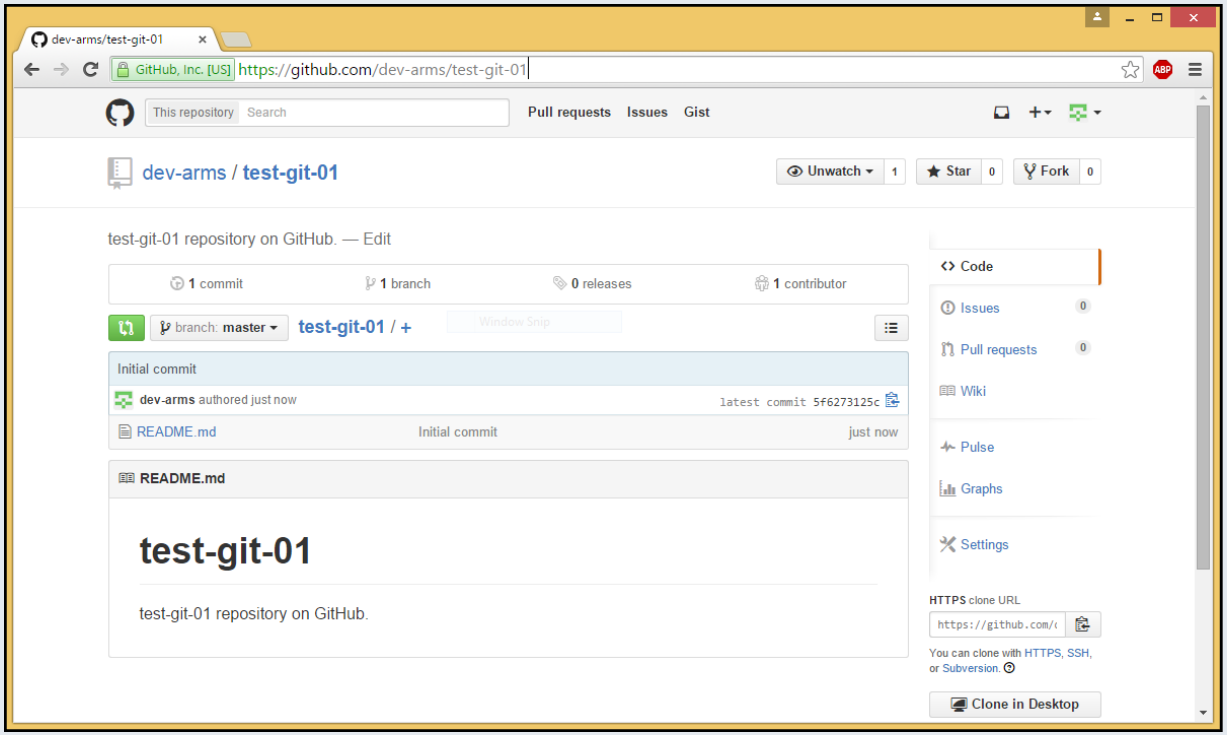
3-1 Create a new repository on GitHub



3. Getting started with Git and GitHub

3-1 Create a new repository on GitHub

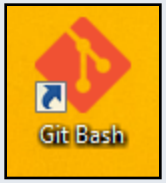
- Now, you can see your repository.

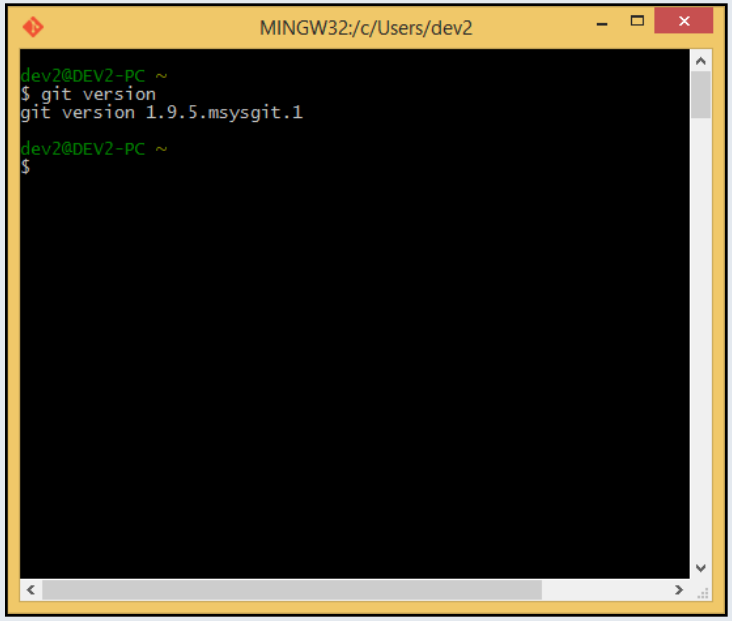


3. Getting started with Git and GitHub

3-2 Get local repository from Github

•You can use Git via Git Bash from desktop shortcut or via Windows Command Prompt

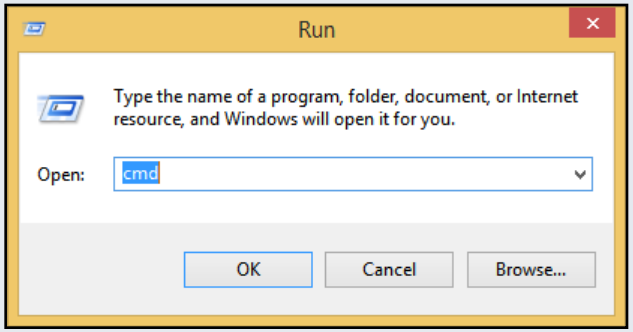


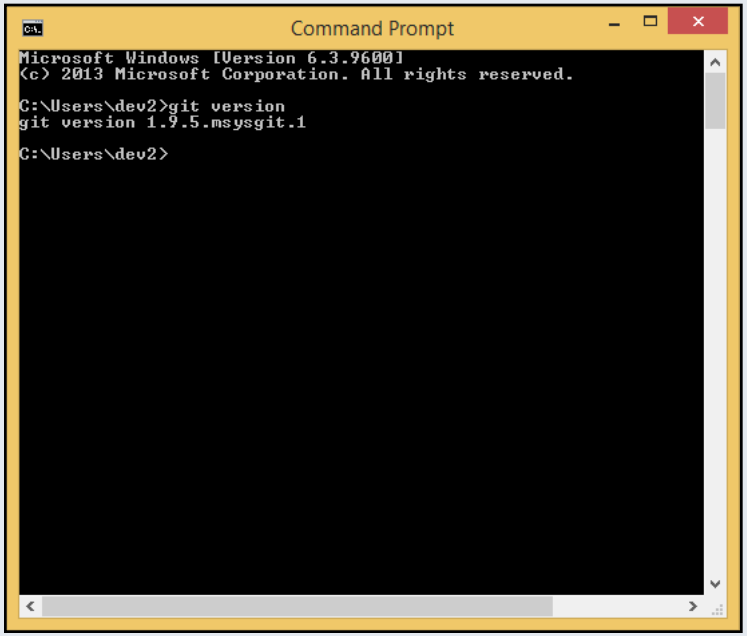


3. Getting started with Git and GitHub

3-2 Get local repository from Github

- You can use git via Windows Command Prompt ("Windows Key + r" and type "cmd")





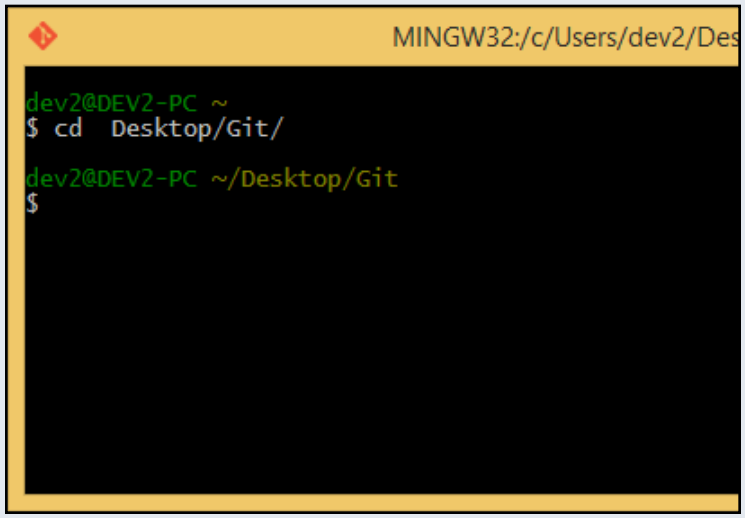
3. Getting started with Git and GitHub

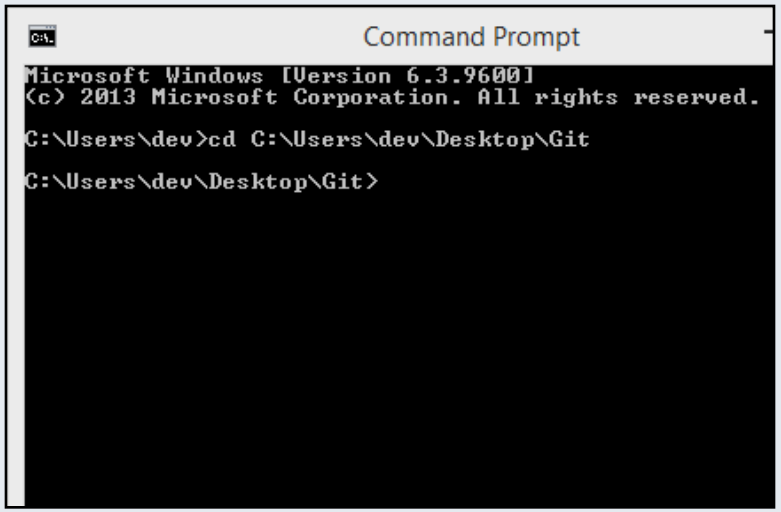
3-2 Get local repository from Github

•Open your git bash or cmd and run "cd [path to directory]" to go to directory that you want to place this repository. For example

via Git bash use	: **cd Desktop/Git**
via cmd use	: **cd C:\Users\dev\Desktop\Git**

*** The above directories are just an example, please create a directory such as Git by using "mkdir" command.**

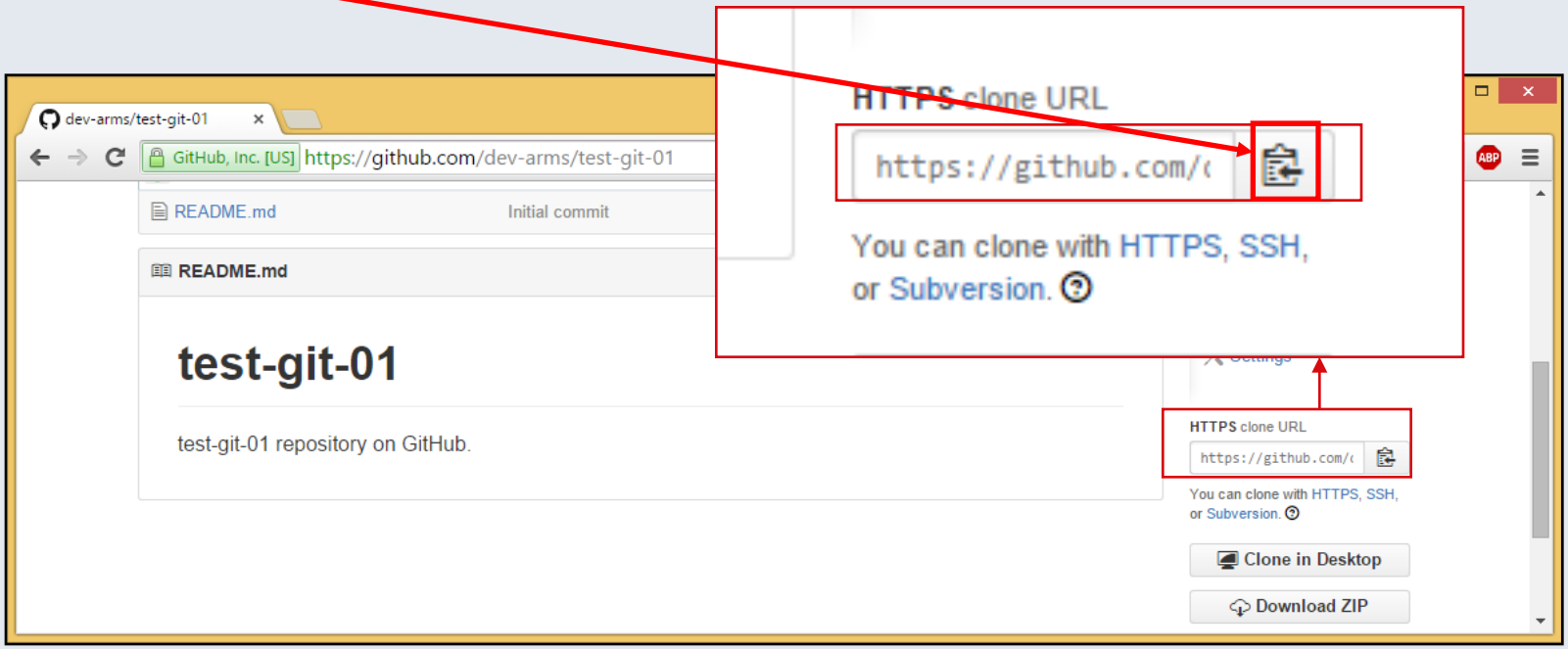




3. Getting started with Git and GitHub

3-2 Get local repository from Github

•Use "git clone [url]" to clone Github repository to local. You can get repository [url] from your Github repository page. Click green "**Clone or download**" button. Copy URL



