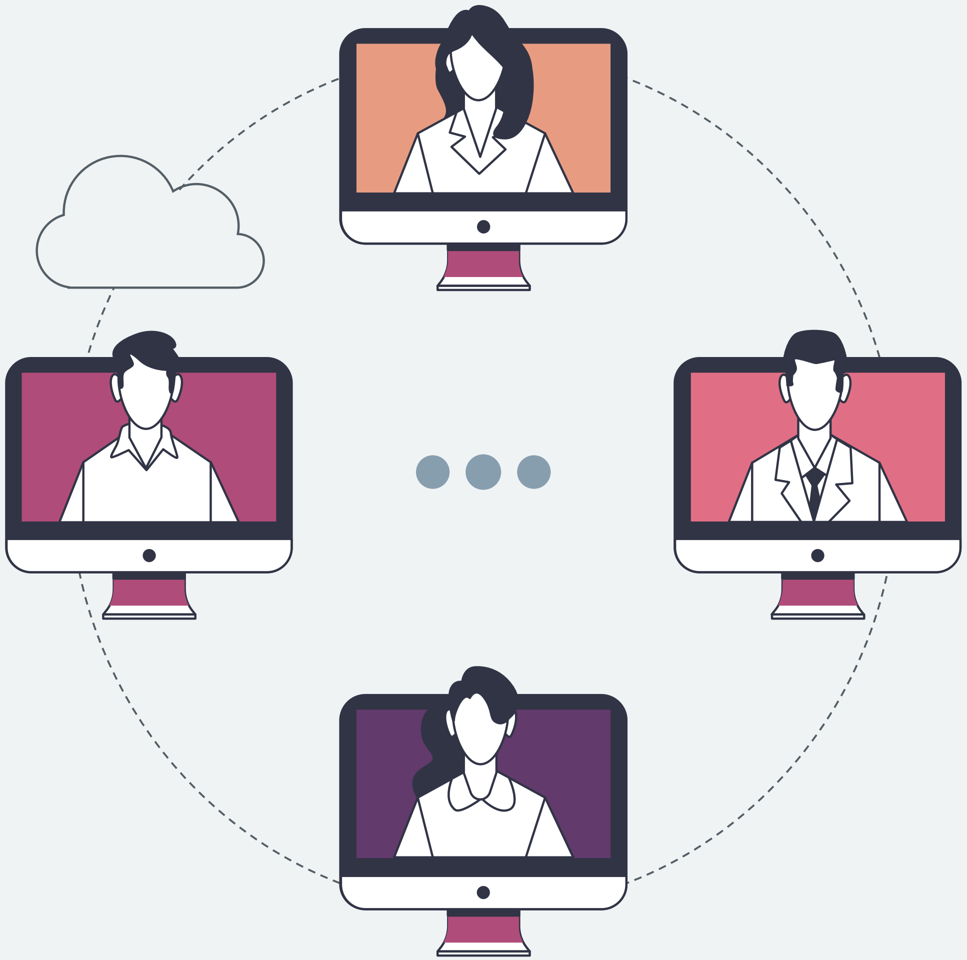
Cloud Computing



01

DevOps

- What is DevOps
- Why use DevOps
- DevOps Tools

01

What DevOps

DevOps is a cross-disciplinary practice where application development (dev) works together with IT operations (ops) to improve product quality and accelerate time to market.

In a DevOps framework, developers and IT managers collaborate with experts in quality control, security, and support.

How DevOps Works ?

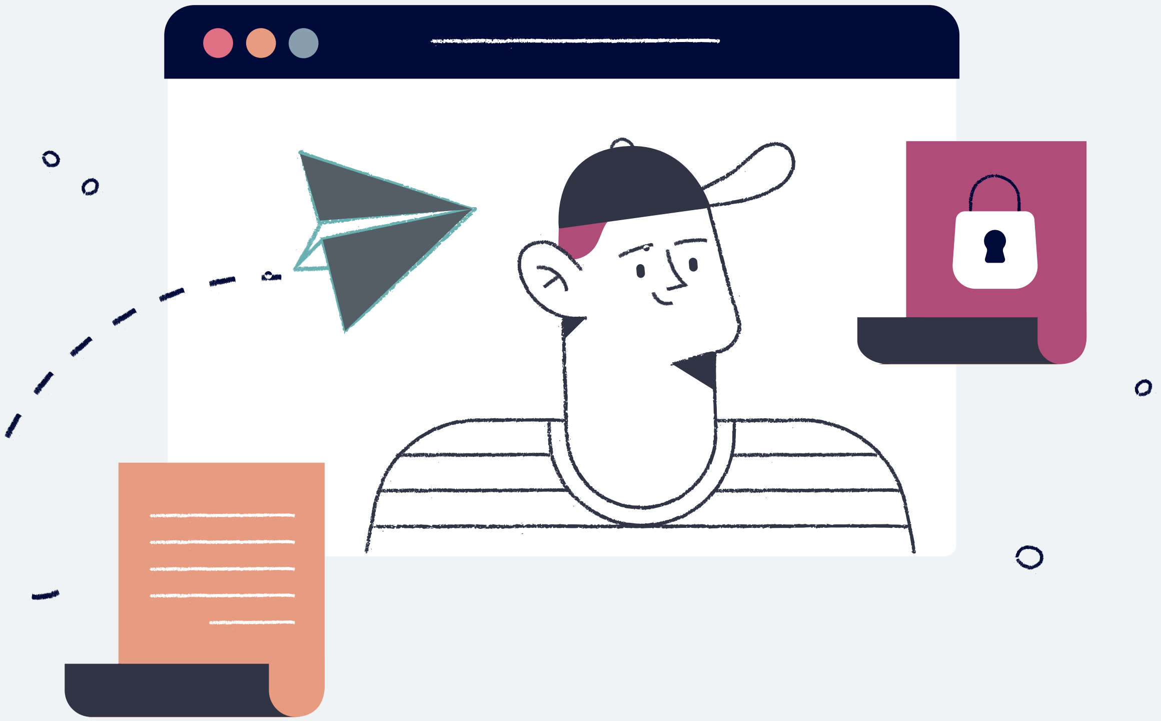
The goal of this cooperative effort it to deliver code rapidly, in a seamless loop of continuous integration and continuous delivery (CI/CD).

DevOps builds on the frequent releases and CI/CD aspects of agile development methods but adds infrastructure management into the development process to make product delivery more flexible and dynamic.

That process integration, in turn, depends on the ongoing collaboration among teams, including business unit management, developers, quality assurance, and IT operations. DevOps approach replaces silos, rigid job descriptions, and bottlenecks with a flexible, cross-disciplinary model that enables innovation and continuous improvement.

01

Cloud DevOps



When DevOps teams work in the cloud, they enjoy easier access to scalable hardware resources that can help build, test, and deploy new updates and offerings more quickly.

Traditional Applications

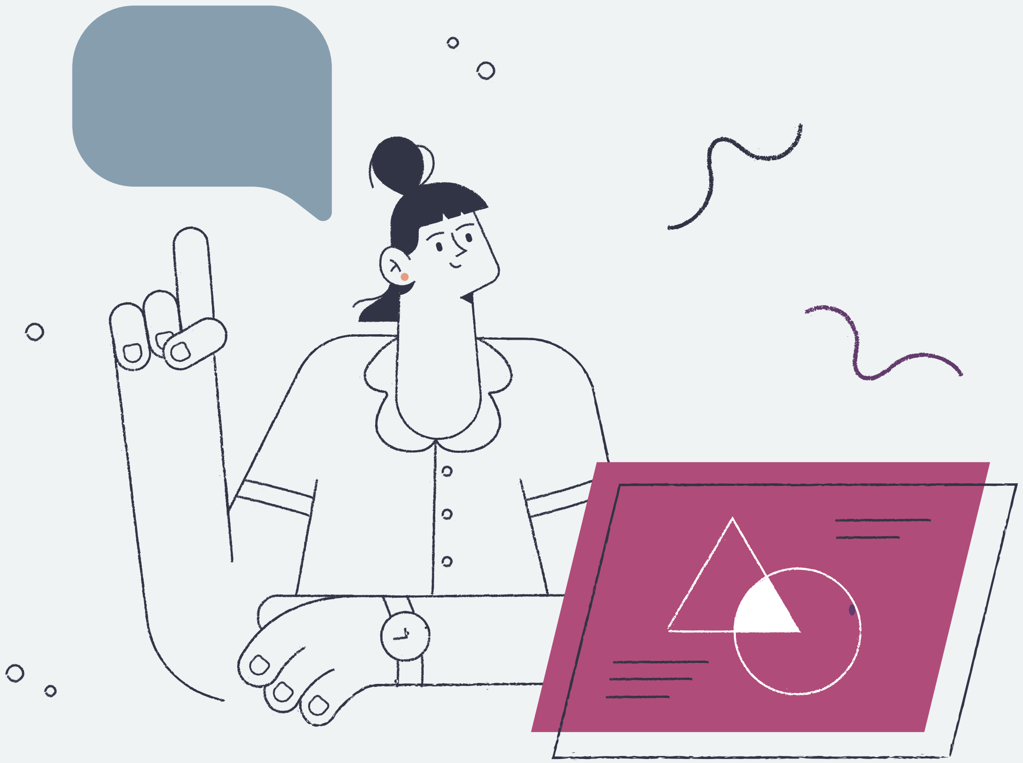
finished application handed off to IT operations for maintenance, with future upgrades

Cloud DevOps

the application stack is likely to continue changing after its initial deployment

Cloud DevOps

A DevOps team can be more efficient as well. The multidisciplinary team can take advantage of virtualization and containerization in the cloud to develop and test applications in identical environments simultaneously and to provision additional resources as needed.



