

Git Tutorial

Dr. Christopher Summa

Git

- Version control system
- Powerful, supports efficient multi-developer collaboration, branching, rollback
- Every clone of a repository is an “equal status”
- Supports cloud-based online repositories such as GitHub, GitLab, etc.

Git – first steps

Step 1: Make sure you have git on your system!

open up a shell and type:

```
which git
```

If you get a response, **you're good to go!** Otherwise....

Windows: git-scm.com (downloads will also install gitbash, a shell)

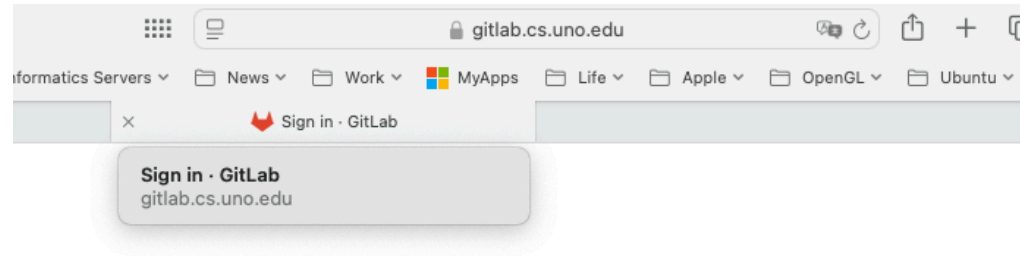
MacOS: Xcode (install command-line developer tools)