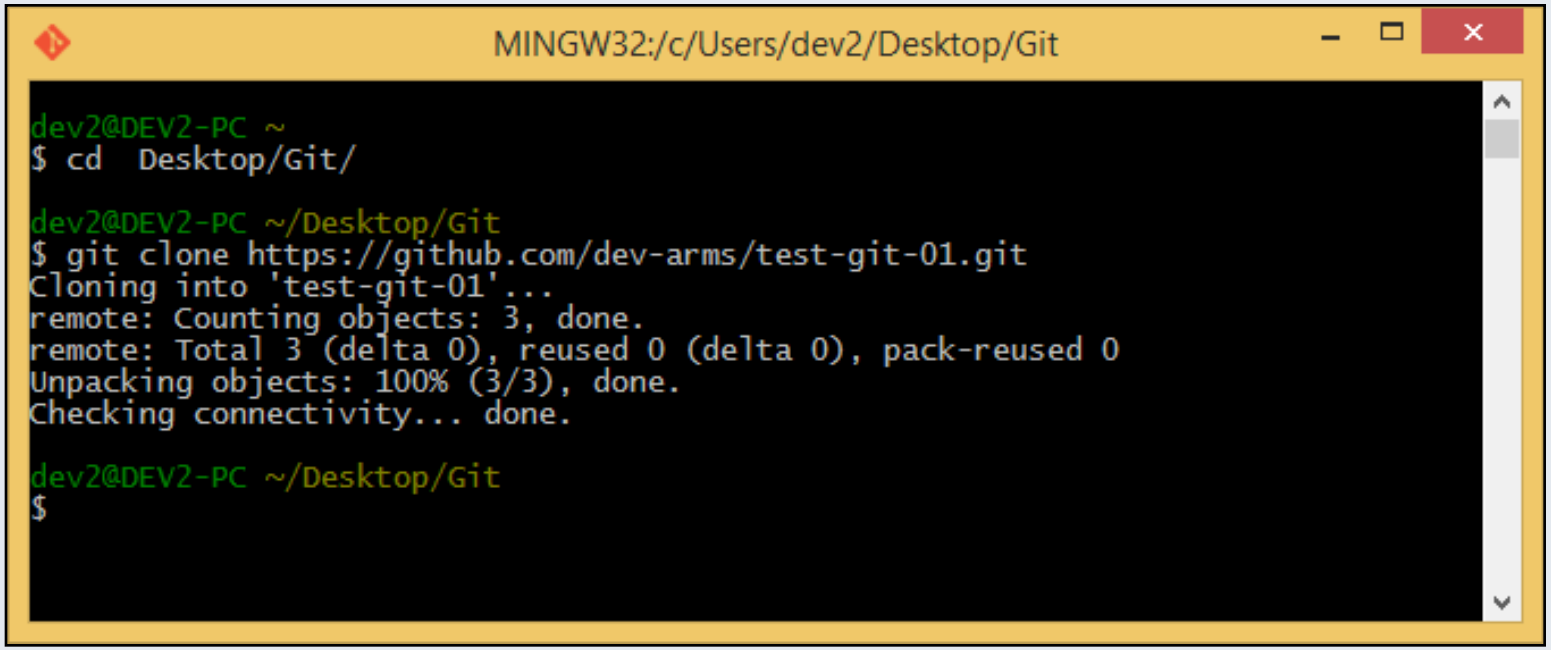
3. Getting started with Git and GitHub

3-2 Get local repository from Github

•Example

* URL from Github repository

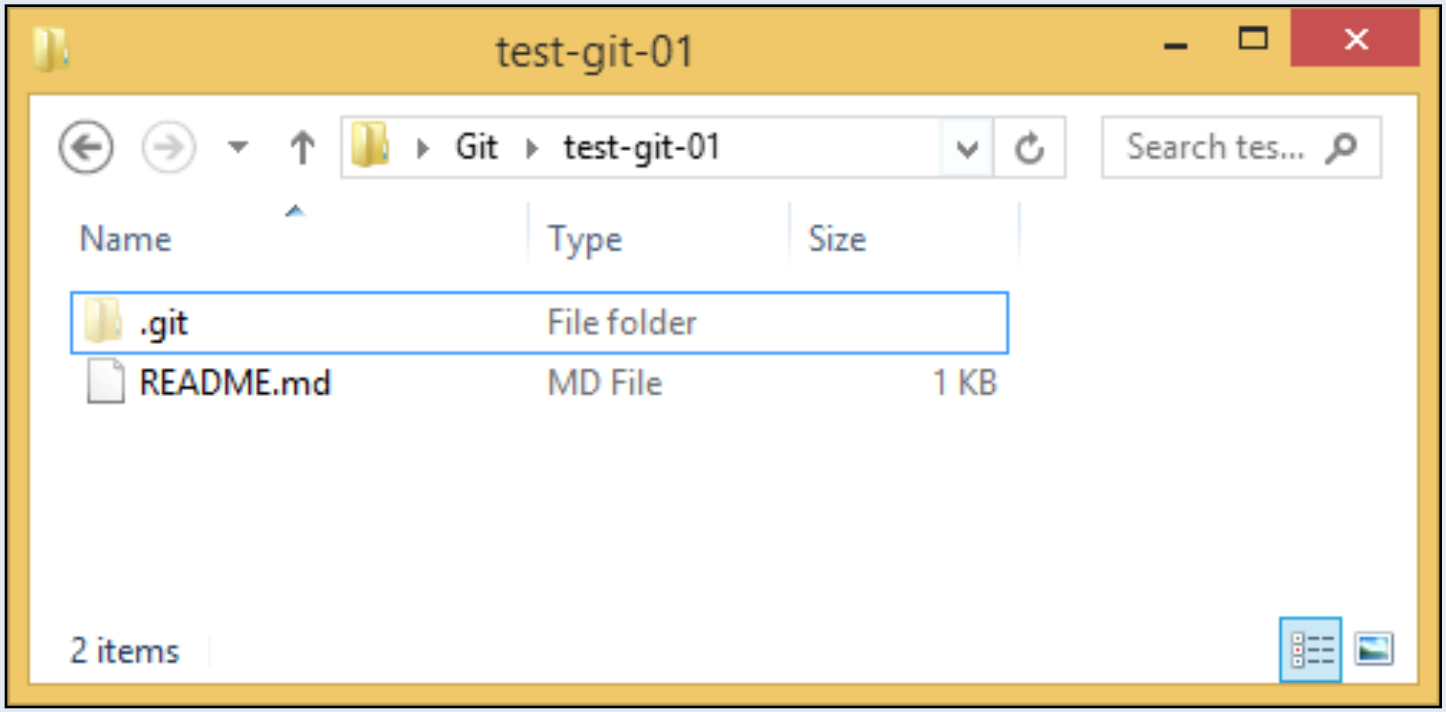
git clone **https://github.com/dev-arms/test-git-01.git**



3. Getting started with Git and GitHub

3-2 Get local repository from Github

•Now, you can see "test-git-01" on your PC



3. Getting started with Git and GitHub

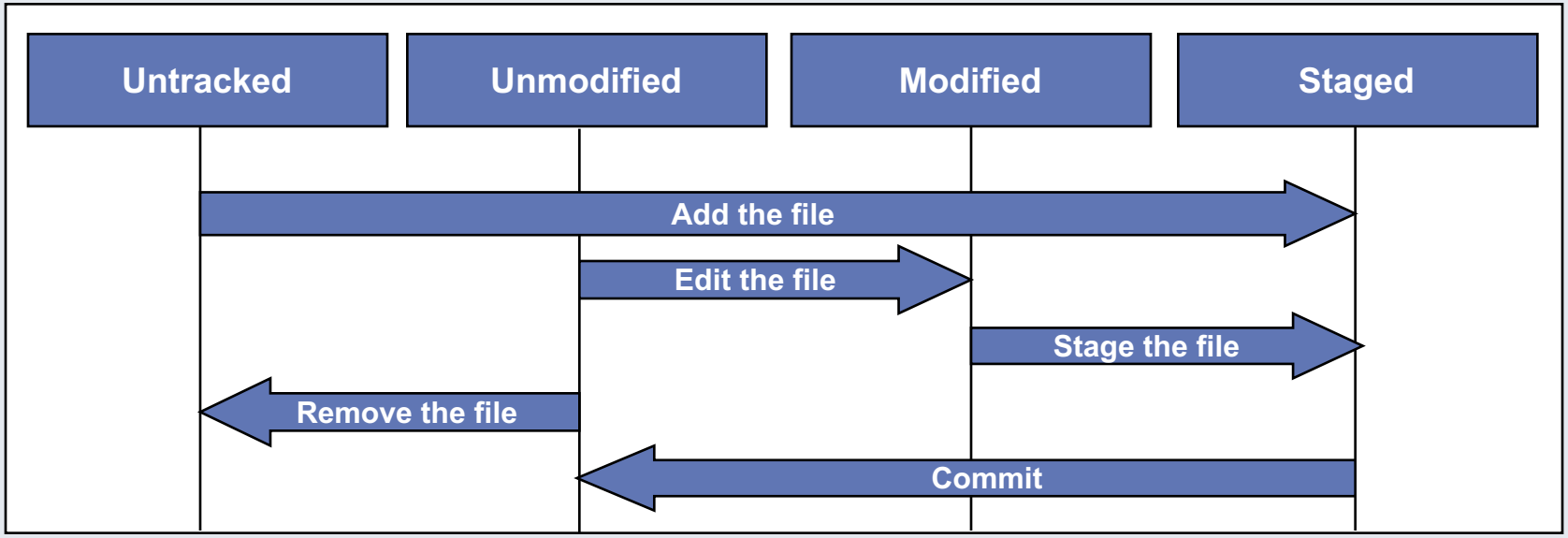
3-3 Recording Changes to the Repository

- Remember that each file in your working directory can be in one of two states: tracked or untracked
- Tracked files are files that were in the last snapshot. They can be unmodified, modified, or staged
- Untracked files are everything else – any files in your working directory that were not in your last snapshot and are not in your staging area. When you first clone a repository, all of your files will be tracked and unmodified because you just checked them out and haven't edited anything

3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

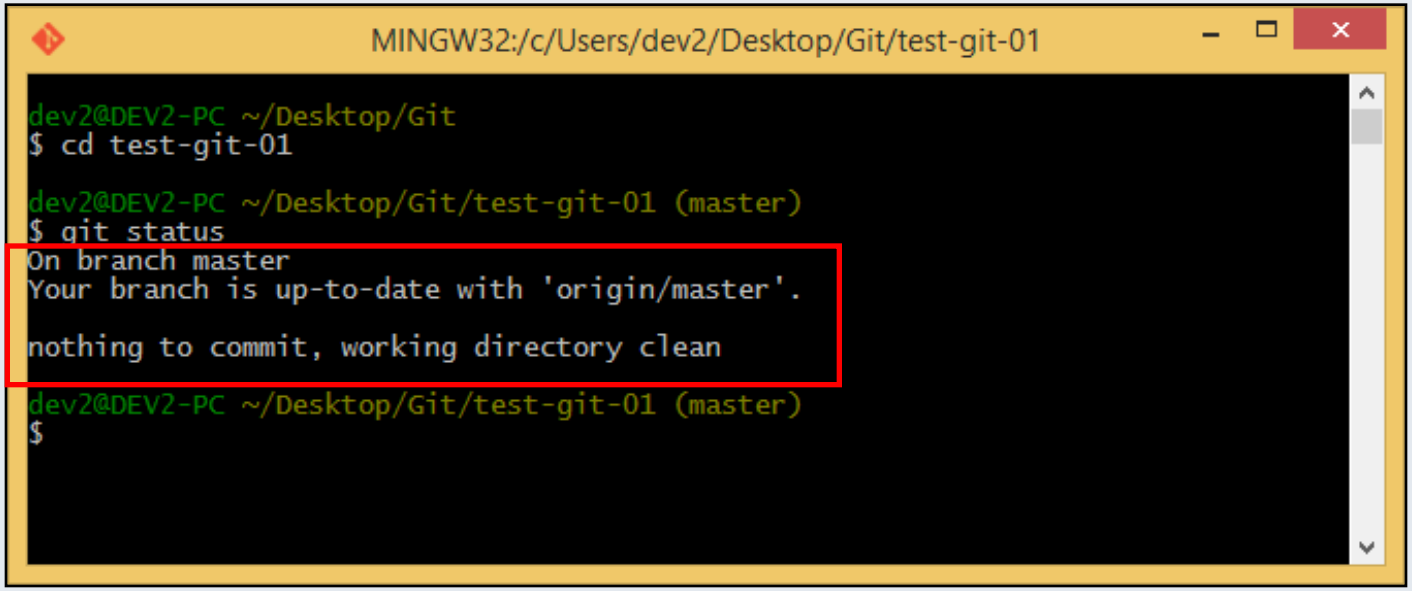
- The lifecycle of your file status



3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

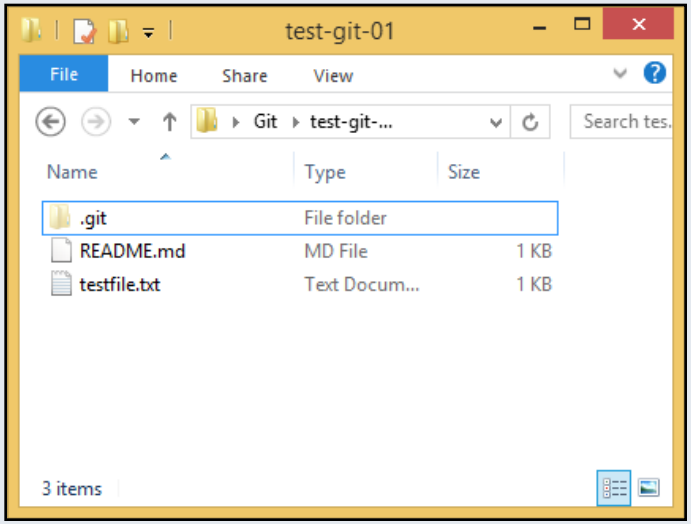
- The main command you use to determine which files are in which state is "**git status**" command
- If you go to local git repository and run this command right after cloning a repository, you will see the screen below.