DevOps Benefits

CODE

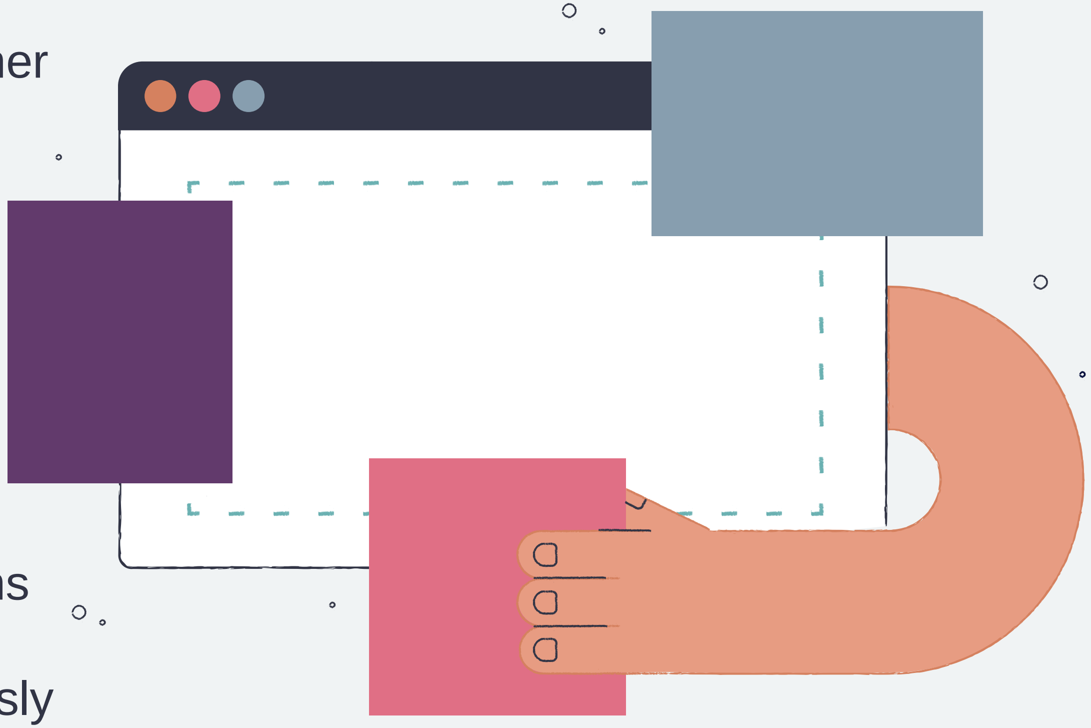
- released faster and more frequently
- bug fixes
- features delivered to the market sooner

IMPROVEMENT SUPPORT

- constant iterations, monitoring, and testing
- positive user experience
- strong competitive advantage

SECURITY

- security best practices and regulations
- Security management integrated, automated, and monitored continuously



01

Adaptability

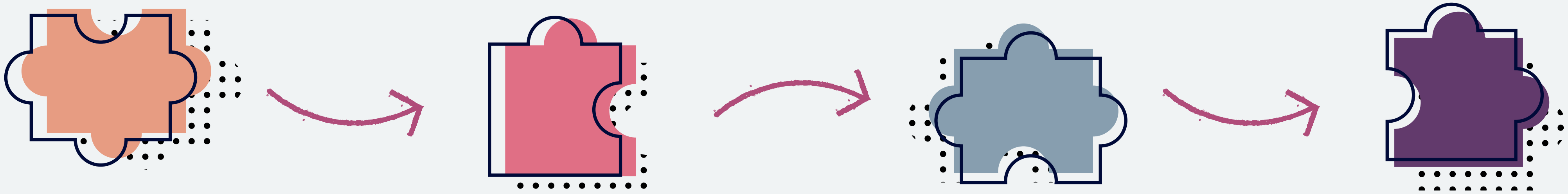
How to Begin Using DevOps in Your Team

Clarify roles and responsibilities

Define working agreements

Anticipate risks

Conduct regular reviews



01

DevOps Tools

Version Control

GitLab, GitHub, Bitbucket

Configuration Management

Chef, Puppet, Ansible

Container Platforms

Docker, Kubernets

Communication& Collaboration

Slack

Cloud Computing & Storgae

AWS, Azure, Google cloud

Testing

Seleinum, Gremlin

02

Git

- What is Git
- How to use Git

What is Git ?

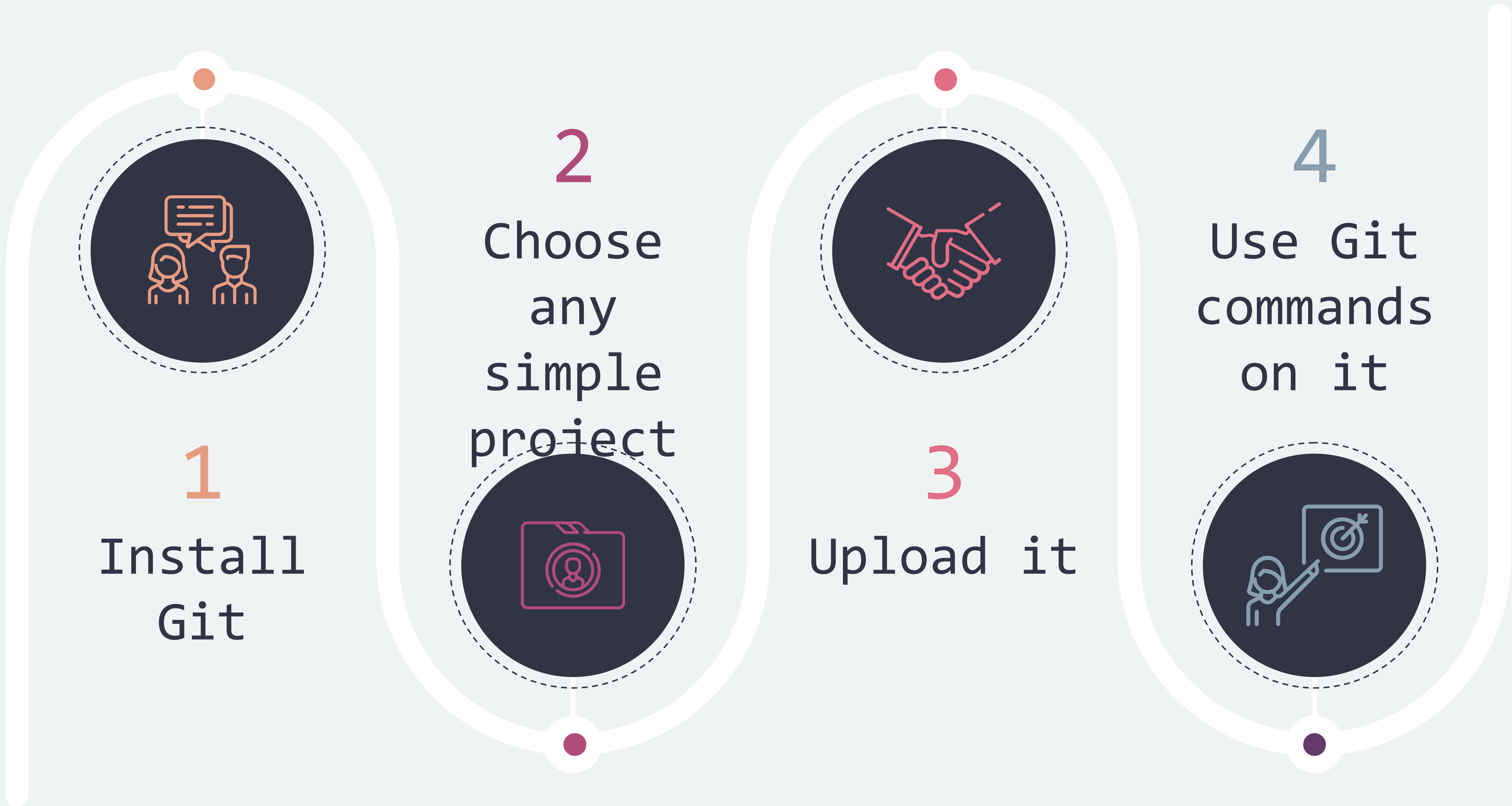
- Git is an open source and free Version Control Management what’s referred to as VCM .
- With Git , you can manage changes to files over time.
- You can use git from its Terminal “ Git Bash” or any other terminals like PowerShell , CMD .

02

Git Installation

- Go to this link to install git for windows : https://git-scm.com/download/win
- Follow the installation in instructions to start using Git
- Make sure to download the latest version of Git .
- Choose what is suitable to your device : 64 or 32 bit .