Linux: use `apt` or whatever package manager is native to your system

Git – first steps

Step 2: Create a repository

A) We'll work with UNO's internal gitlab for now



GitLab Community Edition

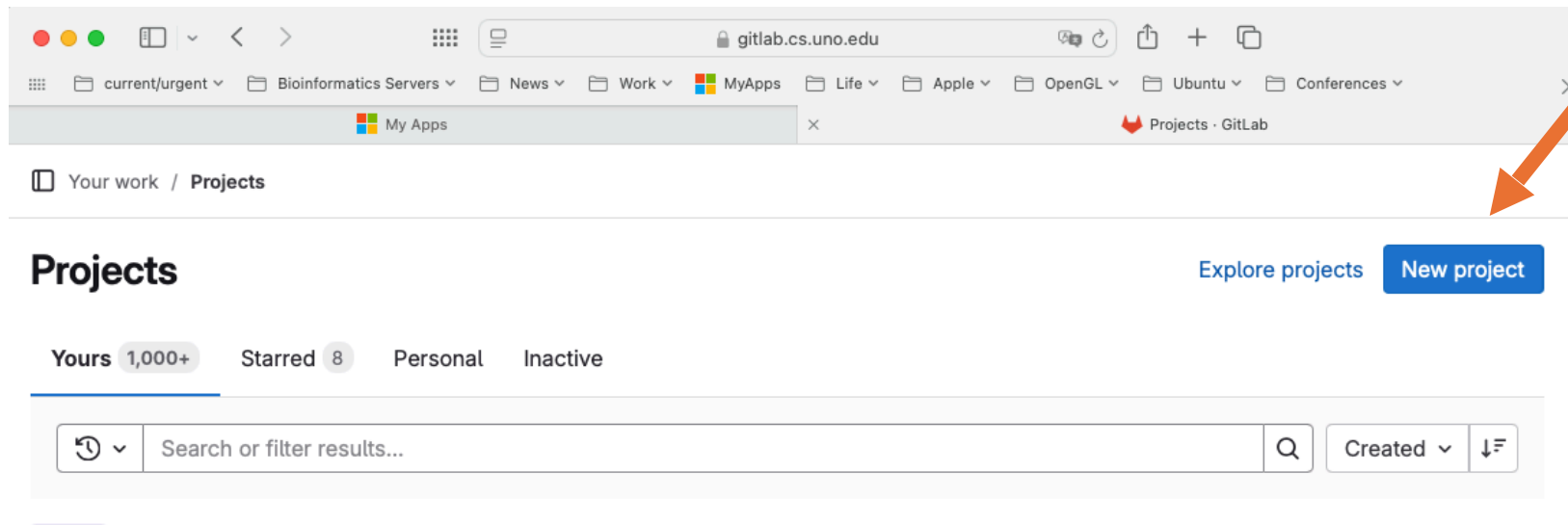
UNO Active Directory login

☐ Remember me

Git – first steps

Step 2: Create a repository

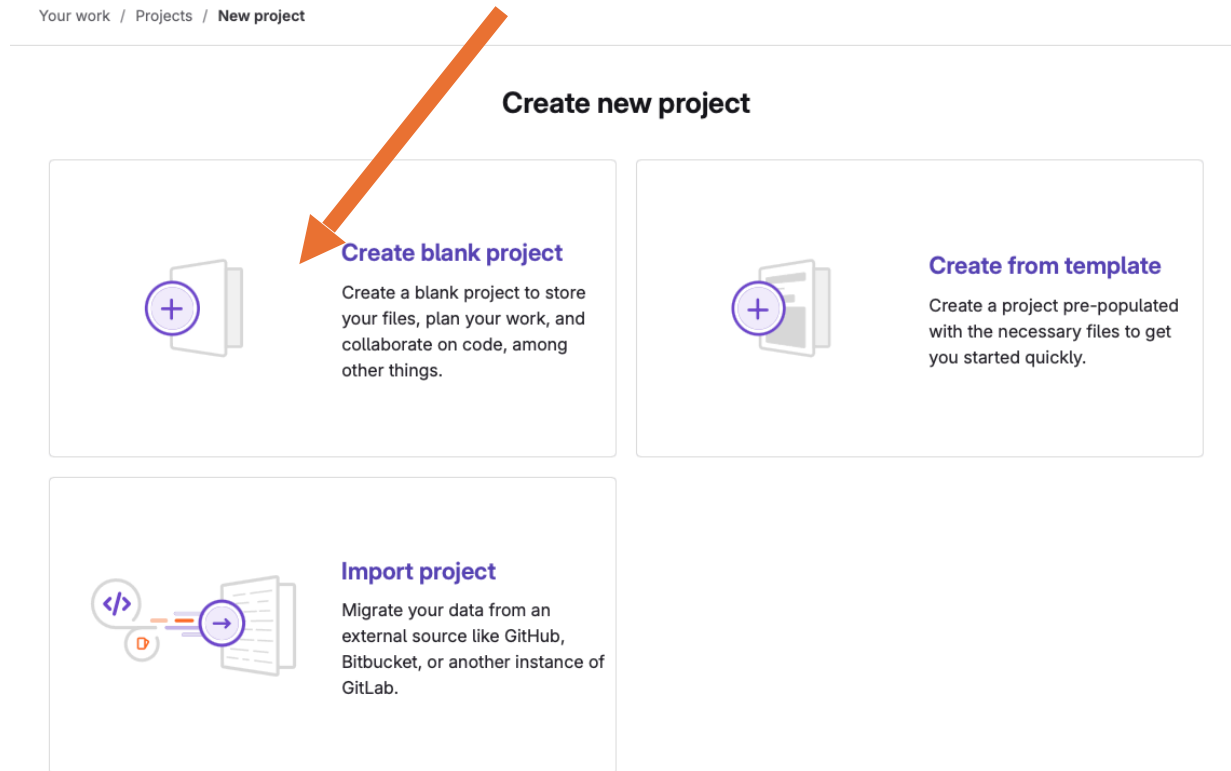
B) Once logged in, create a new repository – click on the blue “New Project” button:



Git – first steps

Step 2: Create a repository

C) Here you have a few choices – let's create a "blank" empty repository

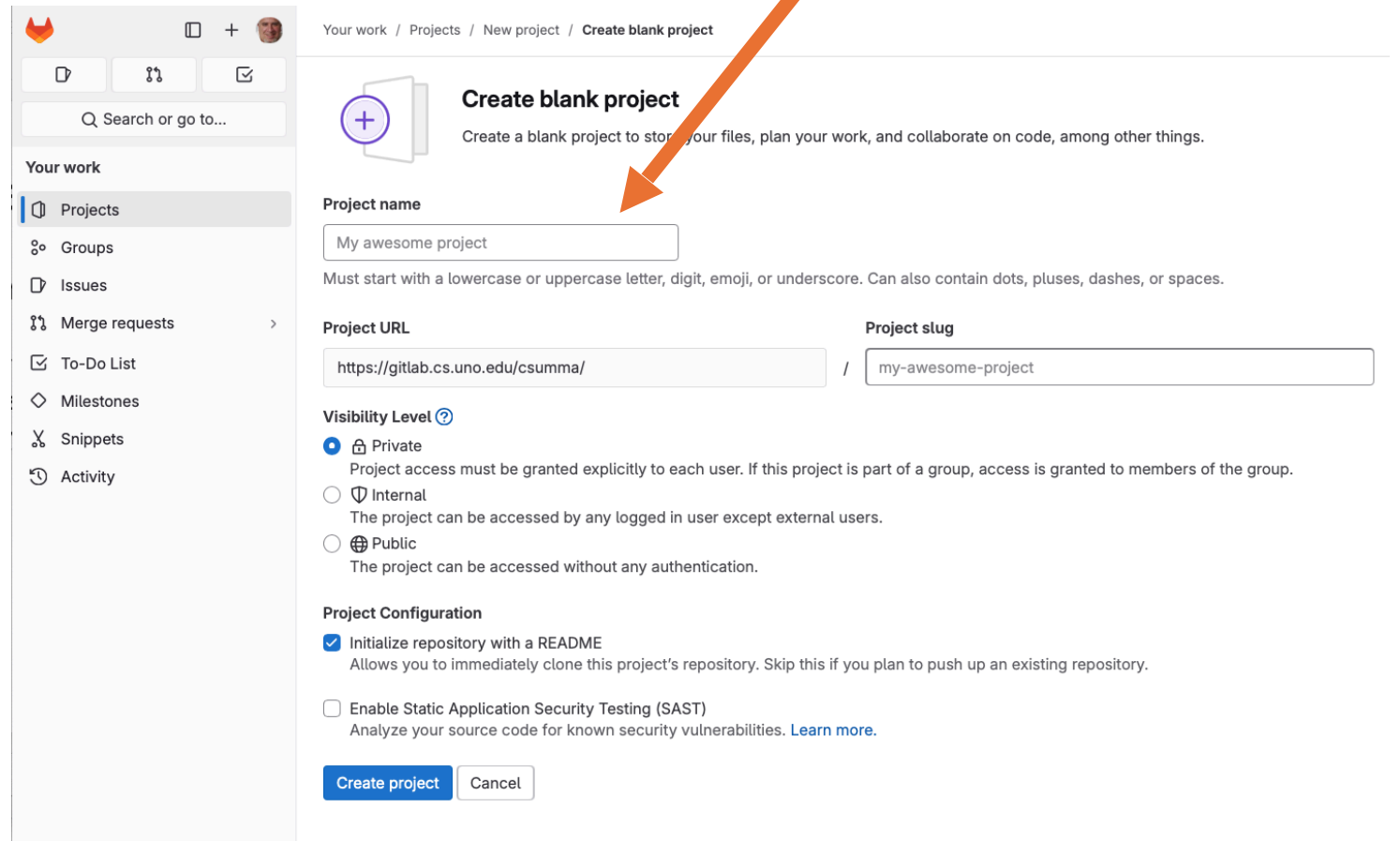


You can also create a project from the command line. [Show command](#)

Git – first steps

Step 2: Create a repository

D) It's here that we can give our repository a name. **NOTE** that everything in this repository, when on your computer, will live in a directory with the same name. Also, since the terminal **REALLY** doesn't like spaces in directory (folder) names, instead of spaces always use the underscore character



The screenshot shows the GitLab web interface for creating a new project. On the left is a sidebar with navigation links: 'Your work' (selected), 'Groups', 'Issues', 'Merge requests', 'To-Do List', 'Milestones', 'Snippets', and 'Activity'. The main content area is titled 'Create blank project' and includes the following fields and options:

- Project name:** A text input field containing 'My awesome project'. An orange arrow points to this field.
- Project URL:** A text input field containing 'https://gitlab.cs.uno.edu/csumma/'.
- Project slug:** A text input field containing 'my-awesome-project'.
- Visibility Level:** Three radio button options: 'Private' (selected), 'Internal', and 'Public'.
- Project Configuration:** Two checkboxes: 'Initialize repository with a README' (checked) and 'Enable Static Application Security Testing (SAST)' (unchecked).

At the bottom of the form are two buttons: 'Create project' and 'Cancel'.

Git – first steps

Step 2: Create a repository

E) Once you've created the project you'll be brought to that project's page and it should look like this (just hit the x in the light blue notification boxes at the top of the page to dismiss them)

The screenshot displays the GitLab web interface for a newly created repository named "test_proejct" by user Christopher M Summa. The interface is divided into three main sections: a left sidebar, a central content area, and a right sidebar.

Left Sidebar: Contains navigation links for Project, Issues, Merge requests, Manage, Plan, Code, Build, Secure, Deploy, Operate, Monitor, Analyze, and Settings. The "Project" section is currently selected.

Central Content Area: Displays the repository details for "test_proejct". It shows the "main" branch selected, with options to view the History, Find file, Edit, or Code. Below this, it lists the "Initial commit" by Christopher M Summa, authored 2 minutes ago, with a commit hash of e1b989c4. A table below the commit shows the file "README.md" with the "Initial commit" and "2 minutes ago" update. The "README.md" file content is displayed, including a "Getting started" section with instructions on how to get started with GitLab and a section for "Add your files".

Right Sidebar: Contains "Project information" and "Project settings". The "Project information" section shows 1 Commit, 1 Branch, 0 Tags, and 3 KiB Project Storage. The "Project settings" section includes links for README, Auto DevOps enabled, Add LICENSE, Add CHANGELOG, Add CONTRIBUTING, Add Kubernetes cluster, Add Wiki, and Configure Integrations. The "Created on" date is January 23, 2025.