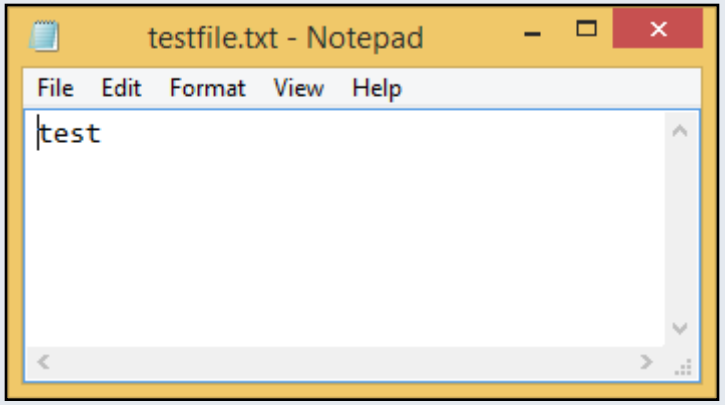


3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

- This means you have a clean working directory. In other words, there are no tracked and modified files.
- Try to create a new file to your project.
- For example, create "**testfile.txt**" file that included some text in your project

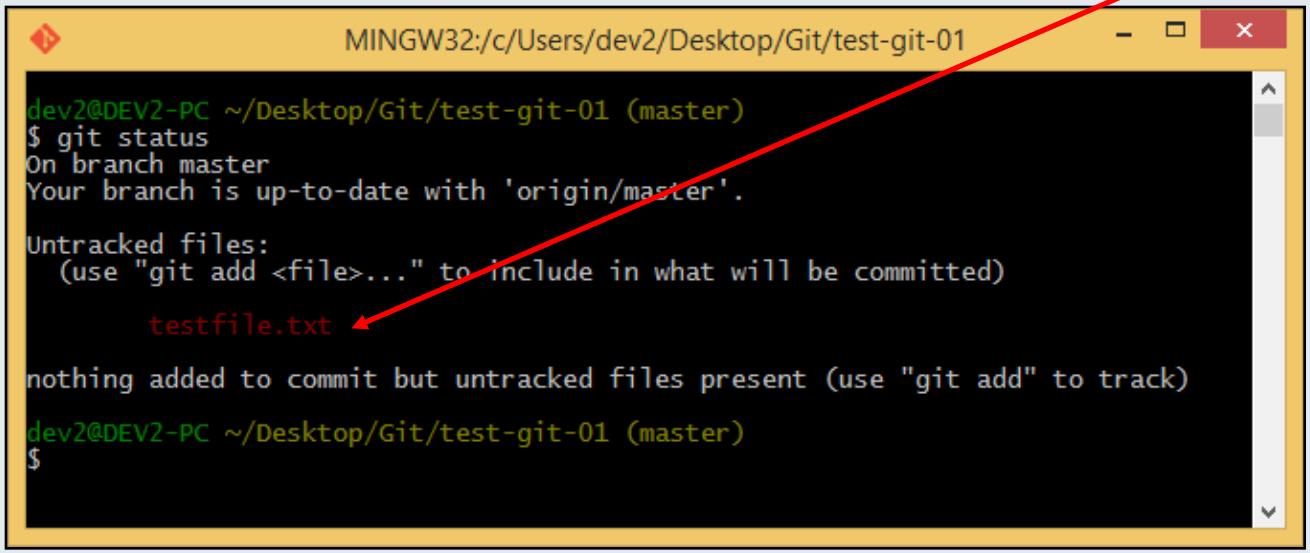




3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

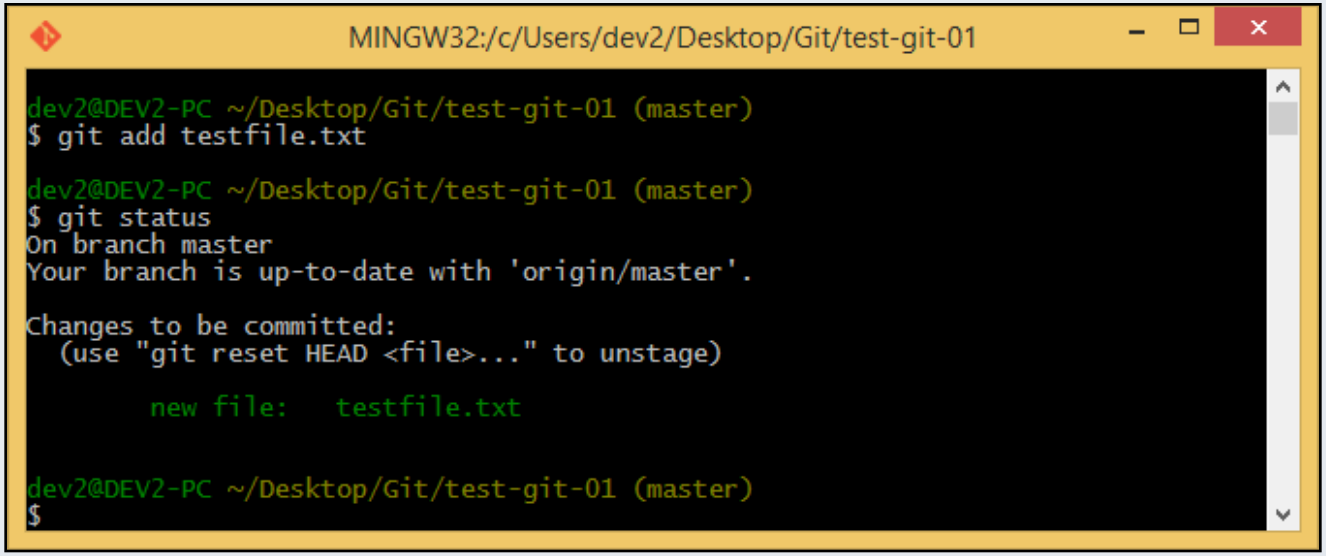
•After creating a new file, and run "**git status**", you will see an untracked file.



3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

•To begin tracking a new file, use "**git add [filename]**" (Or "**git add ***" to track all files) . **git add testfile.txt**" and run "**git status**" again



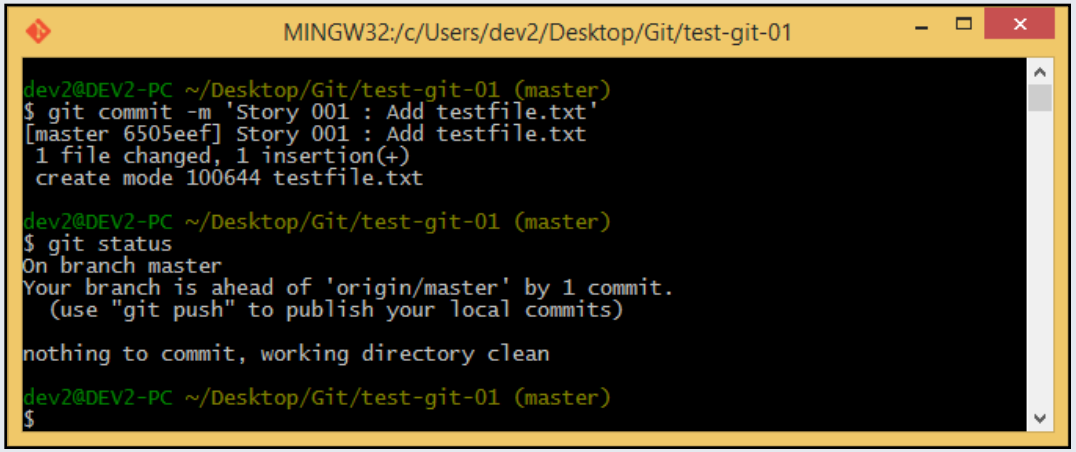
•You see your "testfile.txt" file tracked and staged to be committed.

3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

- Now that your staging area is set up the way you want it, you can commit your changes *(If you want to unstage file because you add a wrong file, you can use "git reset HEAD [filename]")*
- The simplest way to commit is to run "**git commit -m 'commit message'**" and run "git status" again.

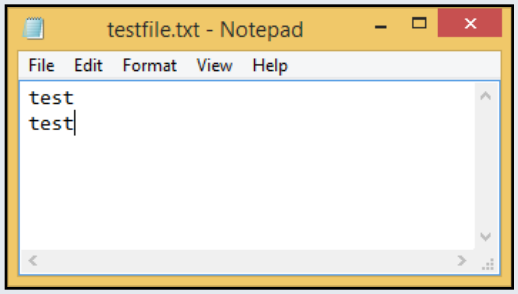
```
git commit -m 'Story 001 : Add testfile.txt'
git status
```

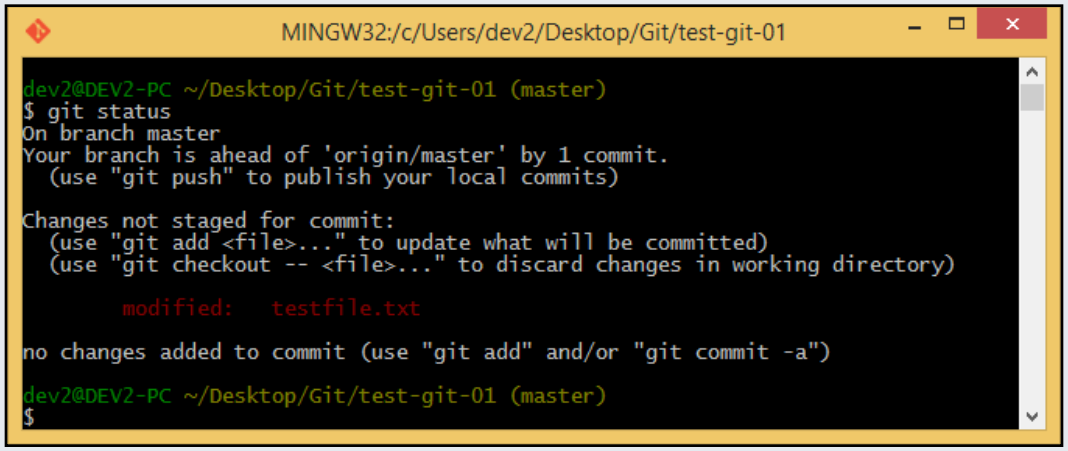


3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

•Try to modify text in "testfile.txt' and run "**git status**"