Steps to Start using Git



1

Install Git

2

Choose any simple project

3

Upload it

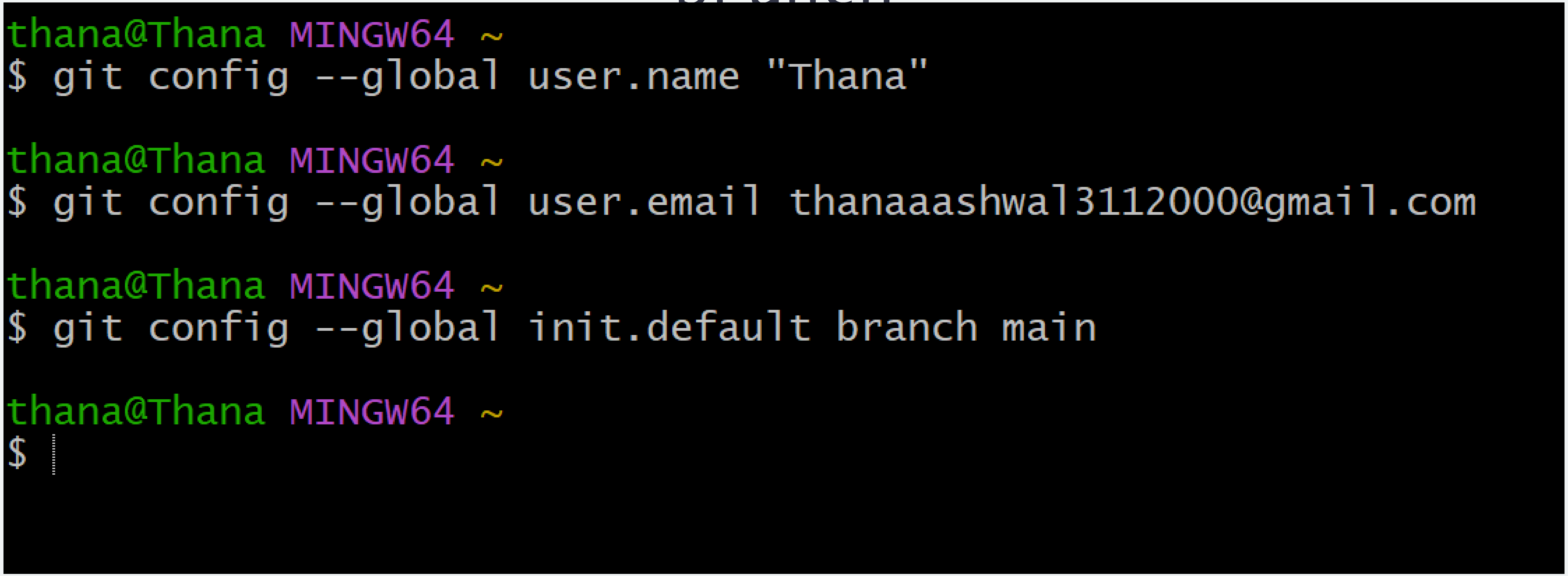
4

Use Git commands on it

<<Lets Open Git Bash and start working>>

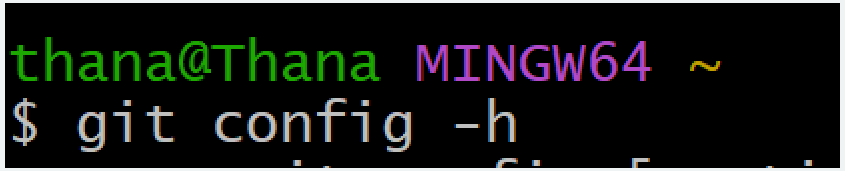
02

Specify username and email and main branch



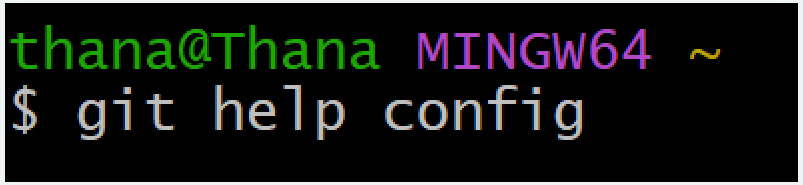
```
thana@Thana MINGW64 ~
$ git config --global user.name "Thana"

thana@Thana MINGW64 ~
$ git config --global user.email thanaaashwal3112000@gmail.com

thana@Thana MINGW64 ~
$ git config --global init.default branch main

thana@Thana MINGW64 ~
$
```

Get help from Git



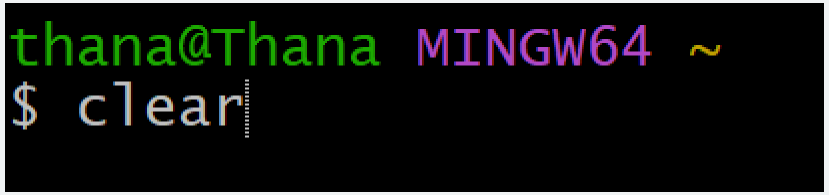
```
thana@Thana MINGW64 ~
$ git config -h
```

This command will present all of git config commands and their jobs



```
thana@Thana MINGW64 ~
$ git help config
```

This command will open an offline site with all git config commands and details about them

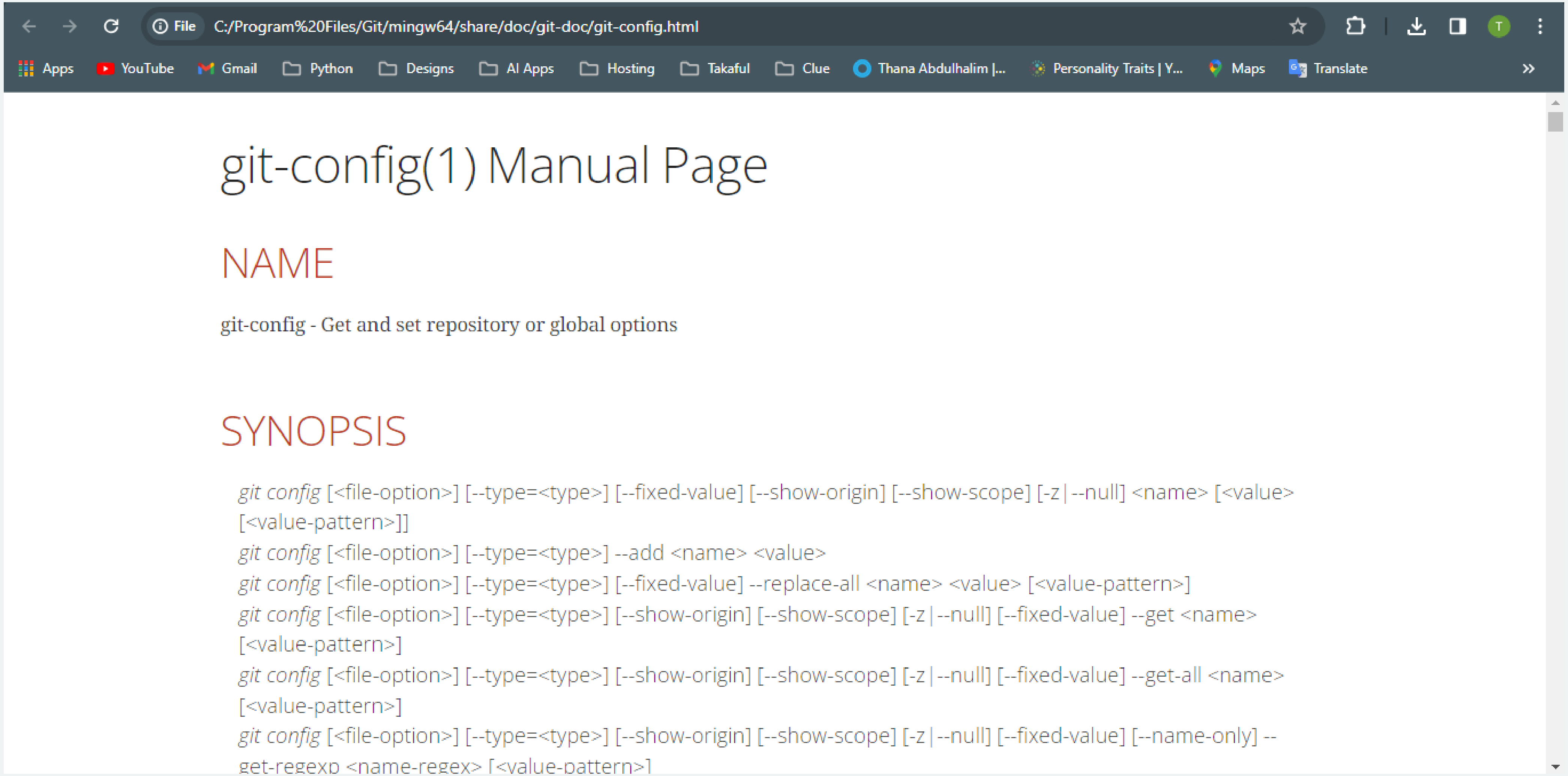


```
thana@Thana MINGW64 ~
$ clear
```

This command will clear the screen

02

Get help from Git



git-config(1) Manual Page

NAME

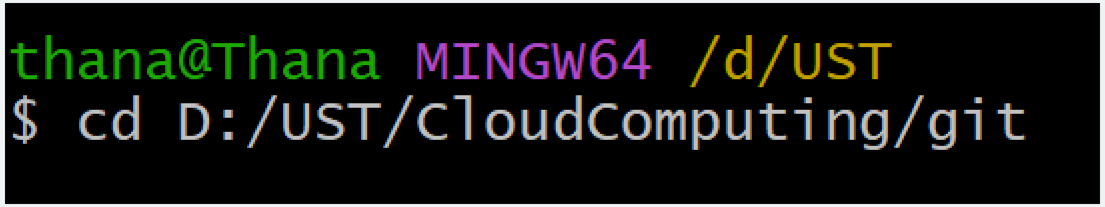
git-config - Get and set repository or global options

SYNOPSIS

git config [<file-option>] [--type=<type>] [--fixed-value] [--show-origin] [--show-scope] [-z|--null] <name> [<value> [<value-pattern>]]
git config [<file-option>] [--type=<type>] --add <name> <value>
git config [<file-option>] [--type=<type>] [--fixed-value] --replace-all <name> <value> [<value-pattern>]
git config [<file-option>] [--type=<type>] [--show-origin] [--show-scope] [-z|--null] [--fixed-value] --get <name> [<value-pattern>]
git config [<file-option>] [--type=<type>] [--show-origin] [--show-scope] [-z|--null] [--fixed-value] --get-all <name> [<value-pattern>]
git config [<file-option>] [--type=<type>] [--show-origin] [--show-scope] [-z|--null] [--fixed-value] [--name-only] --get-regexp <name-regex> [<value-pattern>]