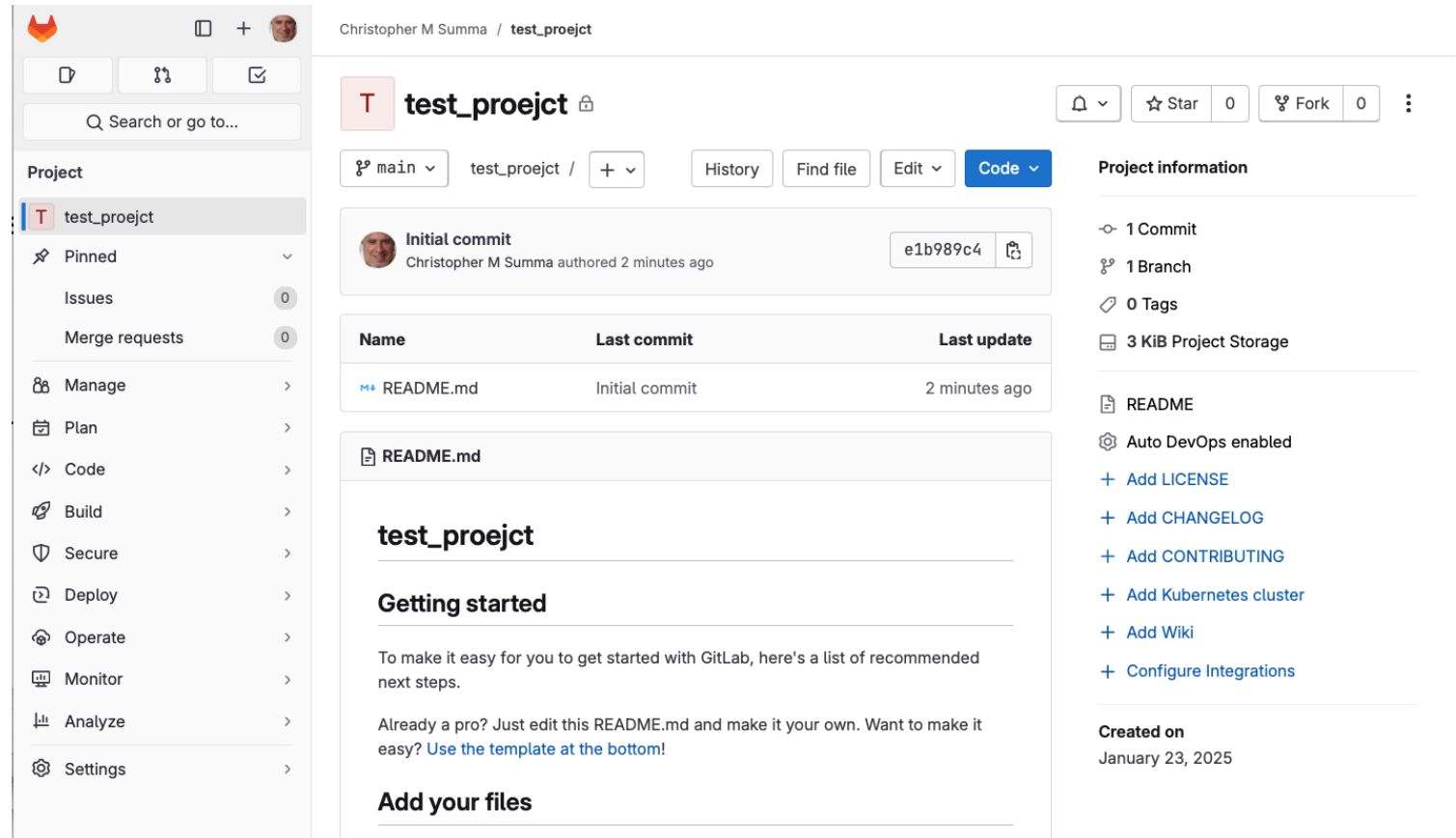
Git – first steps

Step 2: Create a repository

F) Next, let's make sure we've got our ssh keys set up. This is something you'll only need to do once per computer you plan to use to interact with this git server.

Click on the little icon that looks like a circular mandala at the red arrow, which is where your “user setting stuff” is found and choose “Preferences”

(you may have to expand the browser window to see this contextual menu)



The screenshot shows the GitLab web interface for a new repository named 'test_proejct'. An orange arrow points from the text '(you may have to expand the browser window to see this contextual menu)' to a small circular icon in the top navigation bar. The interface includes a sidebar with navigation options like 'Project', 'Pinned', 'Issues', 'Merge requests', 'Manage', 'Plan', 'Code', 'Build', 'Secure', 'Deploy', 'Operate', 'Monitor', 'Analyze', and 'Settings'. The main content area shows the repository details, including the initial commit by Christopher M Summa, a table of files (README.md), and project information such as 1 commit, 1 branch, 0 tags, and 3 KiB project storage. The 'Getting started' section provides instructions for new users and links to templates.

Christopher M Summa / test_proejct

test_proejct

main test_proejct / + History Find file Edit Code

Initial commit
Christopher M Summa authored 2 minutes ago e1b989c4

Name	Last commit	Last update
README.md	Initial commit	2 minutes ago

README.md

test_proejct

Getting started

To make it easy for you to get started with GitLab, here's a list of recommended next steps.

Already a pro? Just edit this README.md and make it your own. Want to make it easy? [Use the template at the bottom!](#)

Add your files

Project information

- 1 Commit
- 1 Branch
- 0 Tags
- 3 KiB Project Storage

README

- Auto DevOps enabled
- [Add LICENSE](#)
- [Add CHANGELOG](#)
- [Add CONTRIBUTING](#)
- [Add Kubernetes cluster](#)
- [Add Wiki](#)
- [Configure Integrations](#)