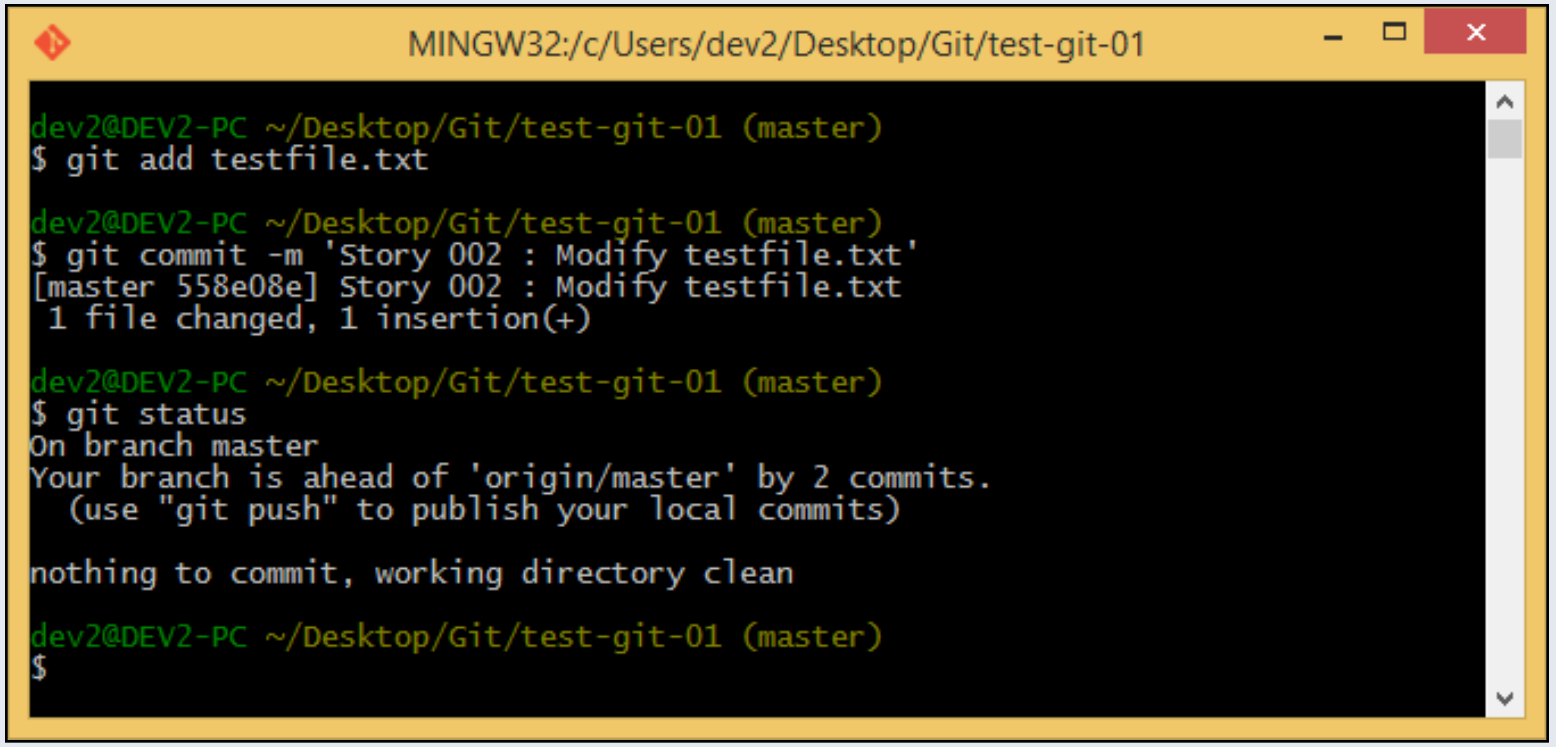
•It's similar to tracking a new file. You must run "git add [filename]" and "git commit -m 'commit message'" again. For example

```
git add testfile.txt
git commit -m 'Story 002 : Modify testfile.txt'
git status
```

3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

- You can see this result

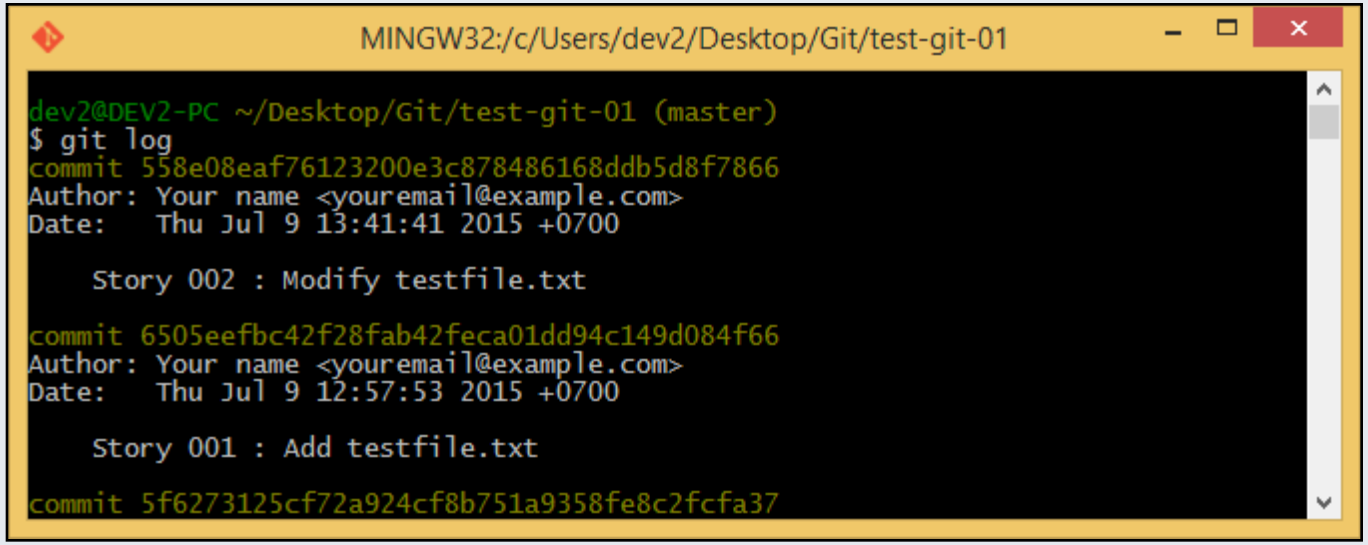


3. Getting started with Git and GitHub

3-3 Recording Changes to the Repository

•Remember that when you create, modify, delete, rename or something that makes a change to a file and you want to track the file, you have to run "git add [filename]" and "git commit 'commit message'" everytime changes are made.

•If you want to see commit history. You can run "**git log**“ to see the commit history.

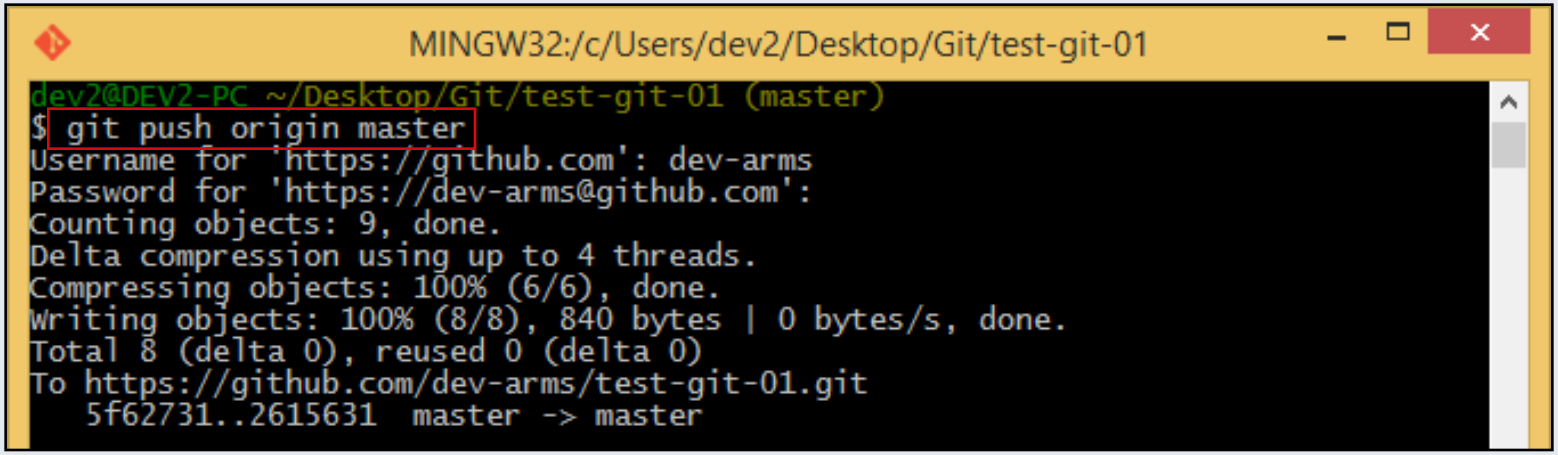


3. Getting started with Git and GitHub

3-4 Pushing to a remote

- Pushing is to put your commits on local to your origin repository (on GitHub)
- Run "git push origin [branch name]" to push to your origin repository
- When you run this command, it will ask username and password. Type your Github's username and password.

git push origin master



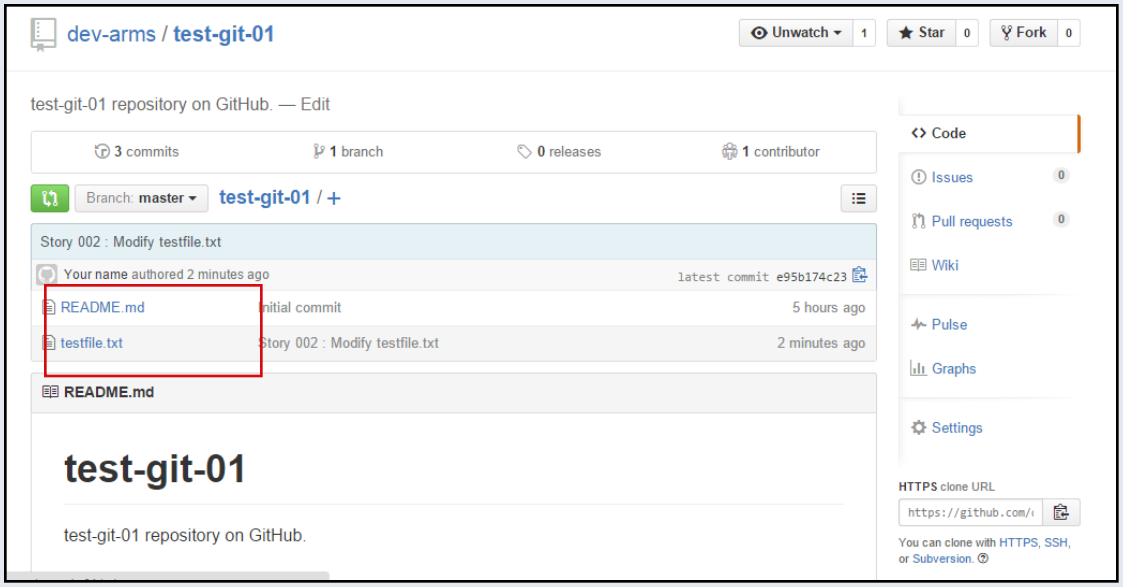
3. Getting started with Git and GitHub

3-4 Pushing to a remote

- Go to your repository on Github website

https://github.com/your_username/your_reponame

- Now you can see new files(testfile.txt) pushed from your local.



3. Getting started with Git and GitHub

3-4 Pushing to a remote

•When you pushed something new to a remote repository on Github and you have collaborators that had cloned your repository before you pushed.

•Those collaborators can run "**git pull**" to merge changes into their local repository, that way they can also have the same code as yours.

•Important point to remember, with Git, you can also manage even Word files and other documents, so you can use Git for even managing your documents.

3. Getting started with Git and GitHub

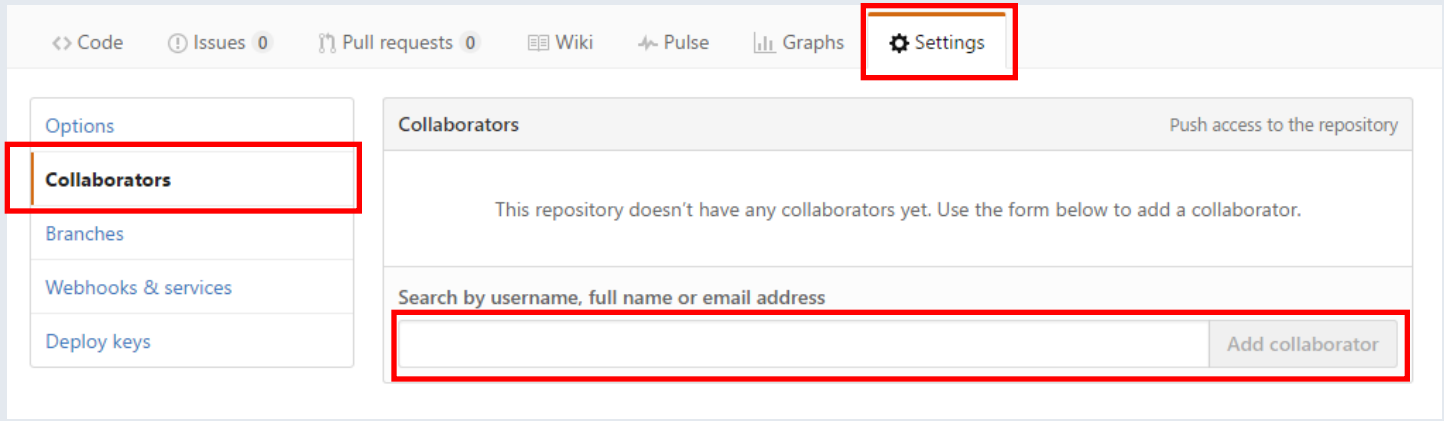
3-5 Team collaboration

- Try team collaboration using “your" repository
- To try team collaboration, you have to add collaborators for your repository.