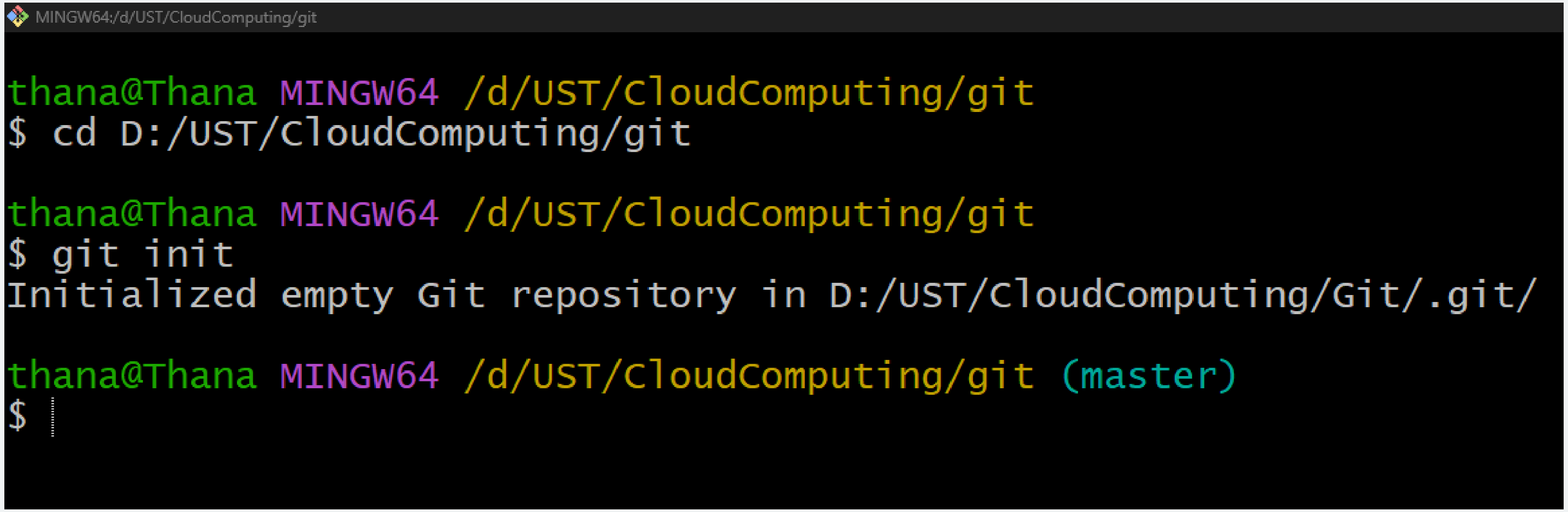
02

Create a git repository



```
thana@Thana MINGW64 /d/UST
$ cd D:/UST/CloudComputing/git
```

Go to your project file location



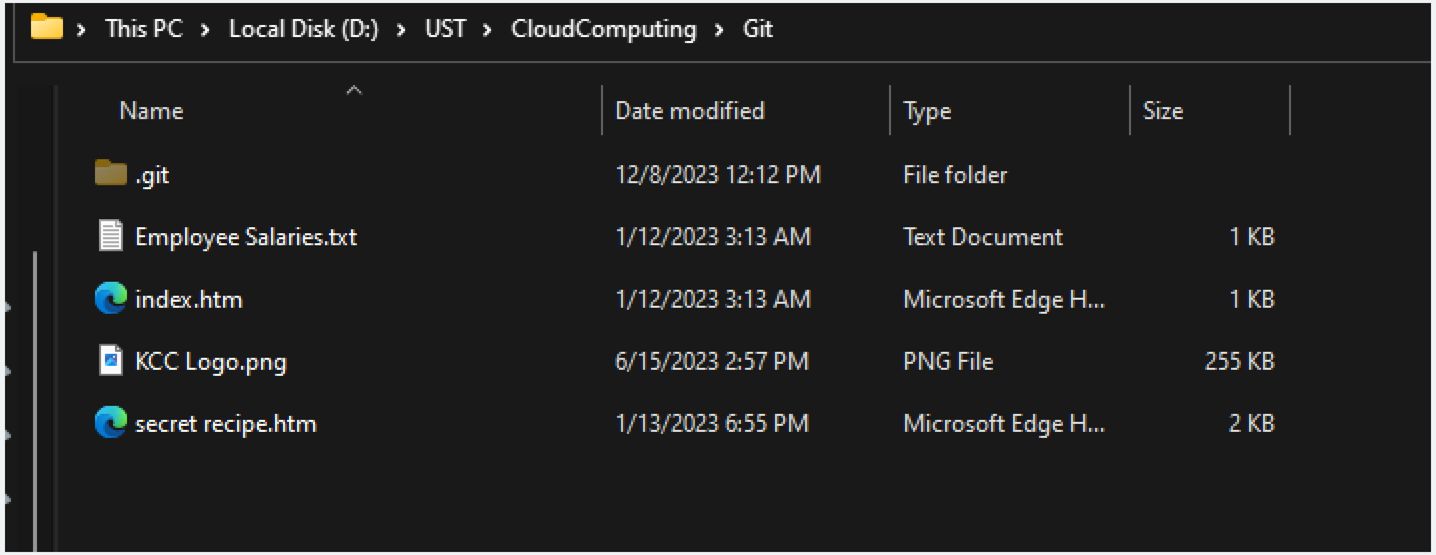
MINGW64:/d/UST/CloudComputing/git

```
thana@Thana MINGW64 /d/UST/CloudComputing/git
$ cd D:/UST/CloudComputing/git

thana@Thana MINGW64 /d/UST/CloudComputing/git
$ git init
Initialized empty Git repository in D:/UST/CloudComputing/Git/.git/

thana@Thana MINGW64 /d/UST/CloudComputing/git (master)
$ 
```

This command will create a git repository

Create a git repository

Go back to your project file , you will see .git file .
If not , click show hidden items option



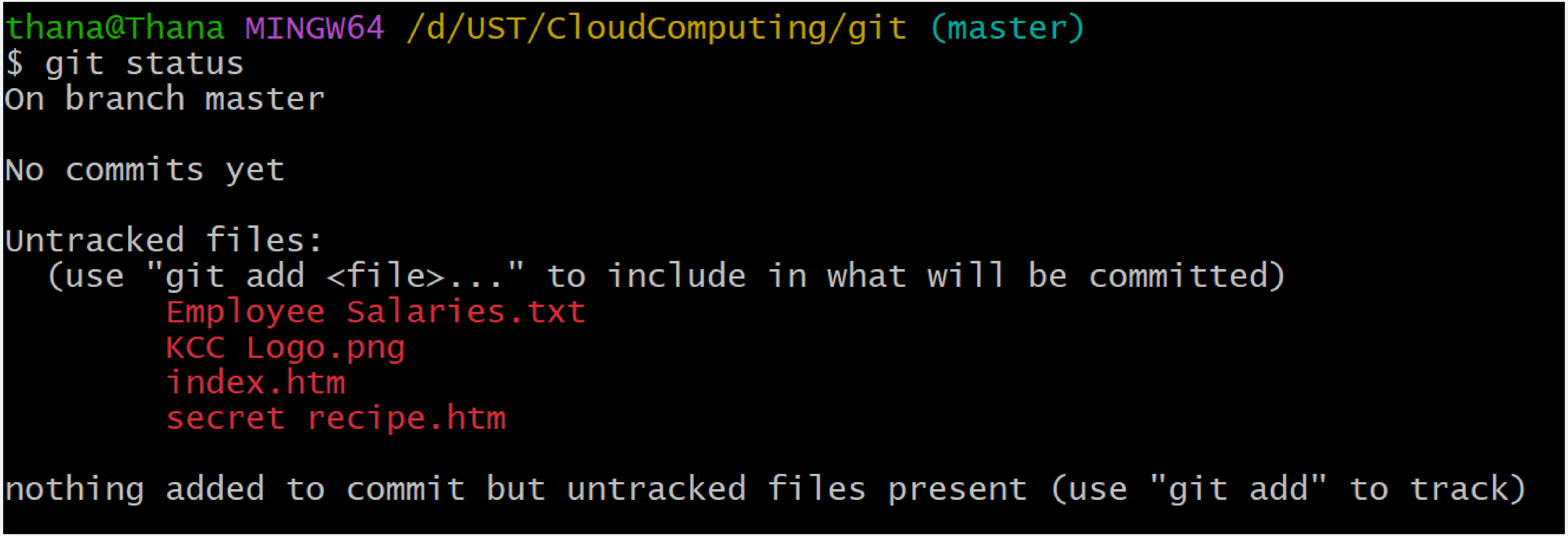
This PC › Local Disk (D:) › UST › CloudComputing › Git

Name	Date modified	Type	Size
.git	12/8/2023 12:12 PM	File folder	
Employee Salaries.txt	1/12/2023 3:13 AM	Text Document	1 KB
index.htm	1/12/2023 3:13 AM	Microsoft Edge H...	1 KB
KCC Logo.png	6/15/2023 2:57 PM	PNG File	255 KB
secret recipe.htm	1/13/2023 6:55 PM	Microsoft Edge H...	2 KB

02

Create a git repository

Go back git bash , write this command to see your branch status



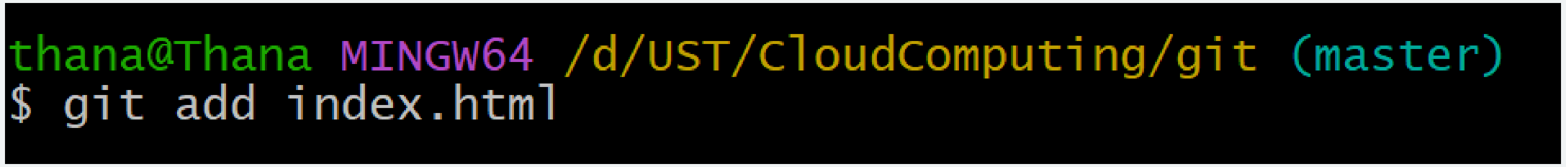
```
thana@Thana MINGW64 /d/UST/CloudComputing/git (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Employee Salaries.txt
        KCC Logo.png
        index.htm
        secret recipe.htm

nothing added to commit but untracked files present (use "git add" to track)
```

Untracked Files means that whatever the changes have happened to these files git will not care about them .