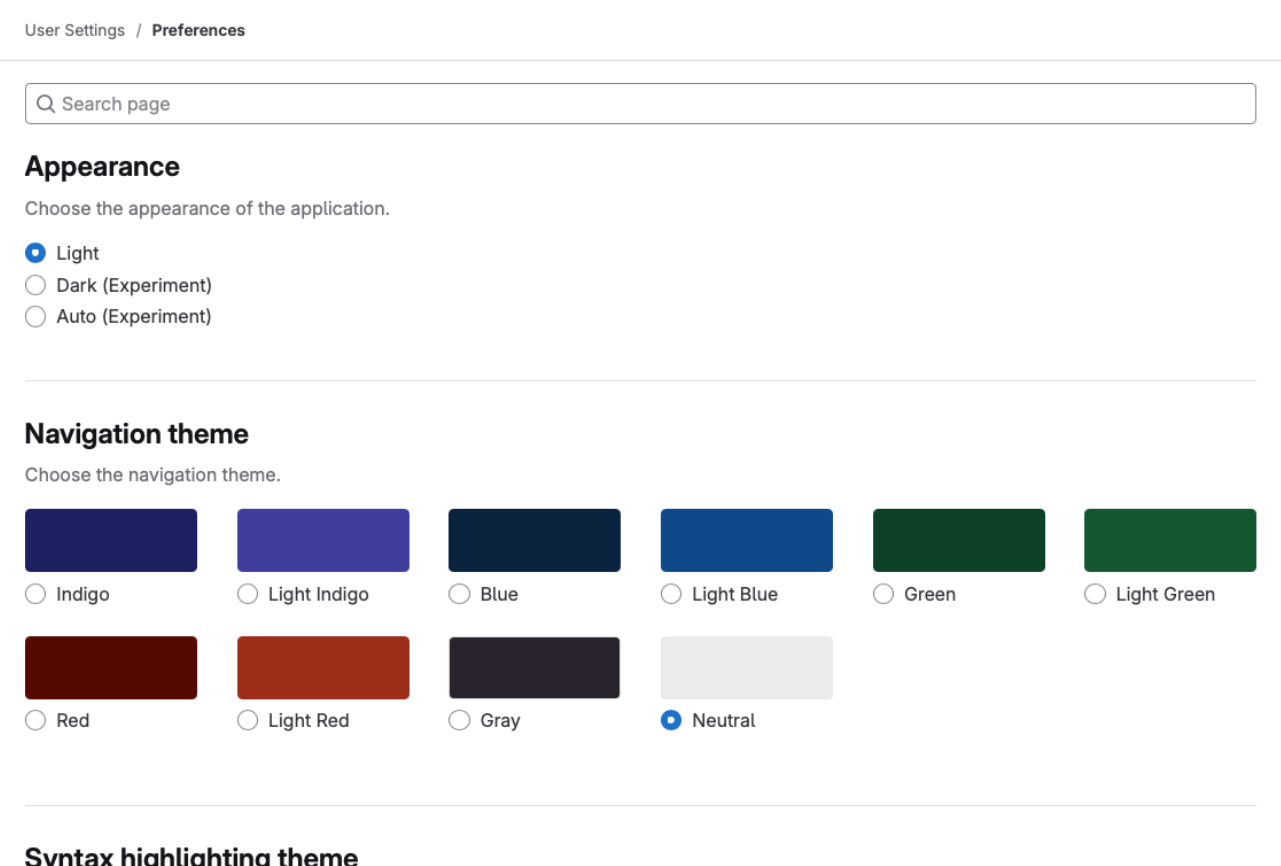
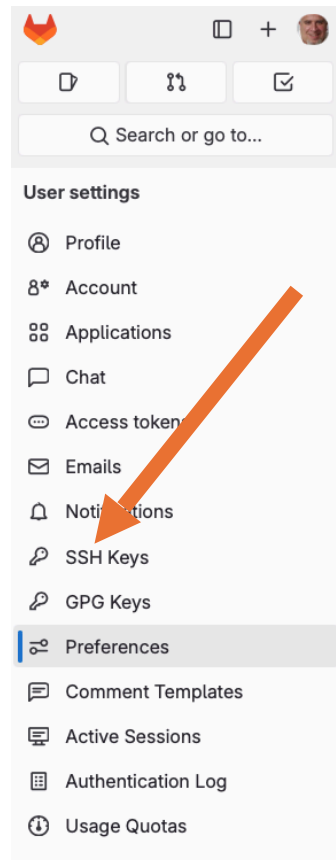
Created on
January 23, 2025

Git – first steps

Step 2: Create a repository

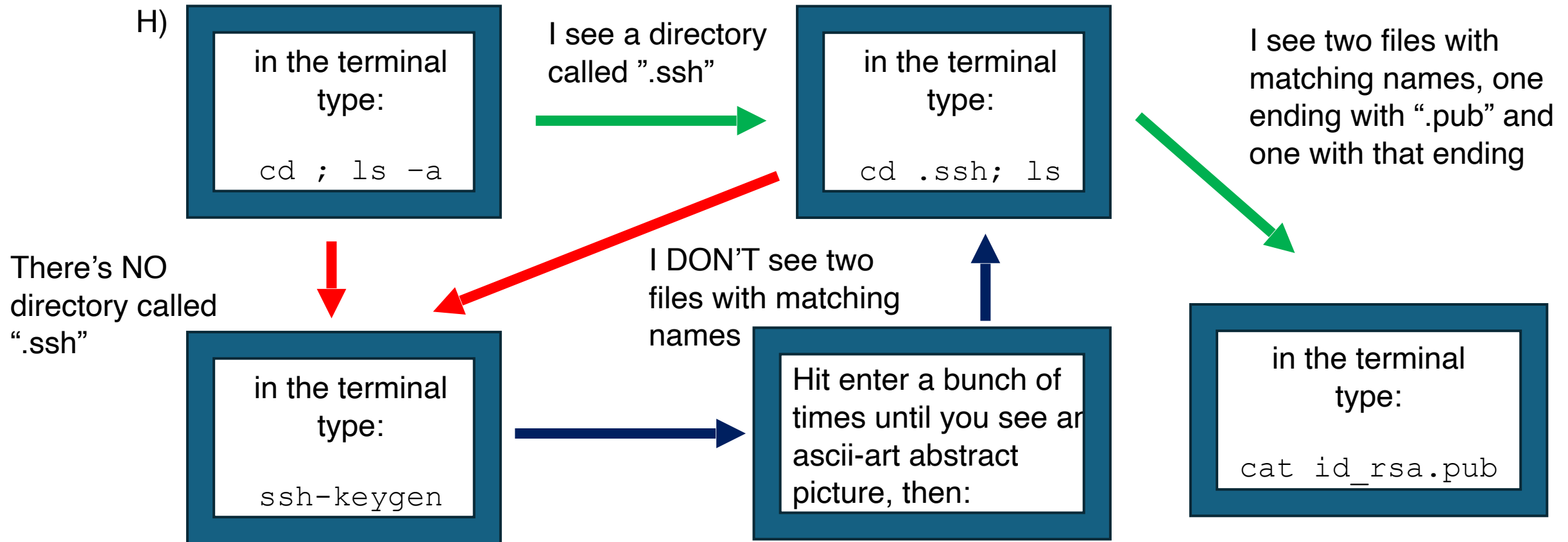
G) in the contextual menu on the left, just past midway down you'll see "SSH Keys". Click on this.