Settings tab

Collaborators

Search and Add collaborators



3. Getting started with Git and GitHub

3-5 Team collaboration

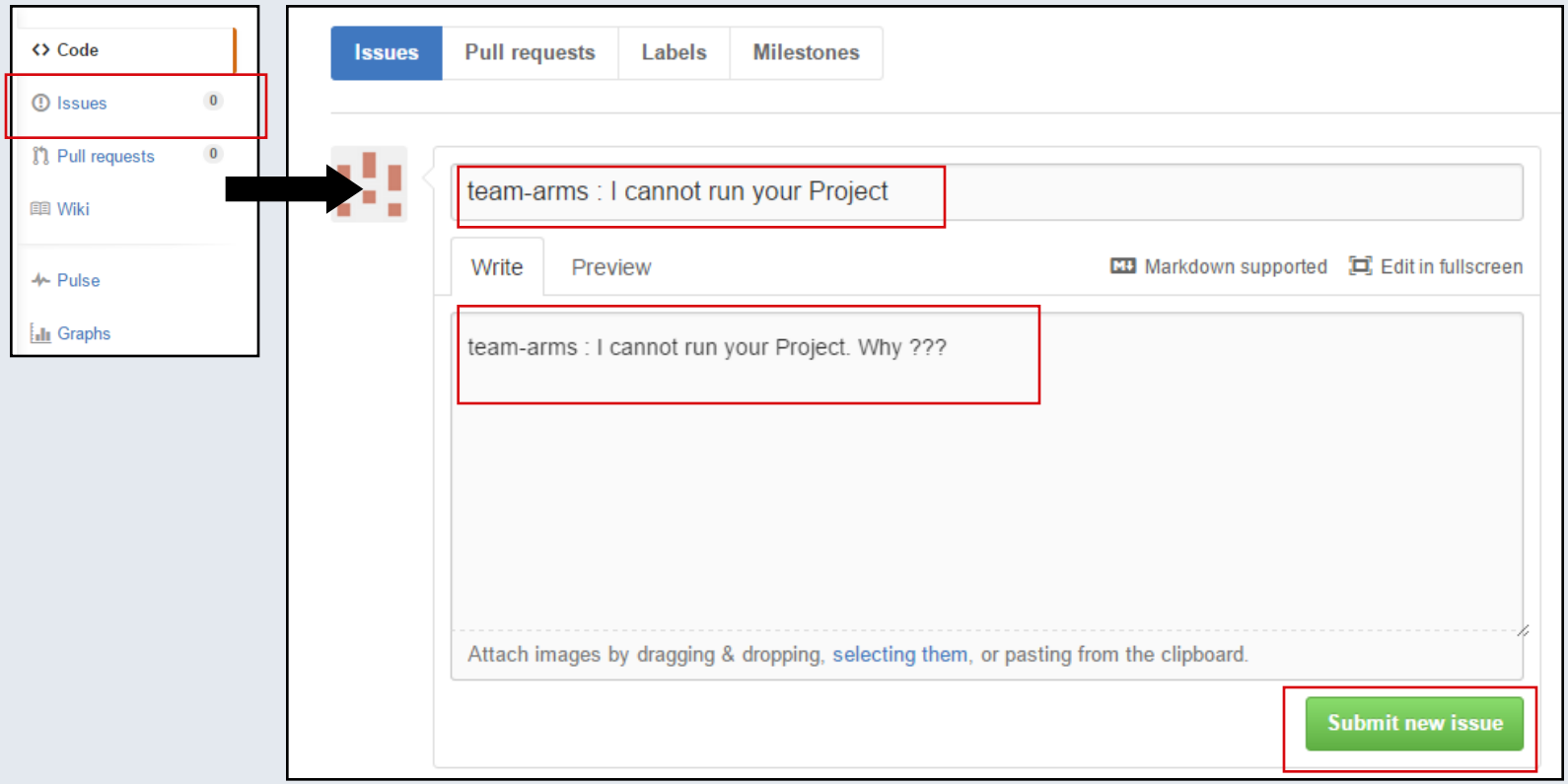
•Try to do something to “your repository". For example, add a new file

•First, Create a new Issue (An Issue is a note on a repository about something that needs attention. It could be a bug, a feature request, a question or lots of other things)

- Click the Issues tab from the sidebar on your "bu-arms-xx" repository
- Click New Issue
- Give your Issue a title and description. For example
  - title : **[your name] : I cannot run your Project**
  - description : **[your name] : I cannot run your Project, Why ???**

3. Getting started with Git and GitHub

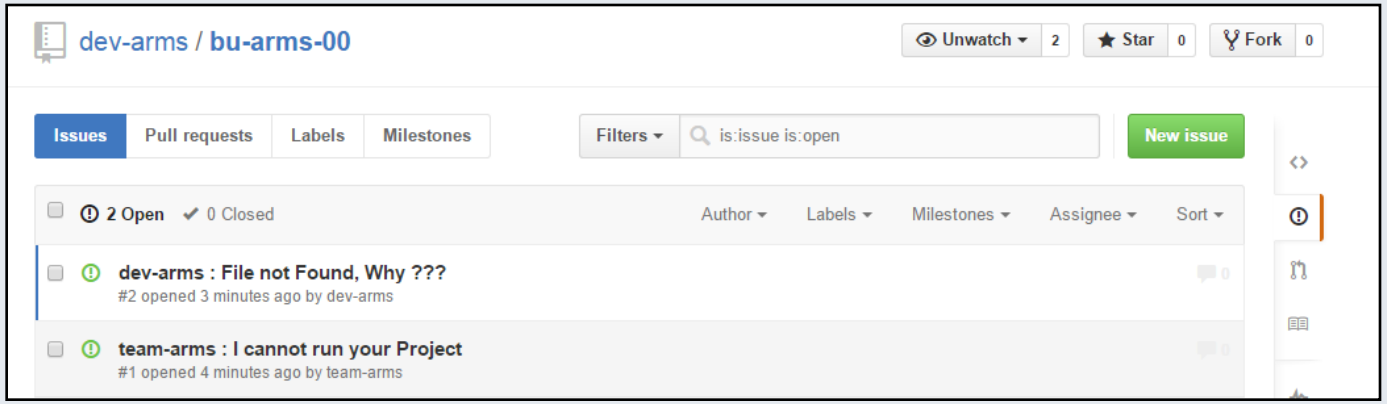
3-5 Team collaboration

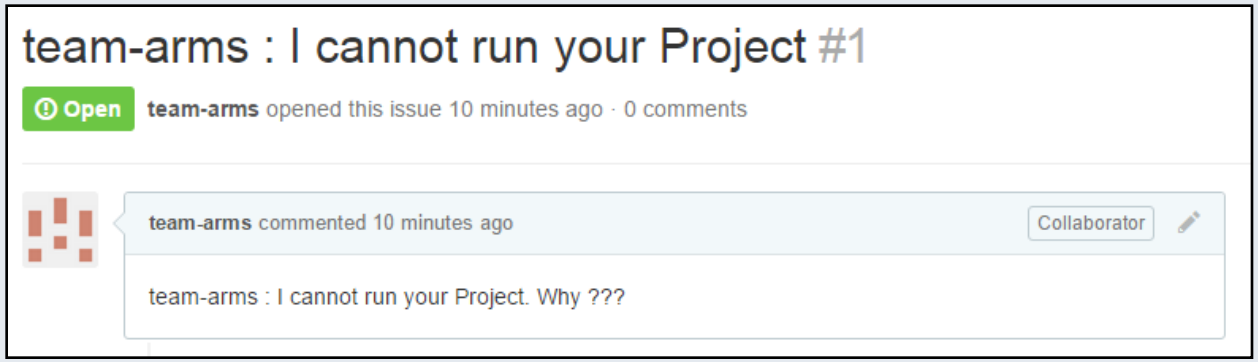


3. Getting started with Git and GitHub

3-5 Team collaboration

- You can see your issue and others' issue

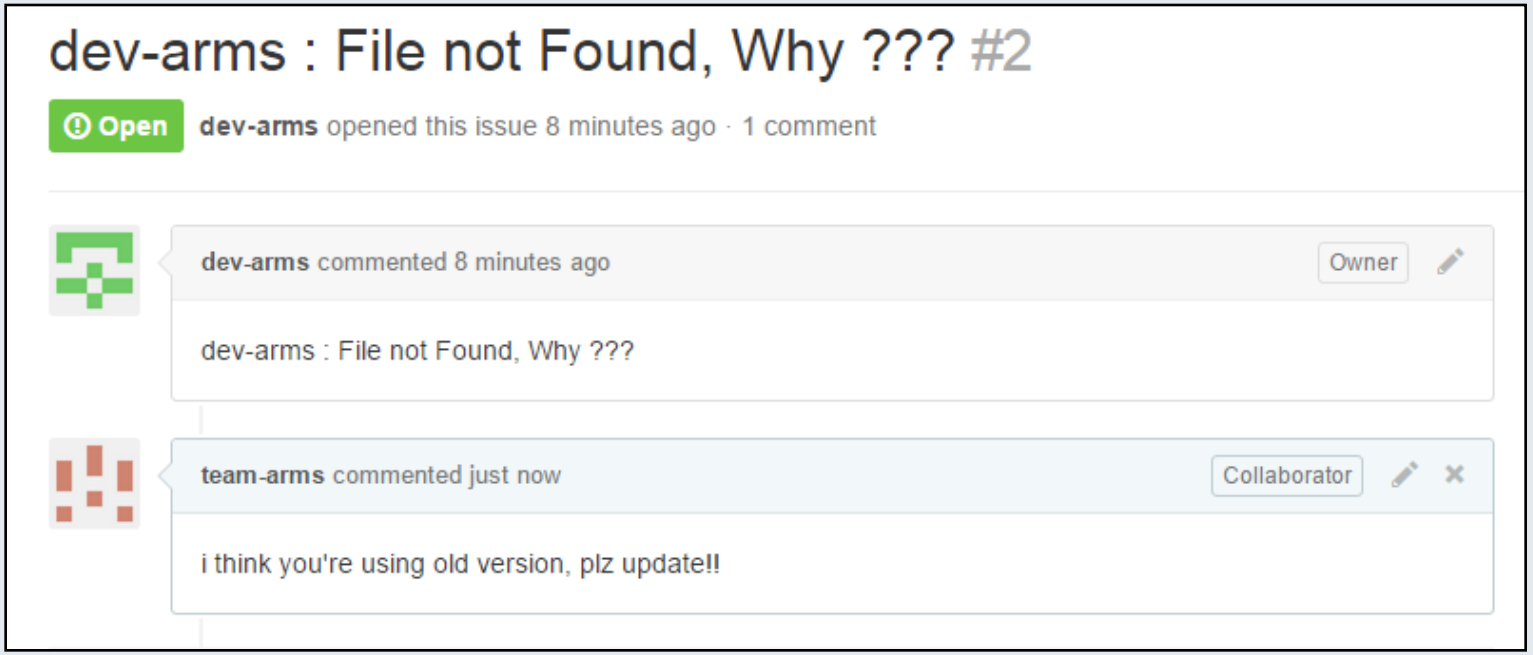




3. Getting started with Git and GitHub

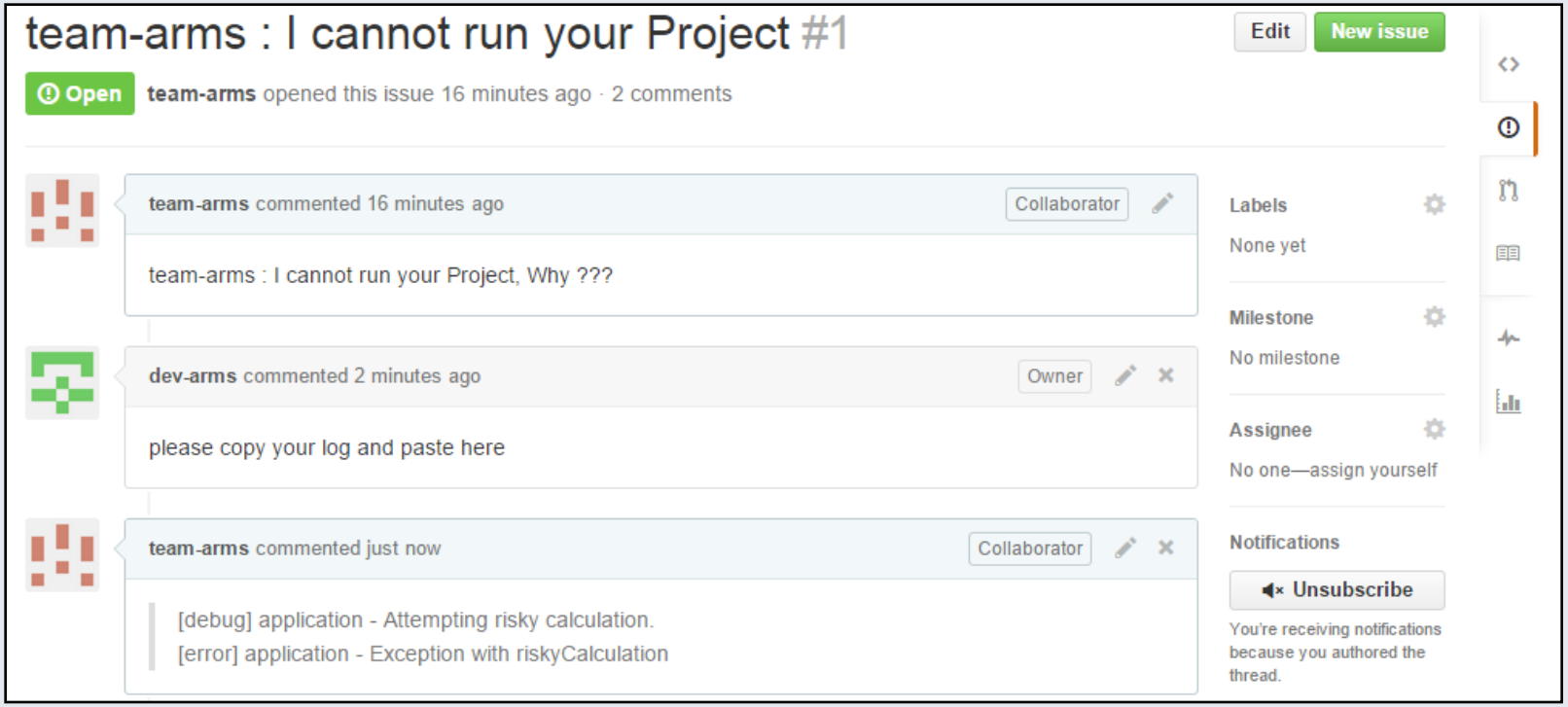
3-5 Team collaboration

•you can share and discuss each issue like below.



3. Getting started with Git and GitHub

3-5 Team collaboration



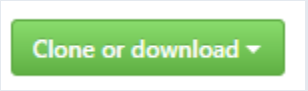
3. Getting started with Git and GitHub

3-5 Team collaboration

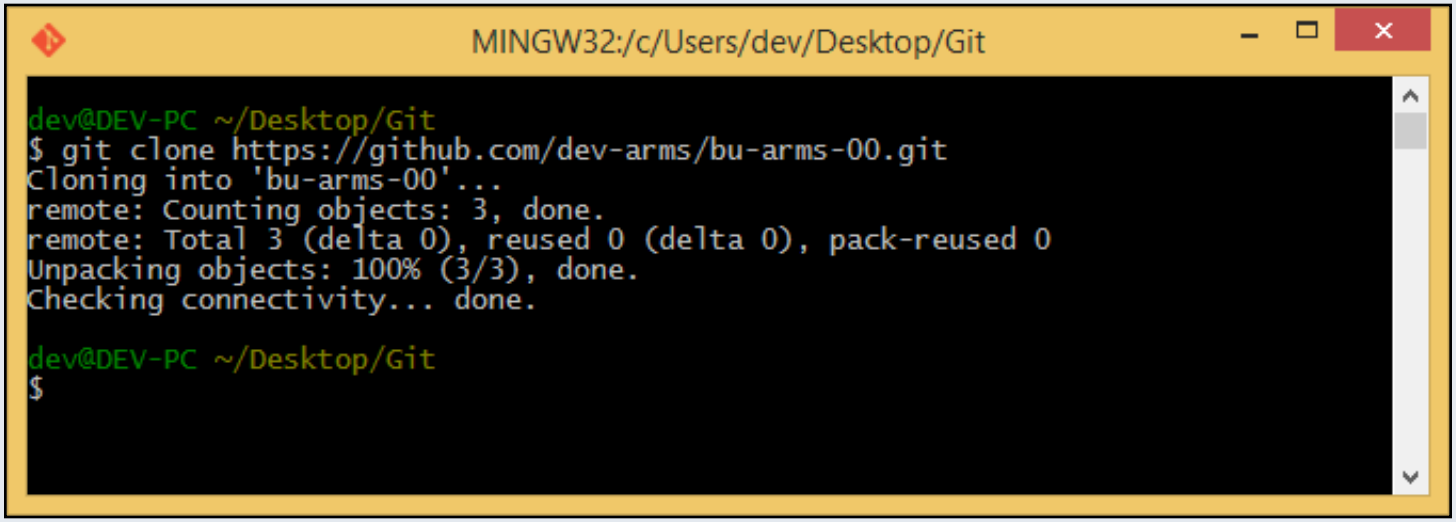
•For example….., if someone found how to solve an issue. You can clone repository, modify, commit, and push, then use pull request to merge.

•Try, Clone your team repository

You can copy URL



git clone **https://github.com/username/repositoryname.git**

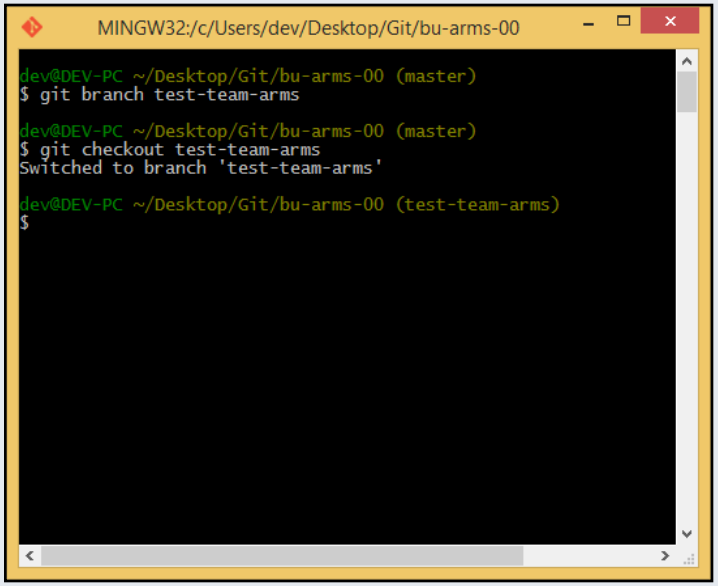


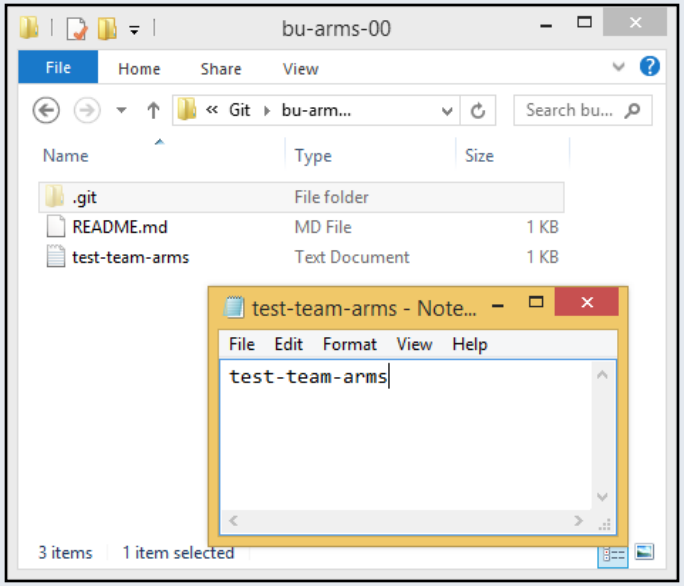
3. Getting started with Git and GitHub

3-5 Team collaboration

- Create branch "test-[your name]",
- check out to the branch and create a new file "test-[your name].txt"

git branch test-[your name]
git checkout test-[yourname]





3. Getting started with Git and GitHub

3-5 Team collaboration

•Add "test-[your name].txt" file to commit, Commit and push to Github using your id

```
git add test-[your name].txt
git commit -m "Add a new text file by [your name]"
git push origin test-[your name]
```

3. Getting started with Git and GitHub

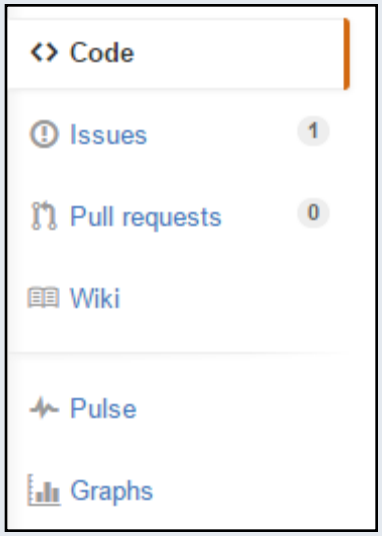
3-5 Team collaboration

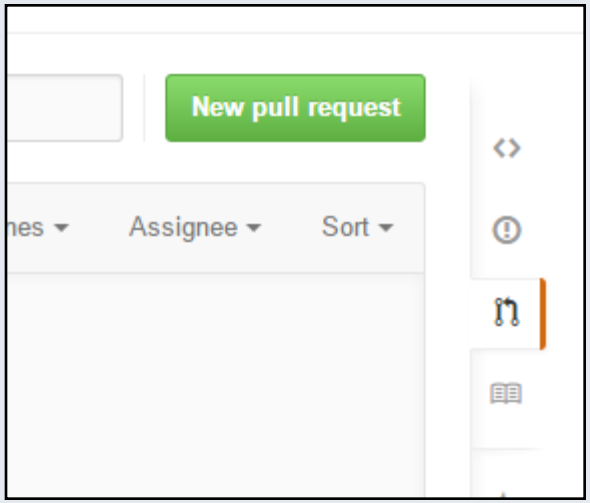
- We will use "Pull request"
- **Pull Requests are the heart of collaboration on GitHub**
- When you make a pull request, you're proposing changes in source code and requesting collaborators(and merge them into their branch)
- GitHub's Pull Request feature allows you to compare the content on two branches. The changes, additions and subtractions, are shown in green and red and called diffs (differences)
- People use Pull Requests to discuss commits (code review) even before the code is finished, that way you can get feedback as you go or help when you're stuck

3. Getting started with Git and GitHub

3-5 Team collaboration

- Create a Pull Request
- Click the Pull Request icon on the sidebar and click "New pull request" on your repository

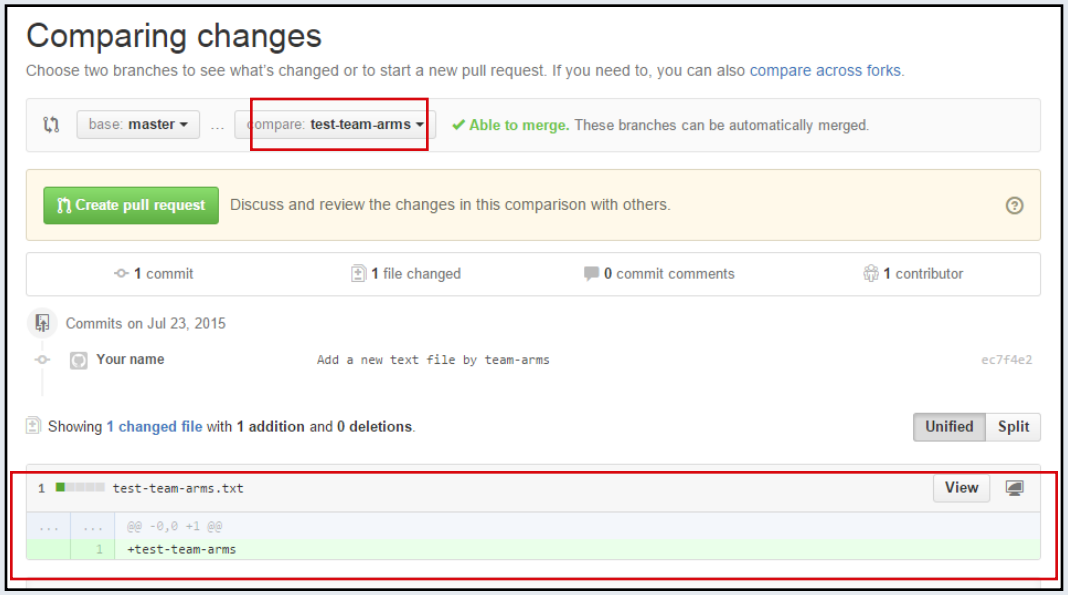




3. Getting started with Git and GitHub

3-5 Team collaboration

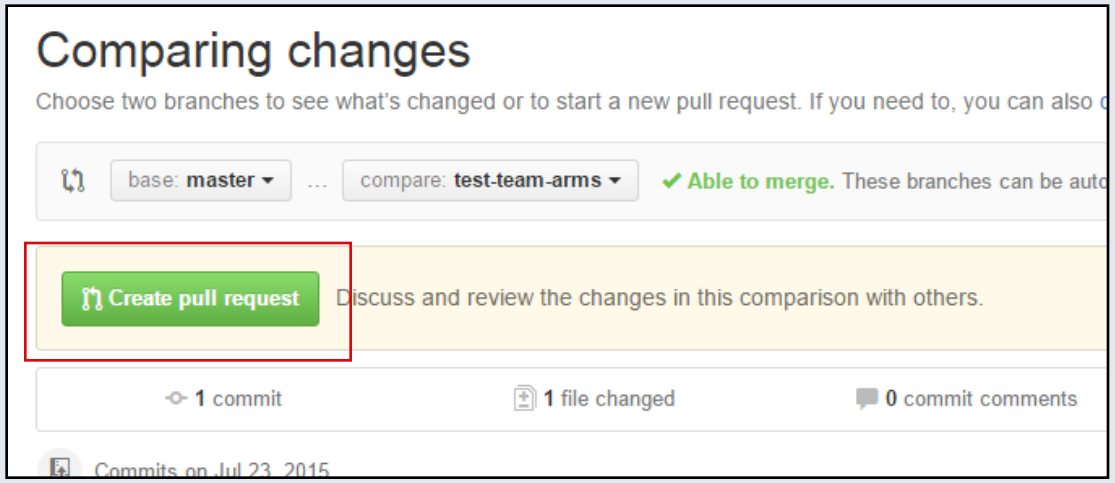
- Select the branch(test-[your name]) to compare with master
- Look over your changes in the diffs, make sure they're what you want to submit



3. Getting started with Git and GitHub

3-5 Team collaboration

•When you're satisfied with changes, click "Create Pull Request"

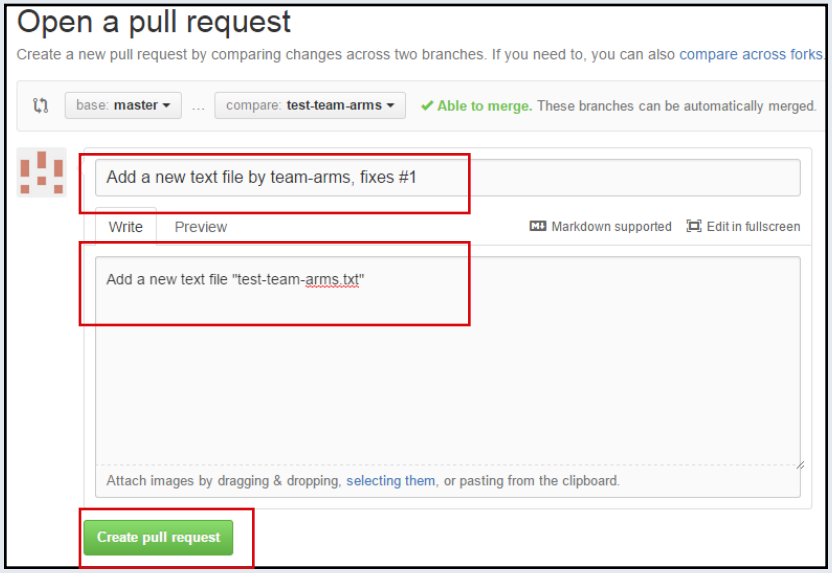


3. Getting started with Git and GitHub

3-5 Team collaboration

•Give your pull request a title that relates directly to an issue, include “fixes #” and the issue number in the title. Write a description of your changes

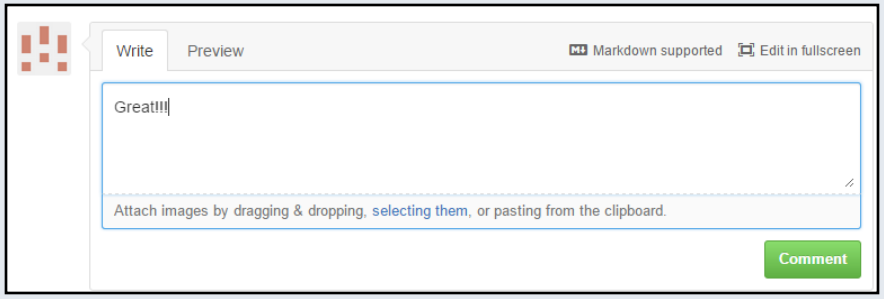
- title : **Add a new text file by [your name], fixes #1**
- description : **Add a new text file "test-[your names].txt"**

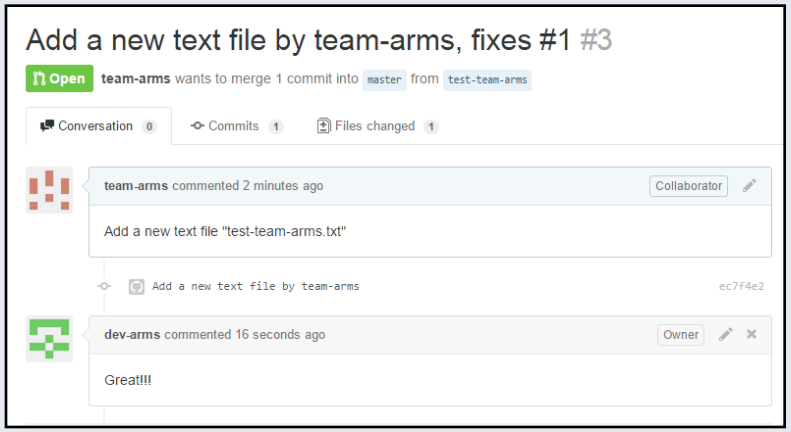


3. Getting started with Git and GitHub

3-5 Team collaboration

- You can comment for suggest, blame, tell the bug, and etc to this Pull request



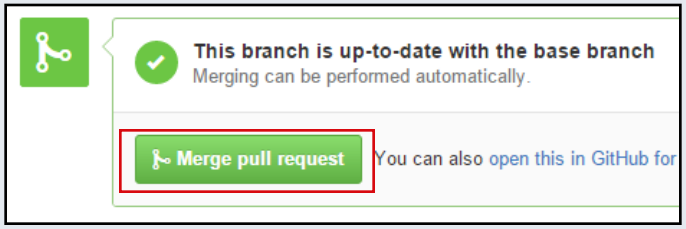


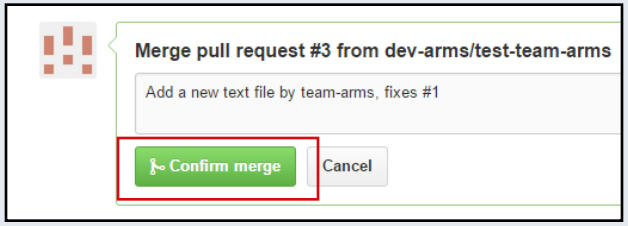
3. Getting started with Git and GitHub

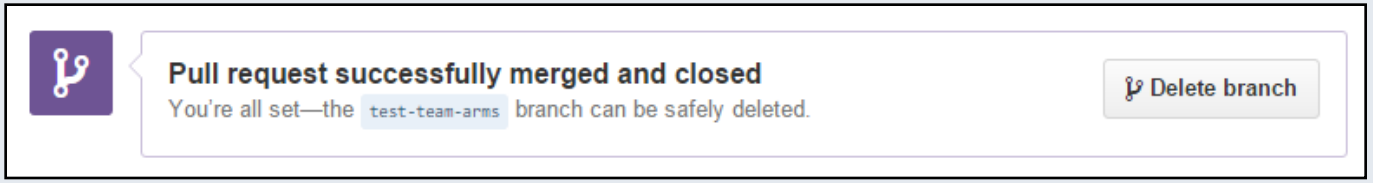
3-5 Team collaboration

•Bring changes together – merge your "test-[your name]" branch into the "master" branch

- Click "Merge pull request" to merge the changes into master
- Click "Confirm merge"
- Go ahead and delete the branch with the "Delete branch", (since its changes have been incorporated)



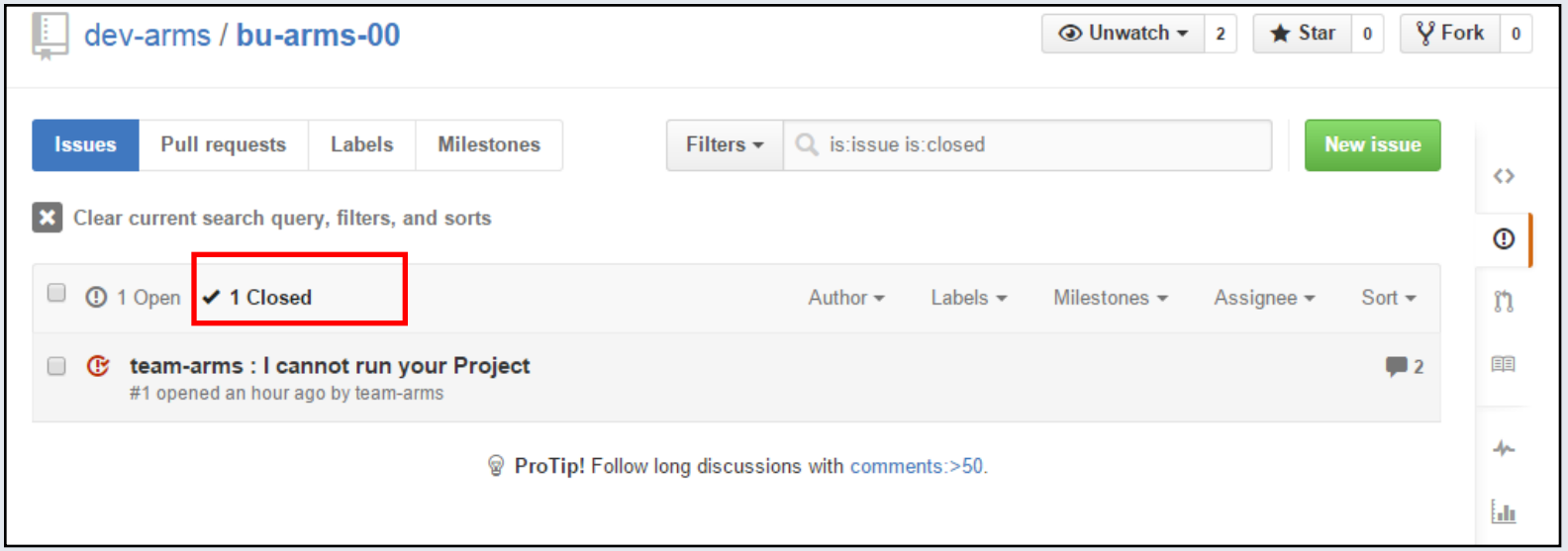




3. Getting started with Git and GitHub

3-5 Team collaboration

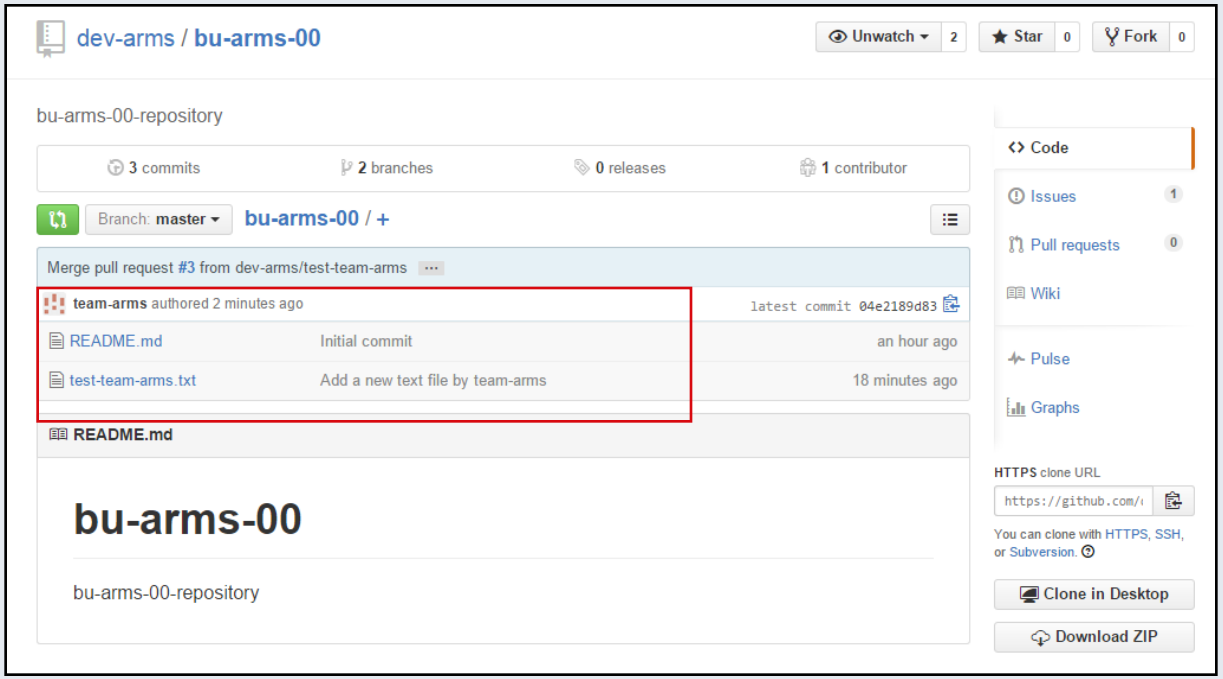
•the issue is closed



3. Getting started with Git and GitHub

3-5 Team collaboration

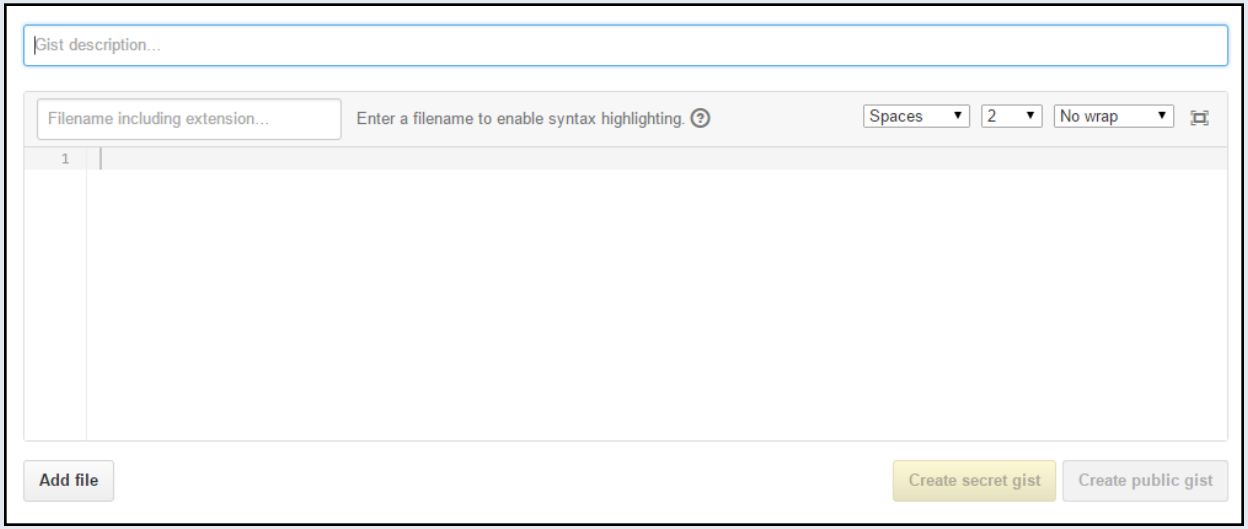
•You can see a new file or a lot of files (depends on the number of your members) on master branch because it's merged by pull request



3. Getting started with Git and GitHub

3-5 Team collaboration

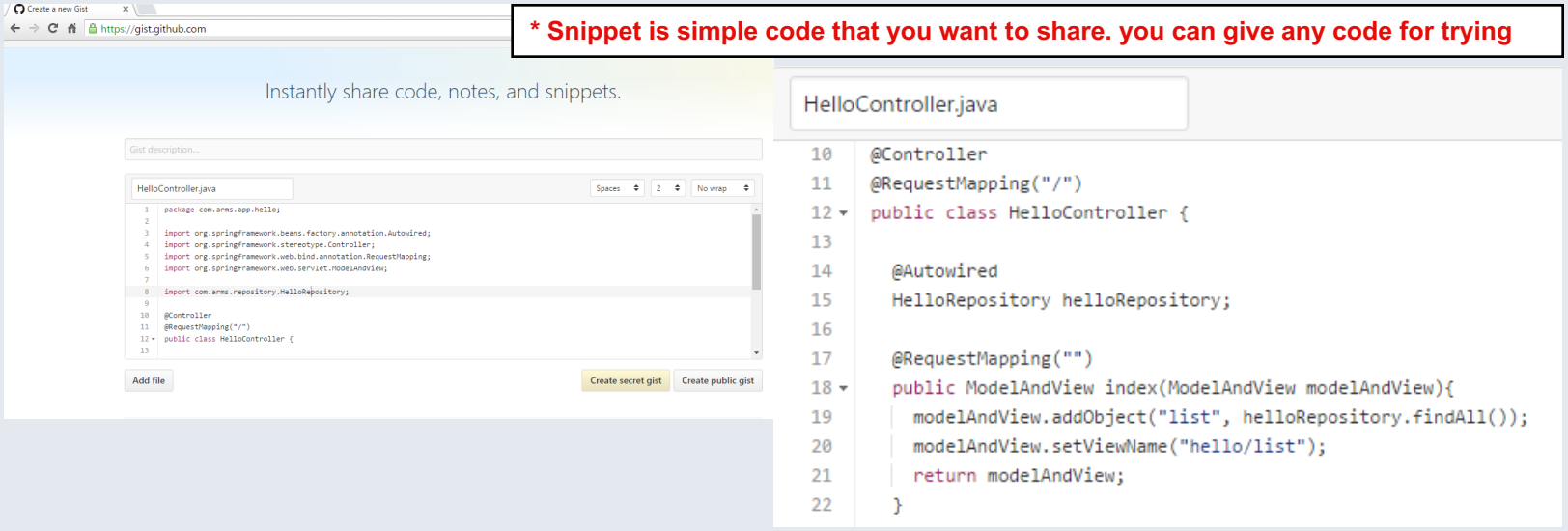
- For improving team collaboration, you can use Gist on Github
- Gist is a simple way to share snippets and pastes with others. All gists are Git repositories, so they are automatically versioned, forkable and usable from Git
- Try to create Gist, Open "**https://gist.github.com/**"



3. Getting started with Git and GitHub

3-5 Team collaboration

- Try to give snippet. For example, give Spring MVC Controller example

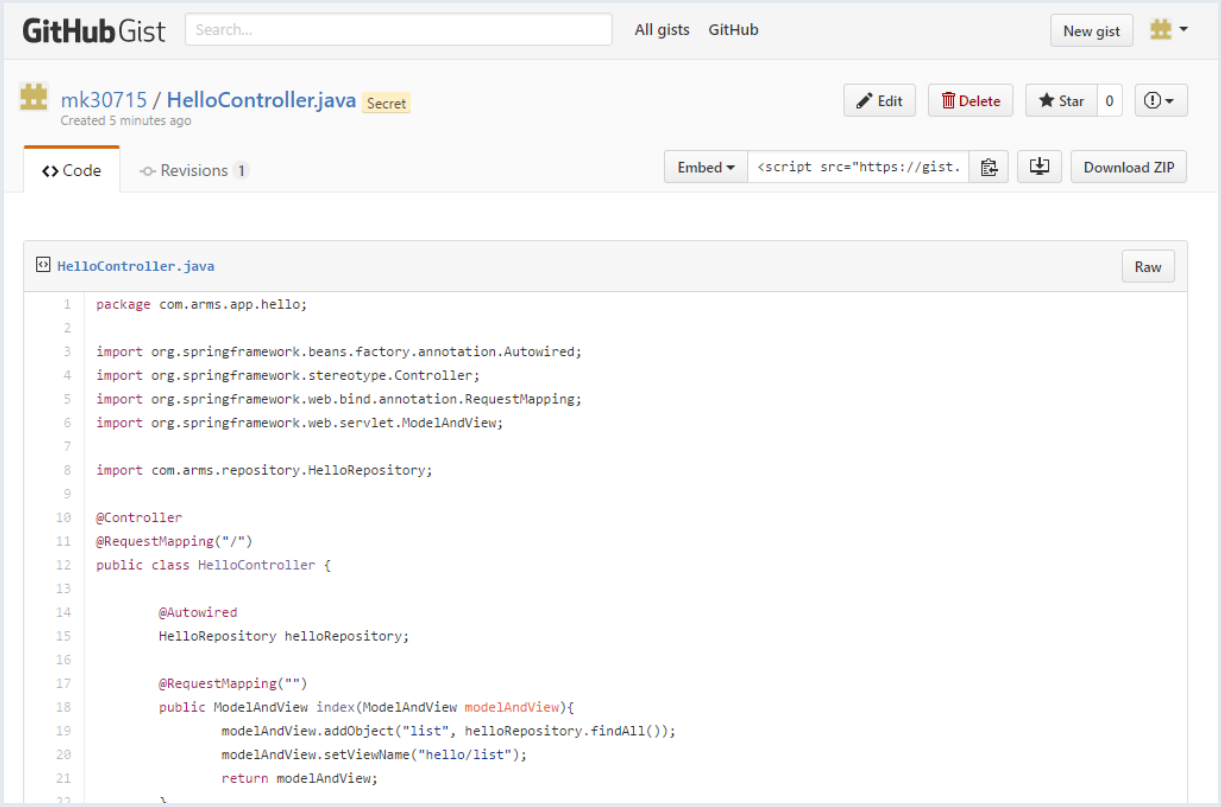


- Click "Create secret gist" or "Create public gist" depends on your need
- Secret gists are hidden from search engines but visible to anyone you give the URL

3. Getting started with Git and GitHub

3-7 Team collaboration

•Now you can share snippet with Gist to your team



4. Git commands

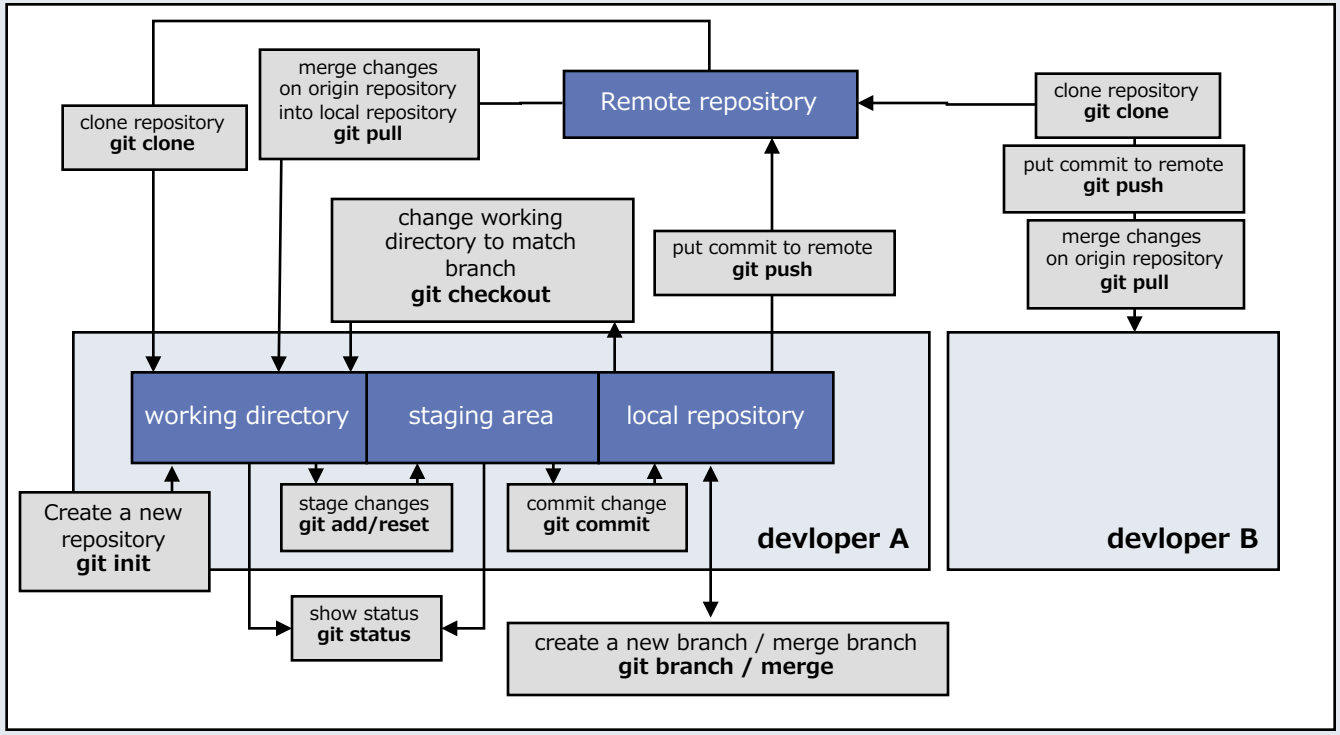
4-1 Git commands

•git init	Create a new Git local repository
•git clone	Clone a Git repository into a new directory
•git status	Show the working directory status
•git add	Add files to the staging area for tracking and committing
•git reset	Remove the specified file from the staging area
•git commit	Record changes to the repository
•git push	Put your commits on local to your origin repository
•git pull	Merge changes on origin repository into local repository
•git checkout	Updates the files in the working directory to match the version stored in that branch
•git branch	Create, list, rename, and delete branches
•git merge	Merge the specified branch into the current branch
•git log	Show commit history

4. Git commands

4-1 Git commands

•Simple command cycle



Your Idea Leads Your Ideals

homepage: http://arms-asia.com/
facebook: https://www.facebook.com/arms.asia?fref=ts