02

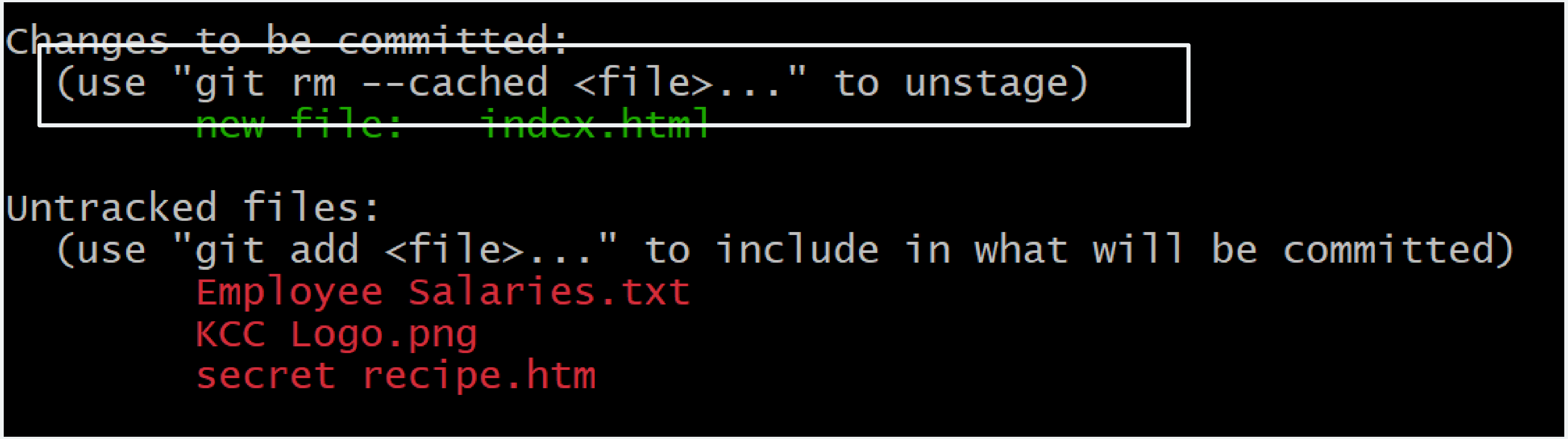
Track Files

Write this command to track any file



```
thana@Thana MINGW64 /d/UST/CloudComputing/git (master)
$ git add index.html
```

Write status command again to check whether the file is tracked or not .
Also you can use this command to untrack a file



```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Employee Salaries.txt
        KCC Logo.png
        secret recipe.htm
```

Create a git ignore file

You can ignore any files that you don non want to use version control on them by doing these steps :

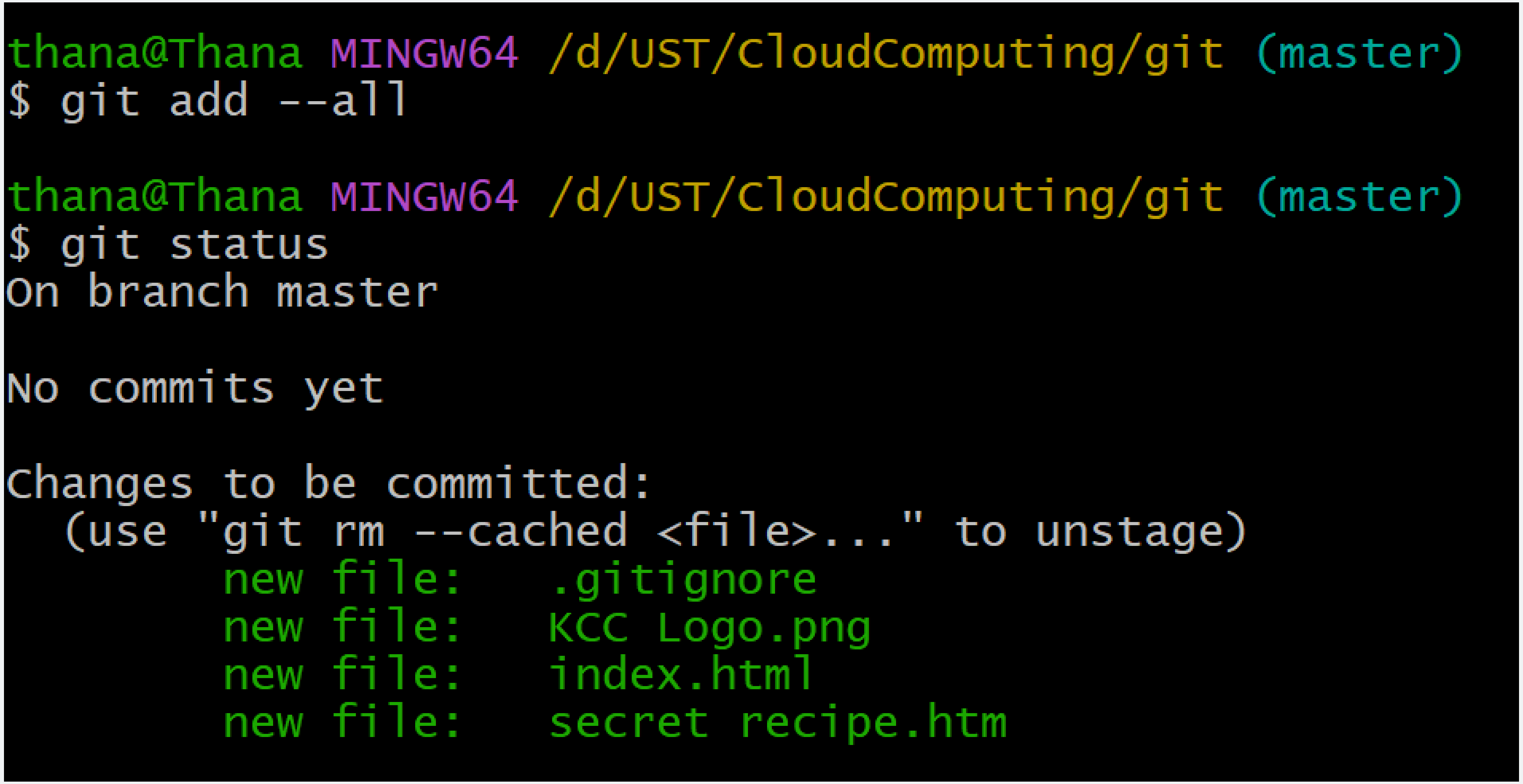
- Go to your project folder and create a new txt file .
- Name the file “ .gitignore ” .
- Delete the file extension .
- Open the file using notepad .
- write these inside the file:

  ```
  # ignore all .txt files
  *.txt
  ```

- Then save the file .
- Go back to bash and use status again , all txt files will be ignored

02

Track all files

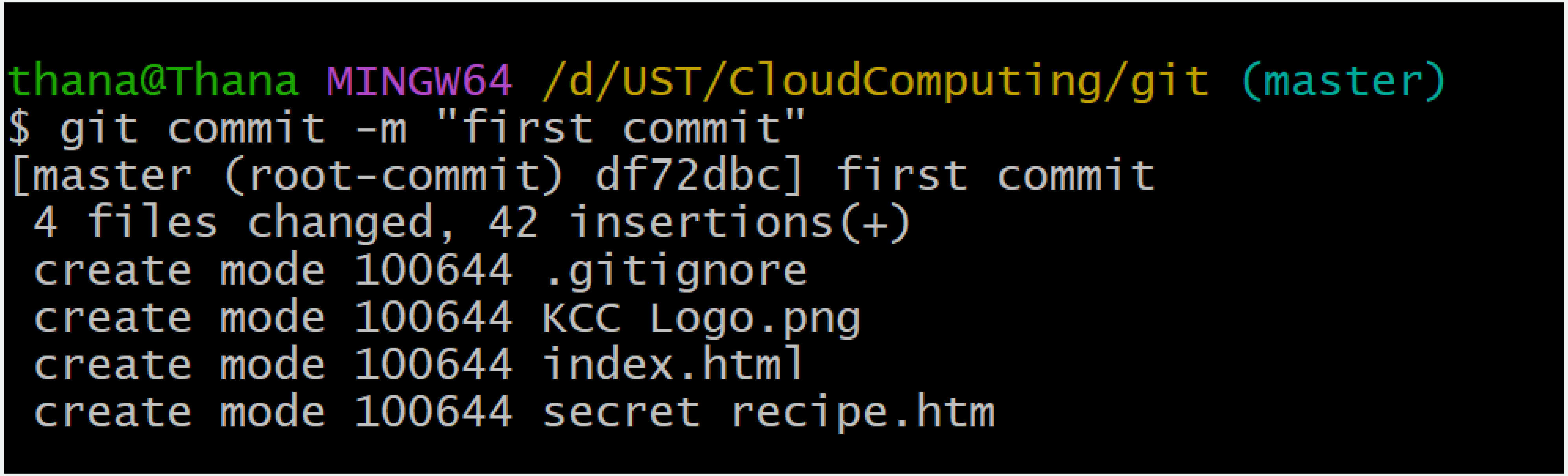


```
thana@Thana MINGW64 /d/UST/CloudComputing/git (master)
$ git add --all

thana@Thana MINGW64 /d/UST/CloudComputing/git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   KCC Logo.png
        new file:   index.html
        new file:   secret recipe.htm
```

02

Commit Files

Commit means to take a snapshot of your project at a point of time.
Which mean what your project looks like in this exact time.
When you commit something you have to write a commit message.

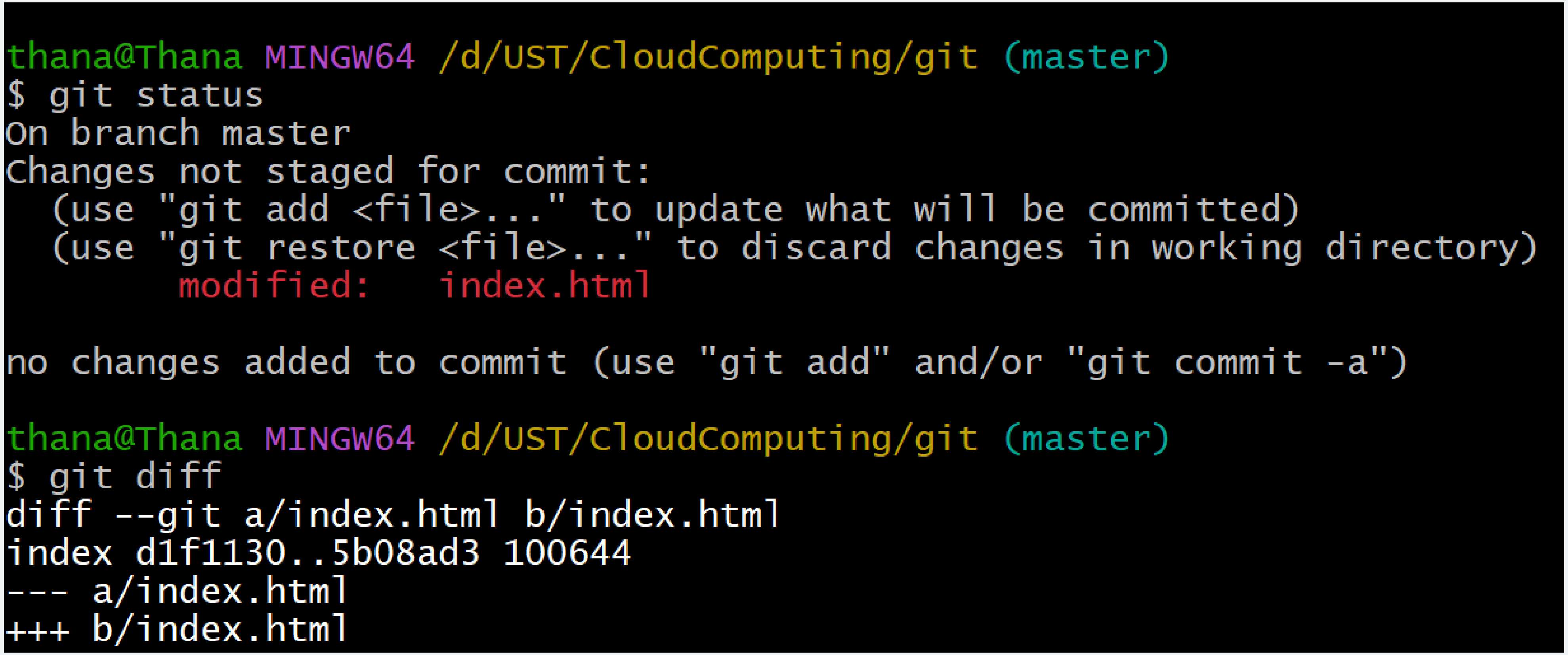


```
thana@Thana MINGW64 /d/UST/CloudComputing/git (master)
$ git commit -m "first commit"
[master (root-commit) df72dbc] first commit
 4 files changed, 42 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 KCC Logo.png
 create mode 100644 index.html
 create mode 100644 secret recipe.htm
```

Now if you write status command again , there will not be any files to commit . But if you change anything in files , it will show the file that had changes on it.

Changes

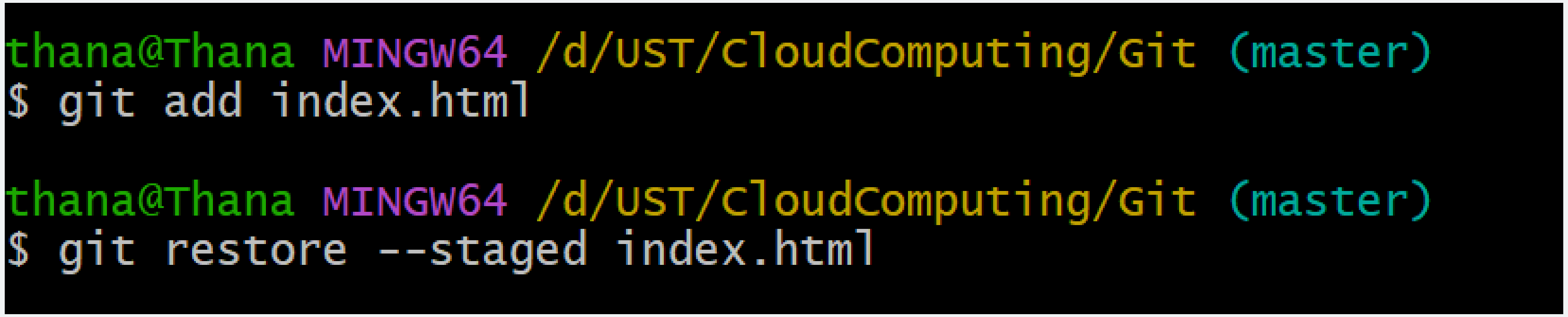
Write “ git diff ” to know where the changes had happened exactly.
Scroll down to see the text.



```
thana@Thana MINGW64 /d/UST/CloudComputing/git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

thana@Thana MINGW64 /d/UST/CloudComputing/git (master)
$ git diff
diff --git a/index.html b/index.html
index d1f1130..5b08ad3 100644
--- a/index.html
+++ b/index.html
```

Staging

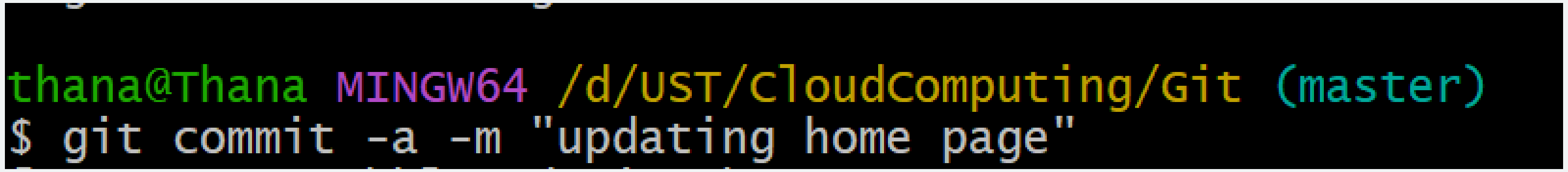
It is the second step after changes, you can stage file and delete them from stage using these commands:



```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git add index.html

thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git restore --staged index.html
```

Add command will make your changes ready to commit while restore command will take them back to unready status

If you want to skip staging step and directly commit write this command:

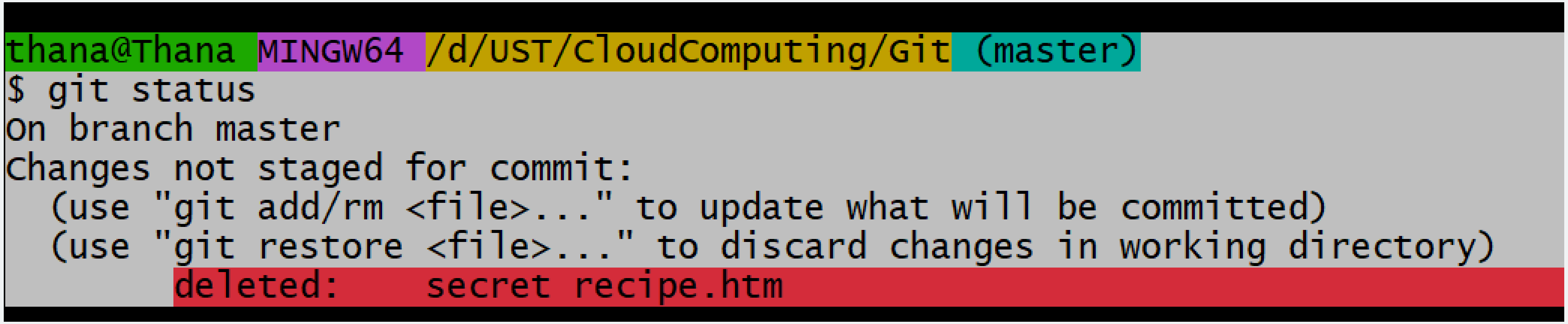


```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git commit -a -m "updating home page"
```

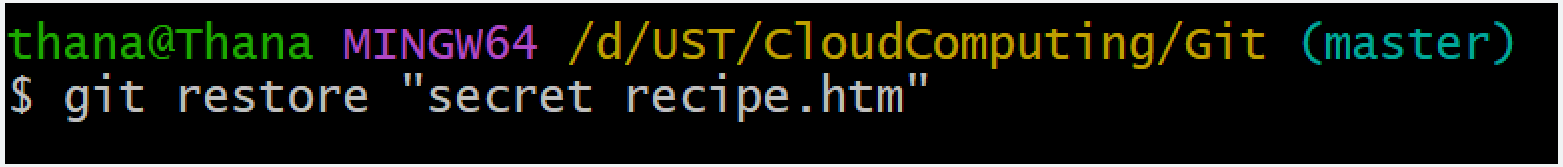
02

Editing Files

If some of your files have been deleted and you want to store them back, use this command:

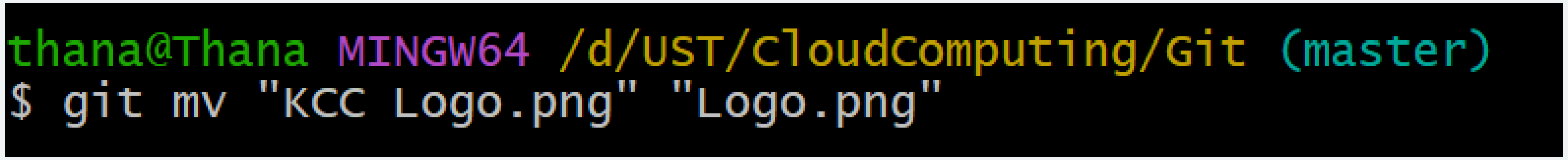


```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    secret recipe.htm
```



```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git restore "secret recipe.htm"
```

You can use this command to change files name:

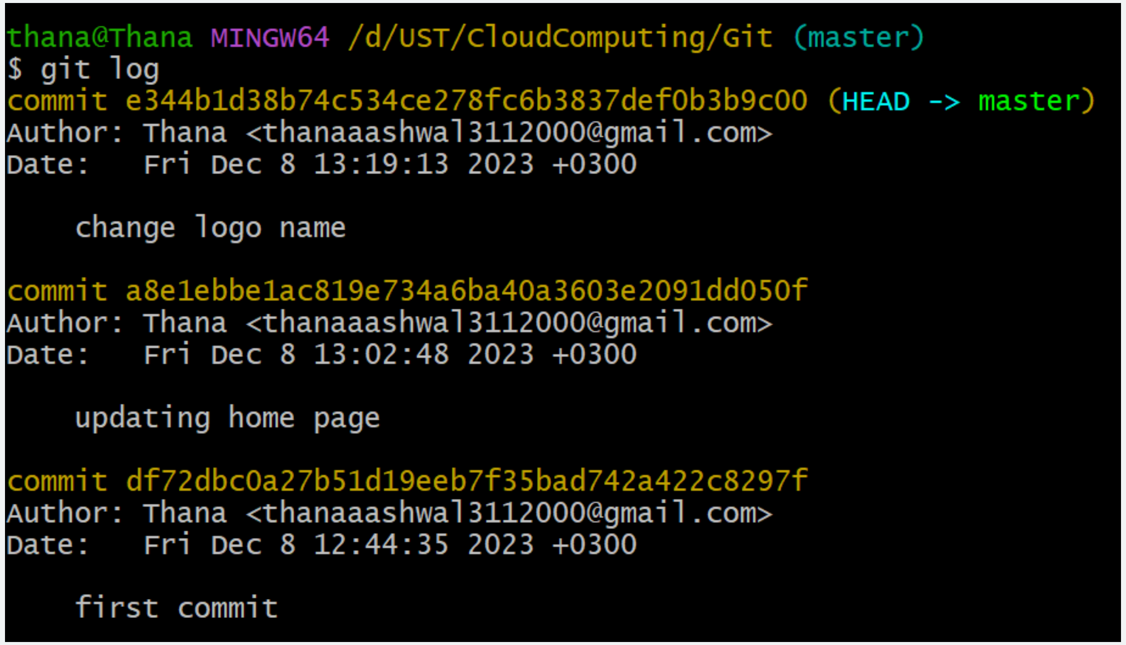


```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git mv "KCC Logo.png" "Logo.png"
```

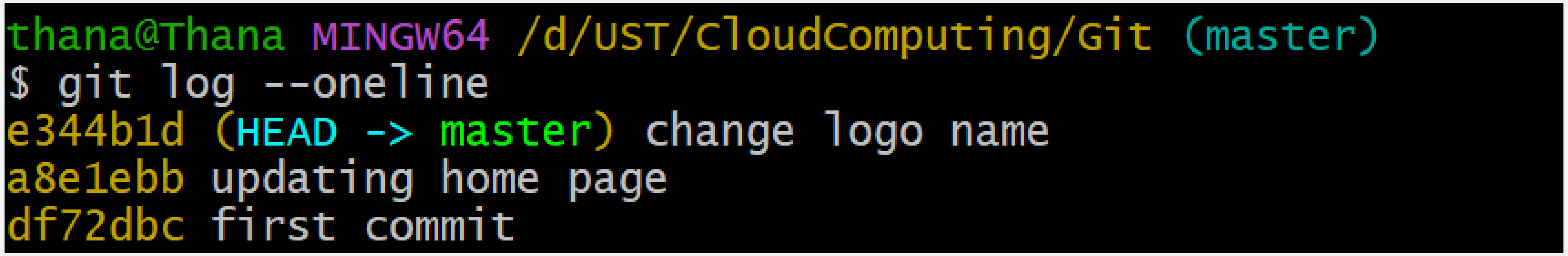
Note: commit after every changes, don not forget the commit message

Commit Log

You can see all commits and all details about them using this command:



```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git log
commit e344b1d38b74c534ce278fc6b3837def0b3b9c00 (HEAD -> master)
Author: Thana <thanaaashwal3112000@gmail.com>
Date:   Fri Dec 8 13:19:13 2023 +0300

    change logo name

commit a8e1ebbe1ac819e734a6ba40a3603e2091dd050f
Author: Thana <thanaaashwal3112000@gmail.com>
Date:   Fri Dec 8 13:02:48 2023 +0300

    updating home page

commit df72dbc0a27b51d19eeb7f35bad742a422c8297f
Author: Thana <thanaaashwal3112000@gmail.com>
Date:   Fri Dec 8 12:44:35 2023 +0300

    first commit
```



```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git log --oneline
e344b1d (HEAD -> master) change logo name
a8e1ebb updating home page
df72dbc first commit
```

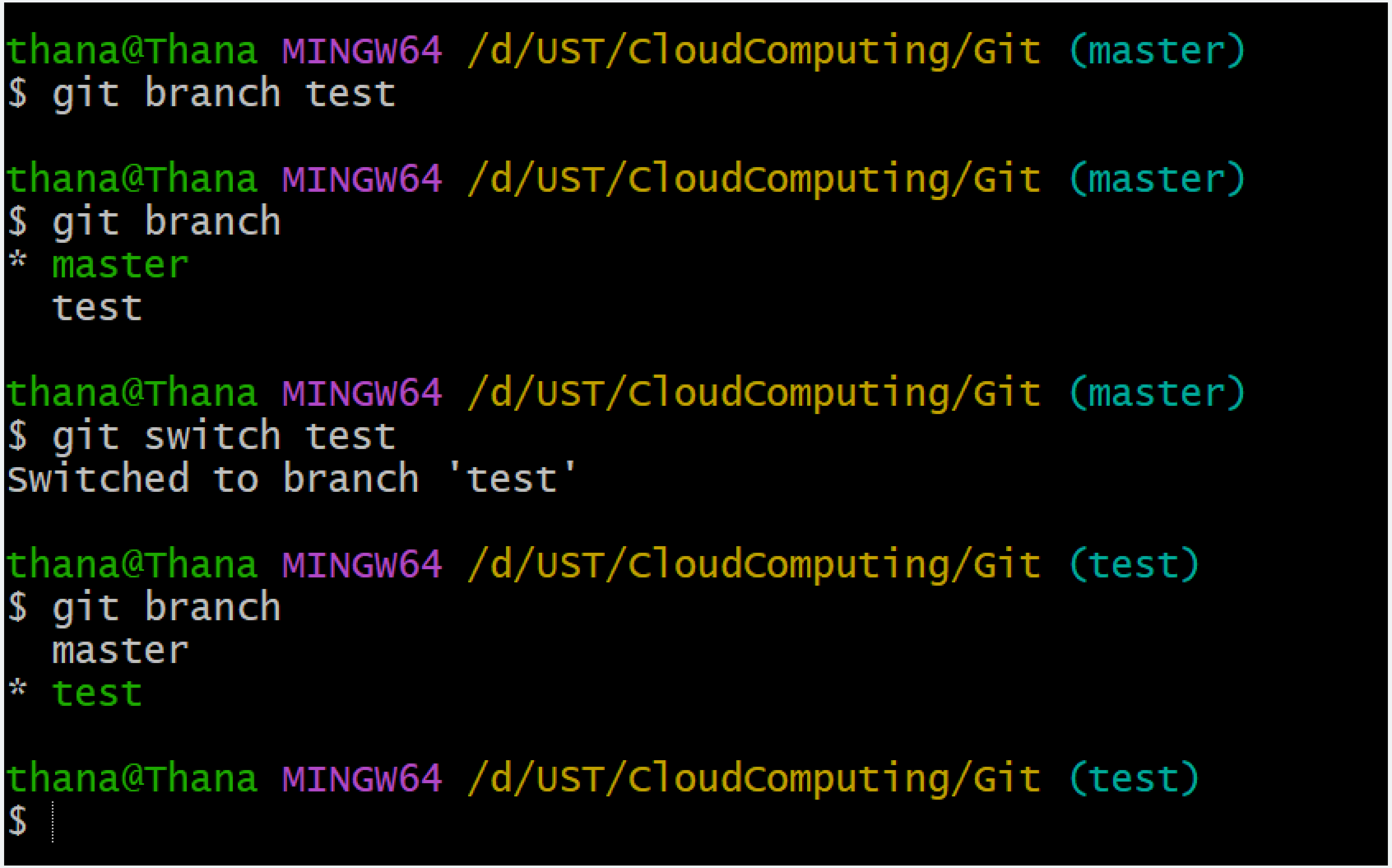
02

Creating Another Branch

You can create another branch to make some changes on it , then you can merge all changes to the main branch .
Use these commands

- Create a new branch
- Check how many branches you have
- Switch to the new branch
- Make any changes in your project
- Commit
- Merge to the main branch
- You can delete the new branch after you are done using it

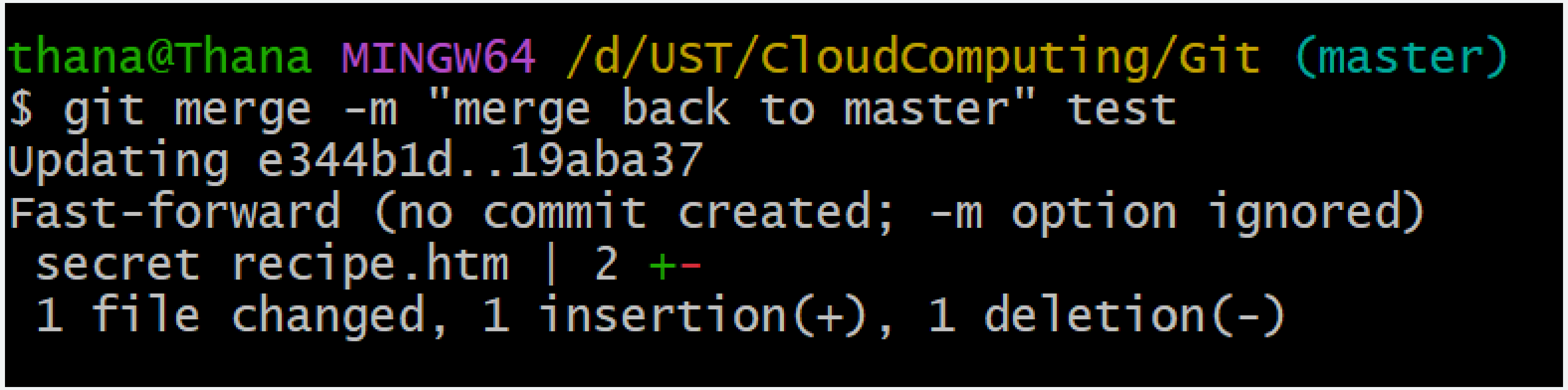
Creating Another Branch



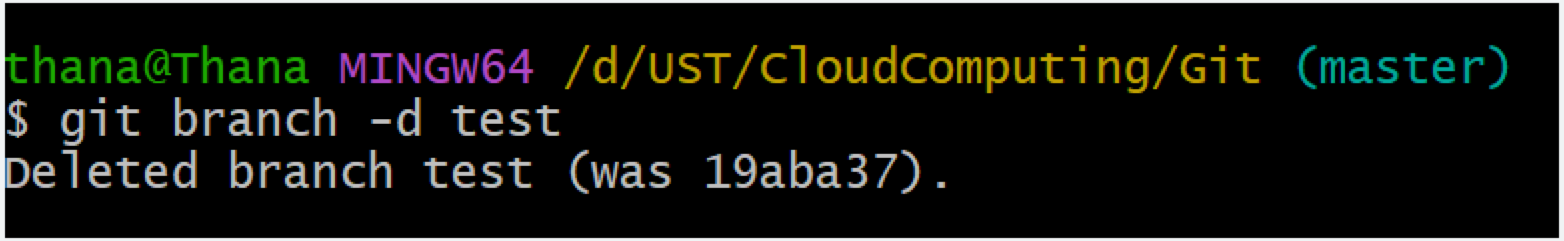
```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git branch test

thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git branch
* master
  test

thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git switch test
Switched to branch 'test'

thana@Thana MINGW64 /d/UST/CloudComputing/Git (test)
$ git branch
  master
* test

thana@Thana MINGW64 /d/UST/CloudComputing/Git (test)
$
```