If you see an ssh key representing you at the machine you're currently working on you can skip the next bit and move on to **Step 3**, if not, open up a terminal window...



Git – first steps

Step 2: Create a repository



Git – first steps

Step 2: Create a repository

1) Copy the output (if using gitbash you'll need to right-click copy) then go back to the website and click “Add new key” in the upper right.

Paste into the Key entry box, change the Title if you'd like and clear out the Expiration Date, then click “Add Key” at the bottom

User Settings / SSH Keys

Search settings

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab. Check the [current instance configuration](#).

Your SSH keys 8

Add an SSH key

Add an SSH key for secure access to GitLab. [Learn more](#).

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

Title

Example: MacBook key

Key titles are publicly visible.

Usage type

Authentication & Signing

Expiration date

2026-01-23



Optional but recommended. If set, key becomes invalid on the specified date.

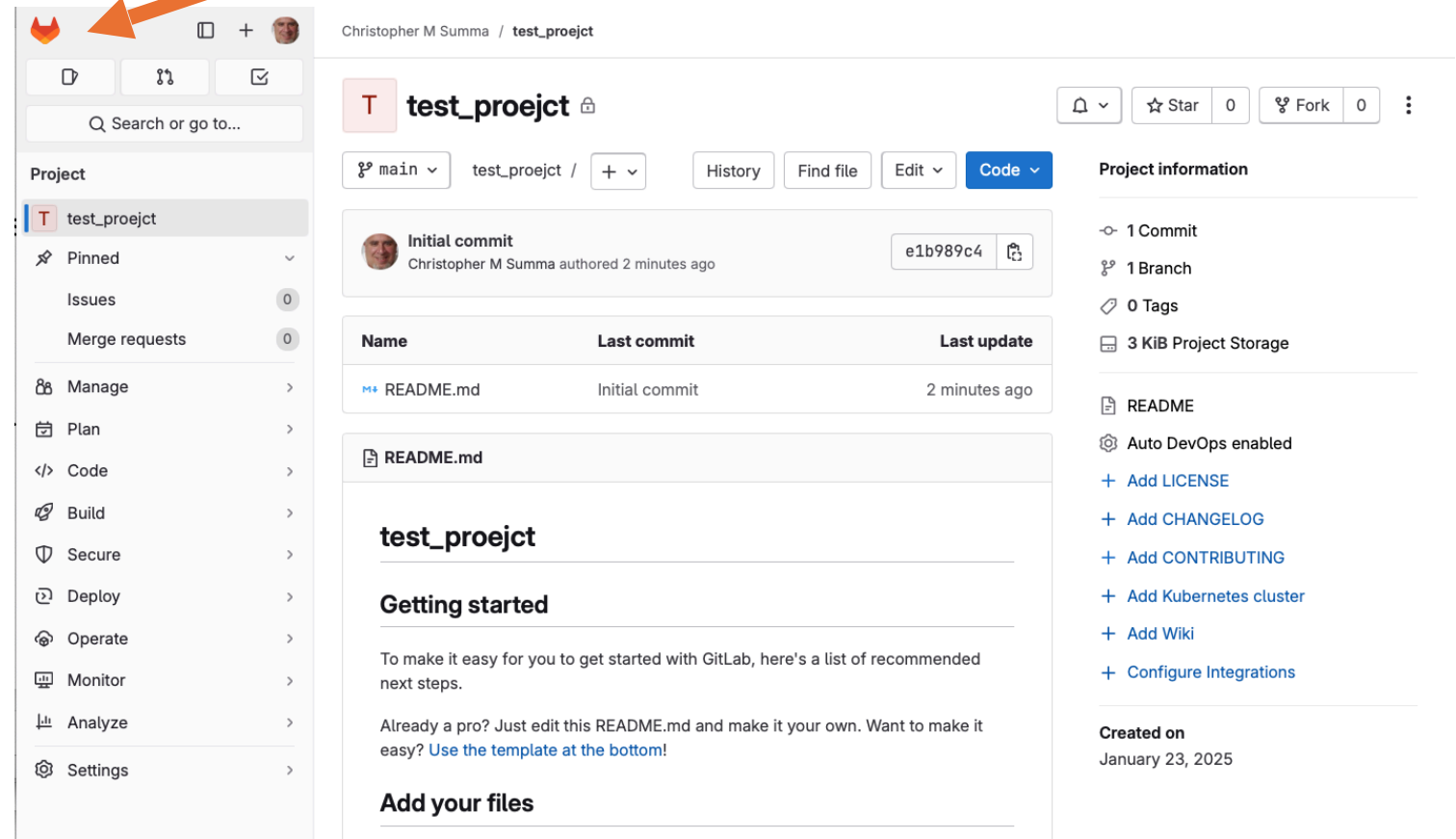
Add key

Cancel

Git – first steps

Step 3: Clone your repository

A) Go back to your “home” by clicking on the fox icon in the upper left. You should see your new repository listed in the window on the right. Click on your repo.

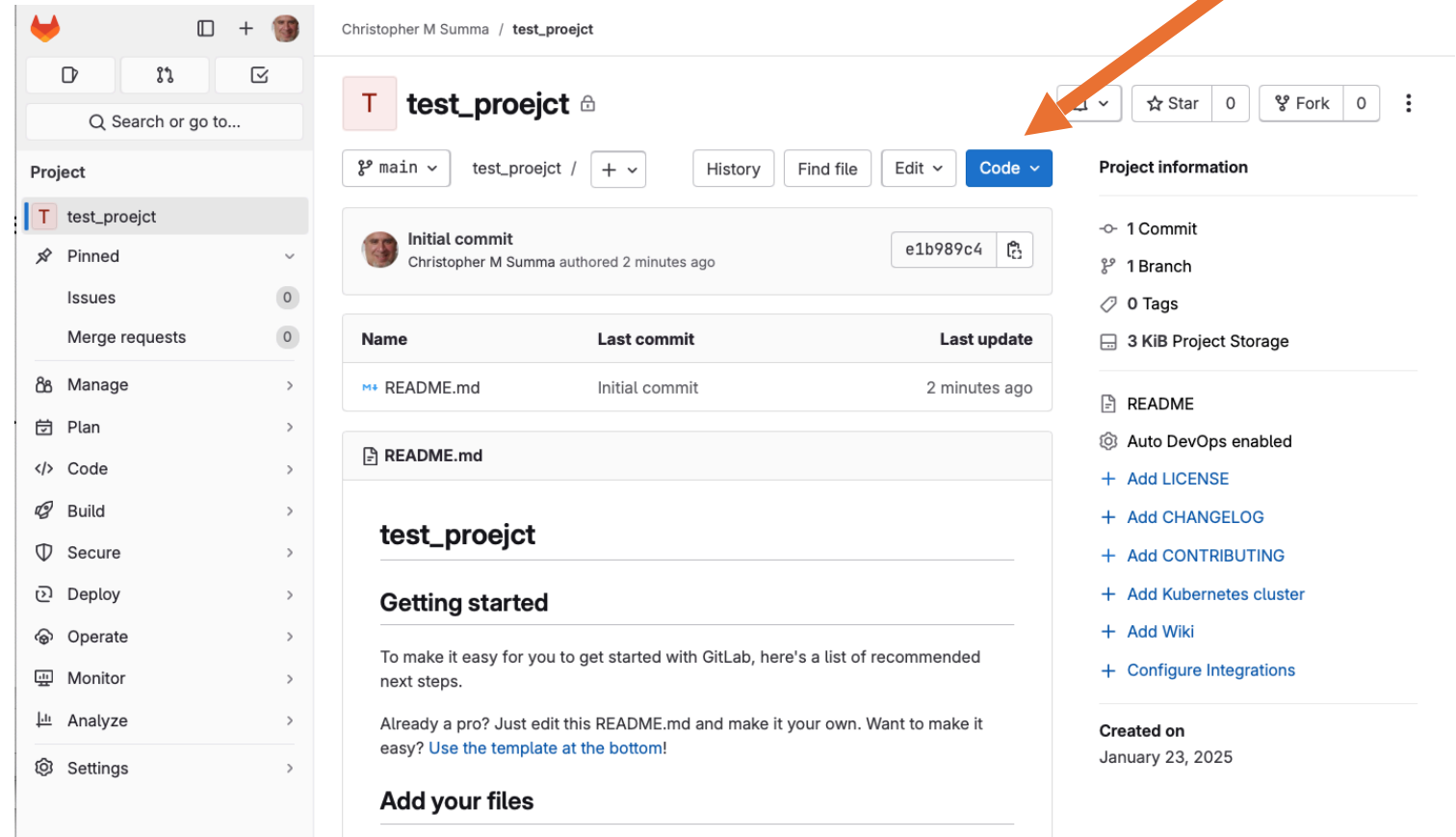
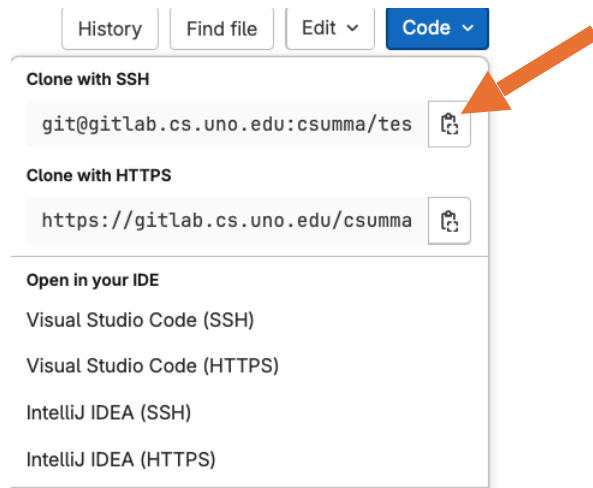