02

Merge and Delete



```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git merge -m "merge back to master" test
Updating e344b1d..19aba37
Fast-forward (no commit created; -m option ignored)
 secret recipe.htm | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```



```
thana@Thana MINGW64 /d/UST/CloudComputing/Git (master)
$ git branch -d test
Deleted branch test (was 19aba37).
```

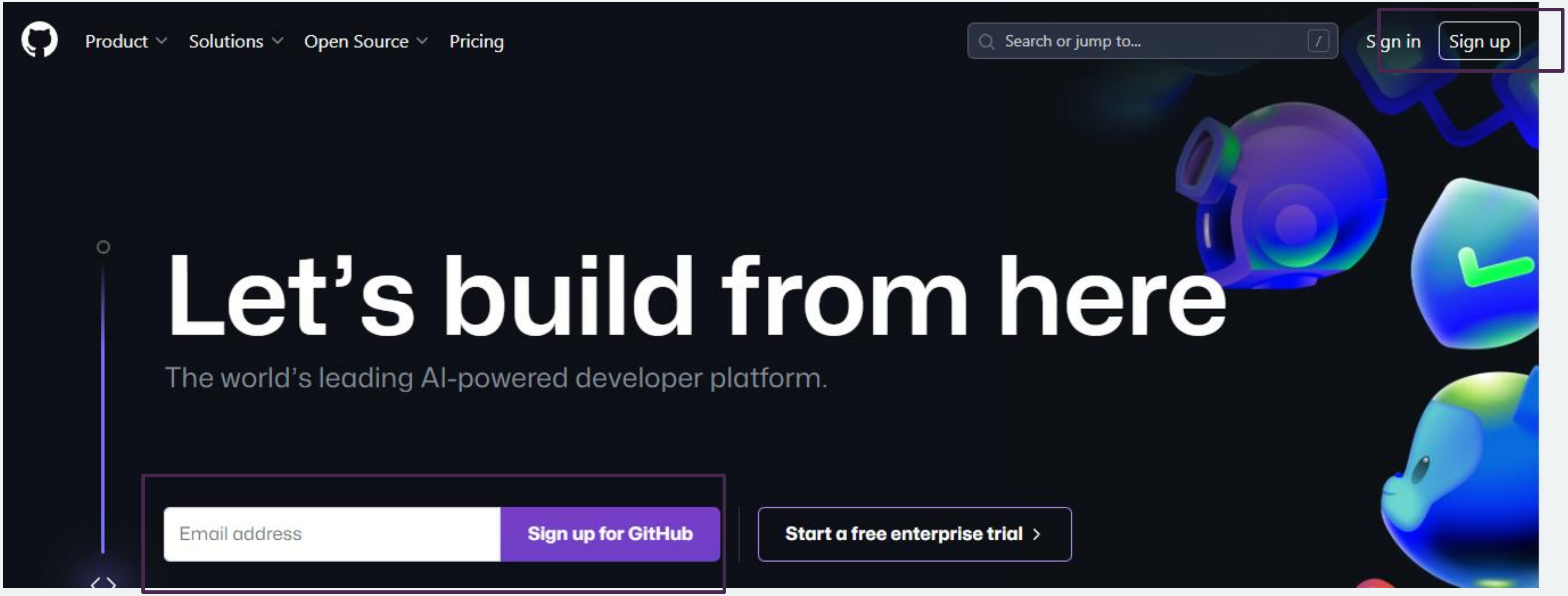
03

GitHub

- What is GitHub
- How to use GitHub

03

Create a GitHub Account

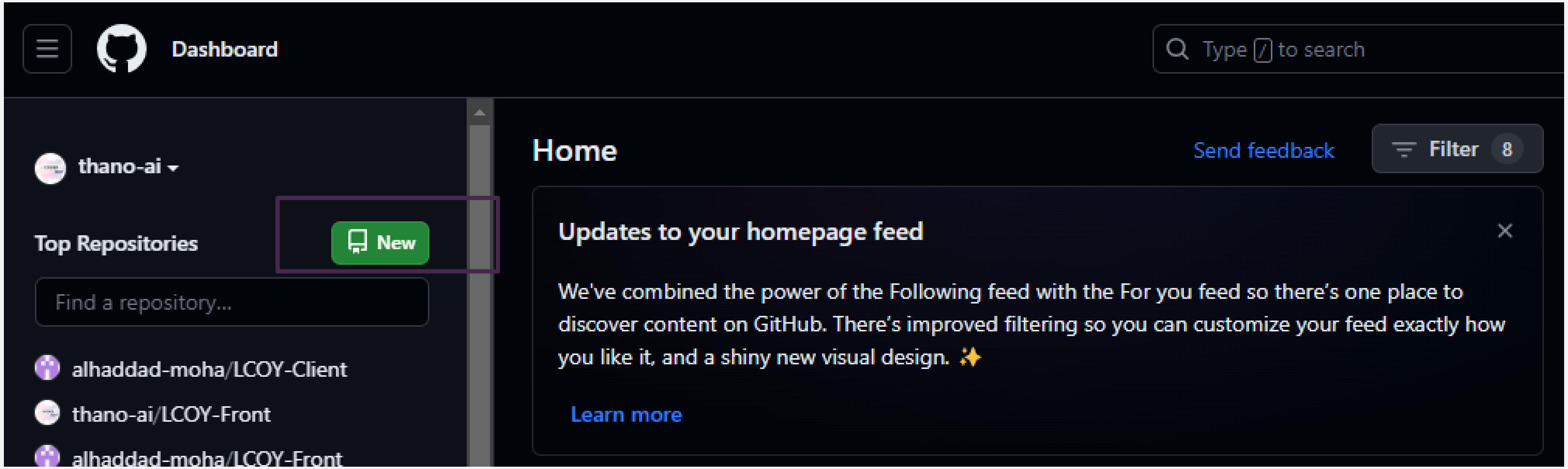


Let's build from here

The world's leading AI-powered developer platform.

03

Create a GitHub Repository



Updates to your homepage feed

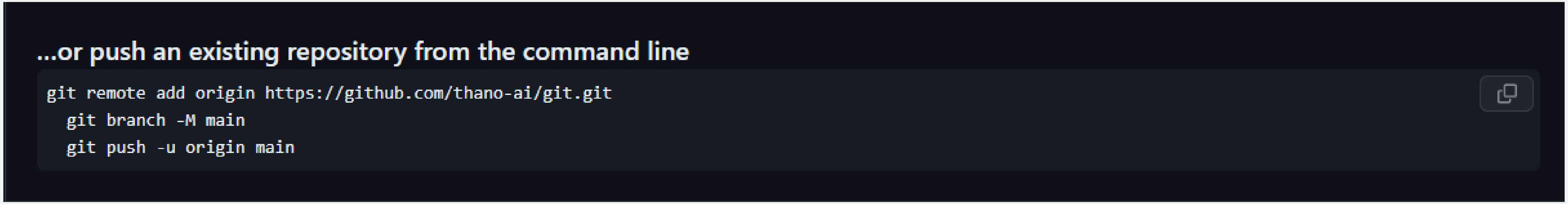
We've combined the power of the Following feed with the For you feed so there's one place to discover content on GitHub. There's improved filtering so you can customize your feed exactly how you like it, and a shiny new visual design. ✨

Learn more

Then follow the instruction to create a repository.

03

Create a GitHub Repository



…or push an existing repository from the command line

```
git remote add origin https://github.com/thano-ai/git.git
  git branch -M main
  git push -u origin main
```

Copy these commands to your git bash to add your exciting repository to GitHub .

Now after you have pushed you project, you can edit, delete files directly from GitHub online editor.

03

Collaboration in GitHub

GitHub also have a collaboration space that you can use, you can add as many people as you want to collaborate with.

- Open your GitHub account
- Choose setting from home taps
- From side navbar, choose collaboration
- Click add people to add anyone to your project
- Add their email address and click send invitation.

GitHub with Visual Studio Code

- First you must have a GitHub account
- Next you must have a new version of VSC
- Then follow the following steps

03

Steps to Publish on GitHub

- Open any project folder
- Click on version control icon
- Choose initialize repository
- Write a commit message then click on commit or press ctrl+enter
- Go to settings icon in the down left side
- Choose commands palette
- Type publish to GitHub
- Authorize your account
- After it finished publishing, click open on GitHub
- You will see your project has been published.

03

Steps to Push on GitHub

- Change anything in your previous project
- Click on version control icon again
- Write a message then commit changes
- After committing, you can either click on Sync Changes button, or you can click on setting three dots then click push.
- You have to know that Sync Changes button will push and pull changes, but push option will only push changes.

Note: Click any file then go down to Time Line to check the time line of the selected file

03

Steps to Clone a GitHub Repository

- Go to GitHub
- Search for any project you want to clone
- Copy the URL of the repository
- Go back to VSC
- Click on file icon
- Choose clone repository
- Paste the URL that you have copied before
- Select choose repository location and save it on your machine

Note: Make sure to delete the .git folder form the project folder that you have cloned.

Quiz Time

Thanks!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**

«What is DevOps and what are its benefits»

—Go to Answer

—Return

«Name 4 of DevOps tools»

—Go to Answer

—Return

«What is the git command that commit changes»

—Go to Answer

—Return

«What are the steps to clone a GitHub repository to VSC»

—Go to Answer

—Return