The screenshot shows the GitLab web interface for a repository named 'test_proejct' by user 'Christopher M Summa'. An orange arrow points to the fox icon in the top left of the sidebar. The sidebar on the left contains a search bar and a list of project items: 'test_proejct' (selected), 'Pinned', 'Issues' (0), 'Merge requests' (0), 'Manage', 'Plan', 'Code', 'Build', 'Secure', 'Deploy', 'Operate', 'Monitor', 'Analyze', and 'Settings'. The main content area displays the repository details, including the 'Initial commit' by Christopher M Summa 2 minutes ago with commit hash 'e1b989c4'. Below this is a table with one row: 'README.md' with the last commit 'Initial commit' 2 minutes ago. The 'README.md' content is visible, showing a 'Getting started' section with instructions on how to get started with GitLab. On the right side, there is a 'Project information' section showing 1 Commit, 1 Branch, 0 Tags, and 3 KiB Project Storage. Below this are links to 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Add Kubernetes cluster', 'Add Wiki', and 'Configure Integrations'. At the bottom right, it says 'Created on January 23, 2025'.

Git – first steps

Step 3: Clone your repository

B) Click the blue “Code” button, and copy text in “Clone with SSH” by clicking on the button just to the right of text box



Git – first steps

Step 3: Clone your repository

C) Back in your terminal:

```
cd
```

if you do an `ls` and see a directory called "source" or "projects" just `cd` into it, otherwise:

```
mkdir source ; cd source
```

Once you are in your source directory:

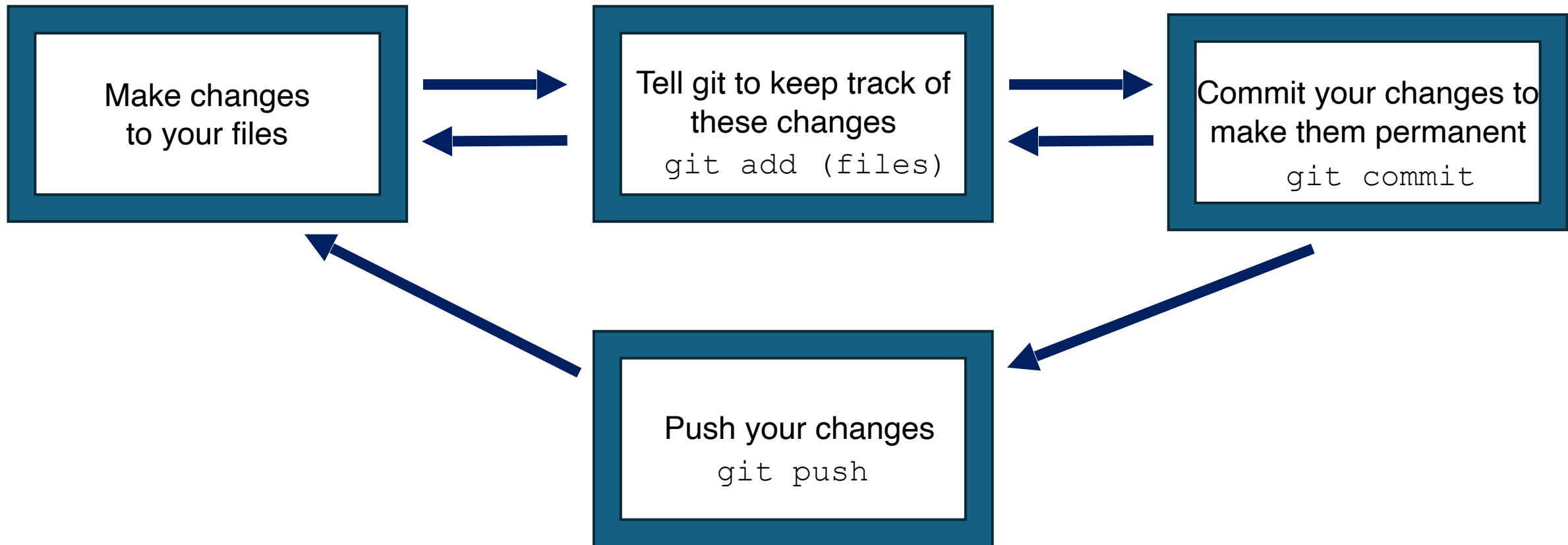
```
git clone (then paste what you've copied from the website, and hit enter)
```

This should have created a new folder, which is now under the control of git...

```
cd (directory name i.e. name of your repository)
```

Git – Basic Idea

The general workflow of working with git looks like this:



Git – Basic Idea

If you are working with others, or using multiple machines, there is an extra step at the beginning before you start to change your files:

