INTODUCTION TO GIT AND GITHUB (BASICS)


It's a git
Distributed,
Open source version control system
designed for speed and efficiency.

Version Control System

What is a VCS ?

- A system that records changes to a set of files over time. Essential for a development project with several people

Why we need it ?

- Keep every version of a file (source code, image ...)
- Work easily with several developers
- Compare changes Revert files back to a previous state...

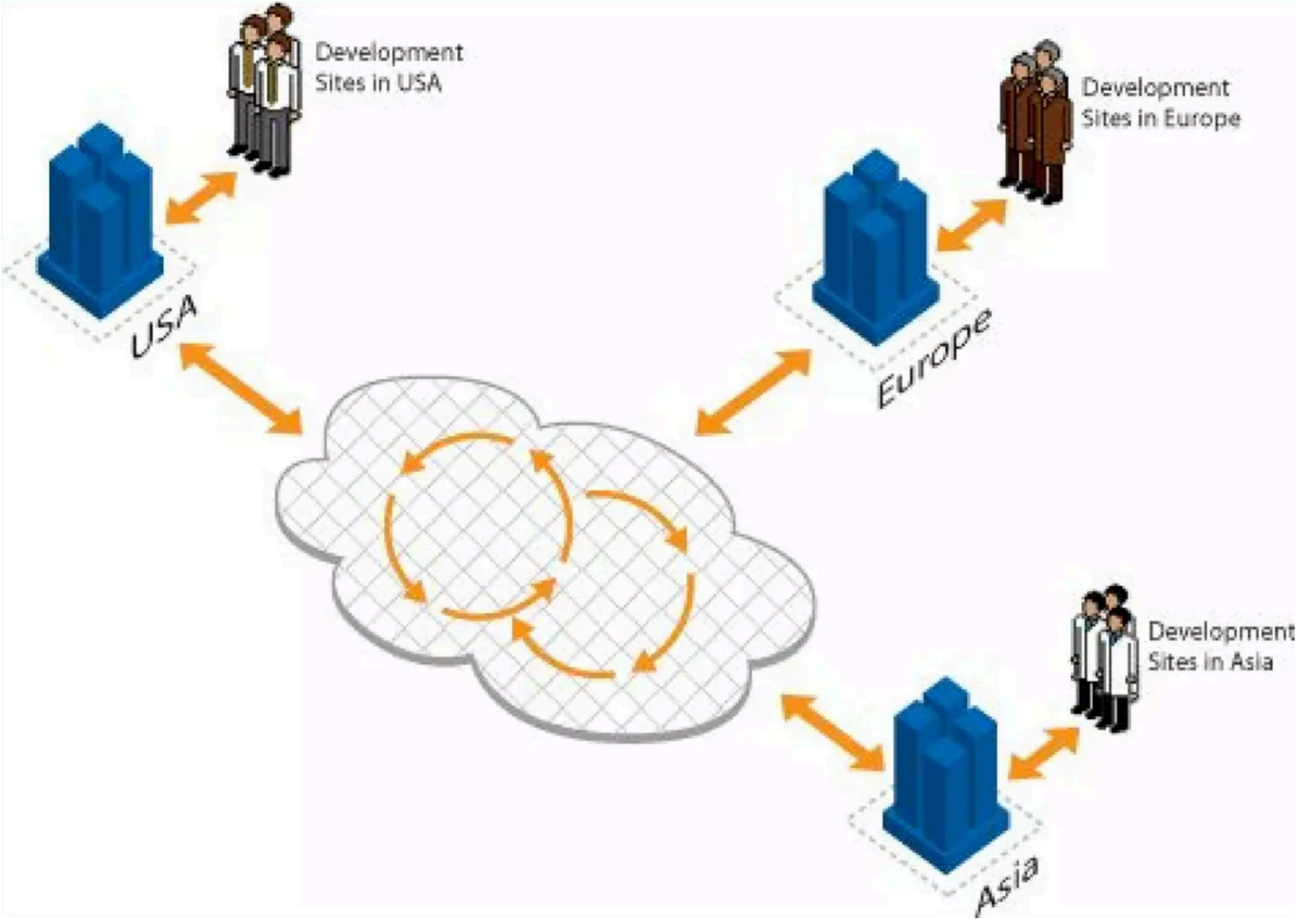
Github

- Largest web based Git repository hosting service
- Github was founded on GIT which was written by Linus Torvalds creator of LINUX
- Github is a version control and allows code collaboration with developers
- Adds extra functionality on top of Git
- It helps to add new improved changes to the software by sharing the code to other developers
- Github is used to store the souce code for a project and track the complete history of all the changes made to that code.

GitHub

THE NEED FOR GITHUB

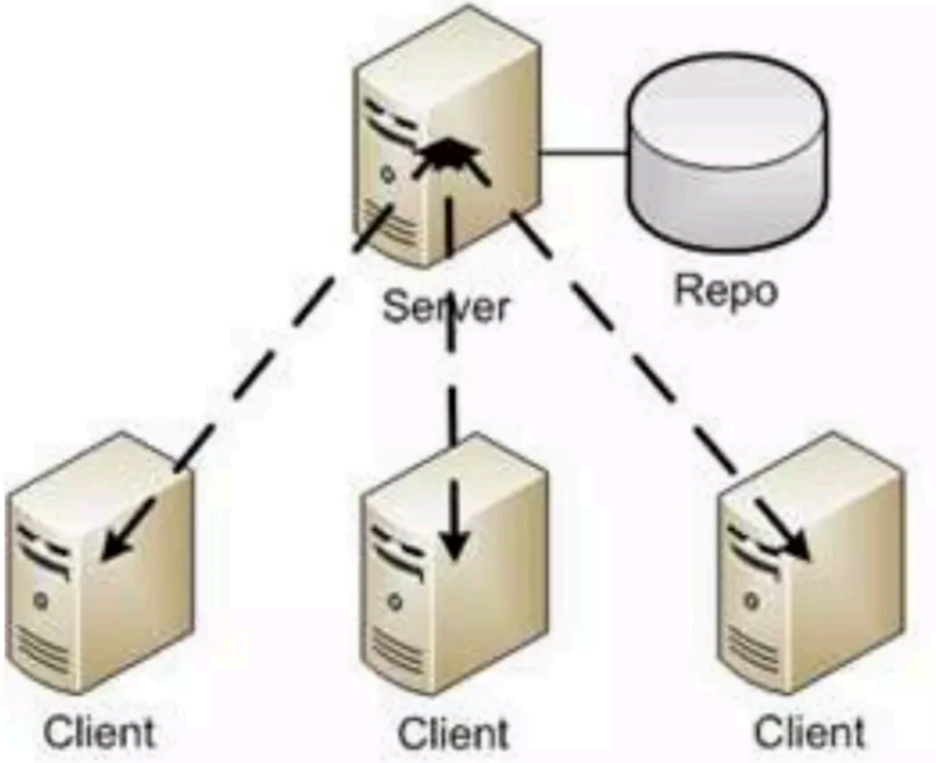
- Developers need a web/cloud based code hosting platform
- Useful for version control
- Enables effective collaboration
- Download projects and files in one go
- Easy evaluation of each other’s work


Development
Sites in USA
USA
Development
Sites in Europe
Europe
Development
Sites in Asia
Asia

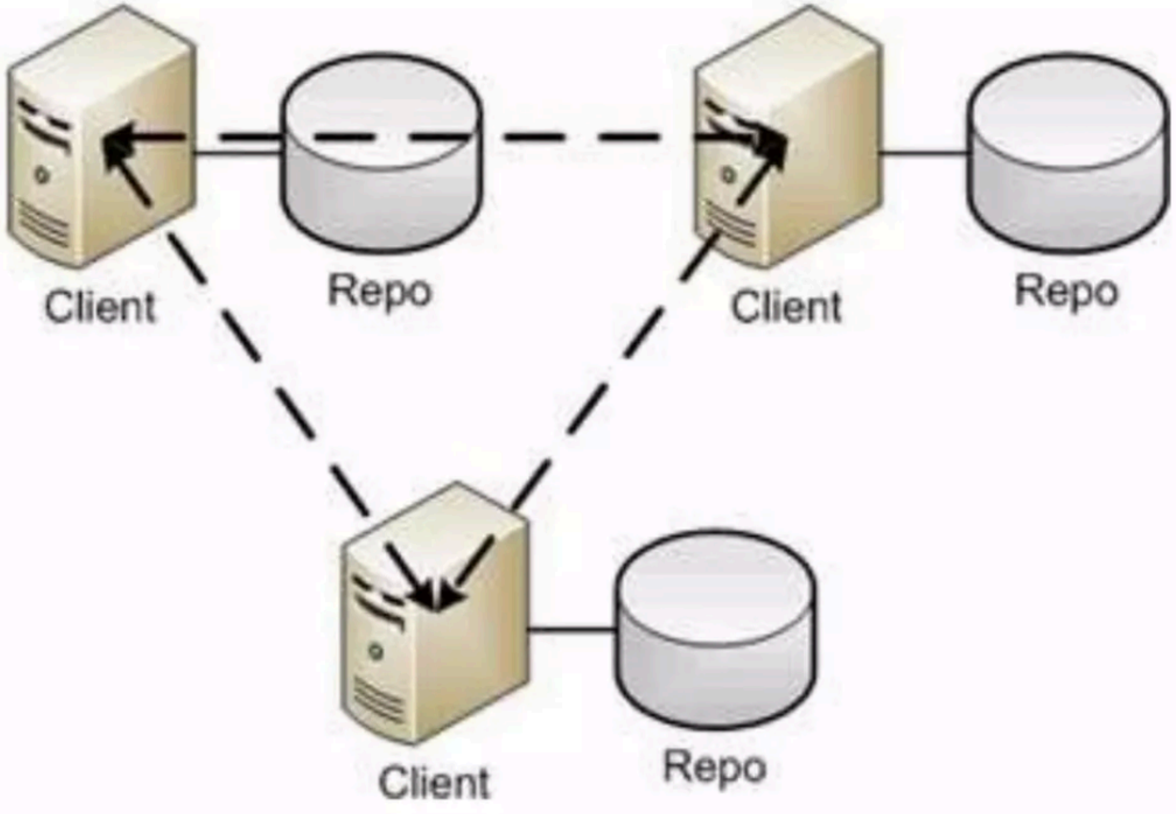
What is a repository?

- “repo” = repository
- usually used to organize a single project
- repos can contain folders and files, images, videos, spreadsheets, and data sets – anything your project needs

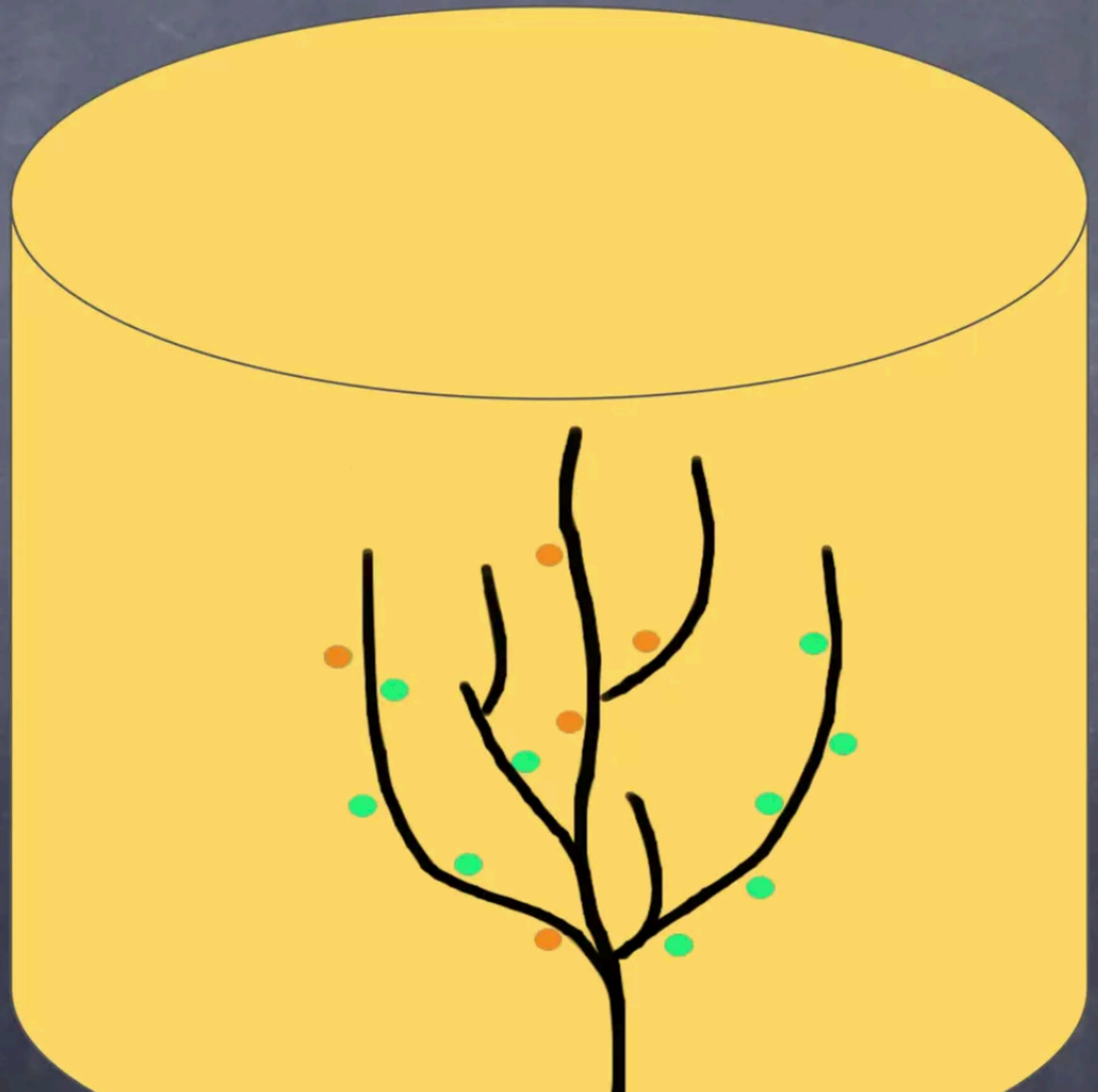
Traditional



Distributed



Git Repository



What is Branch?

-In Git, a `branch` is a new/separate version of the main repository.

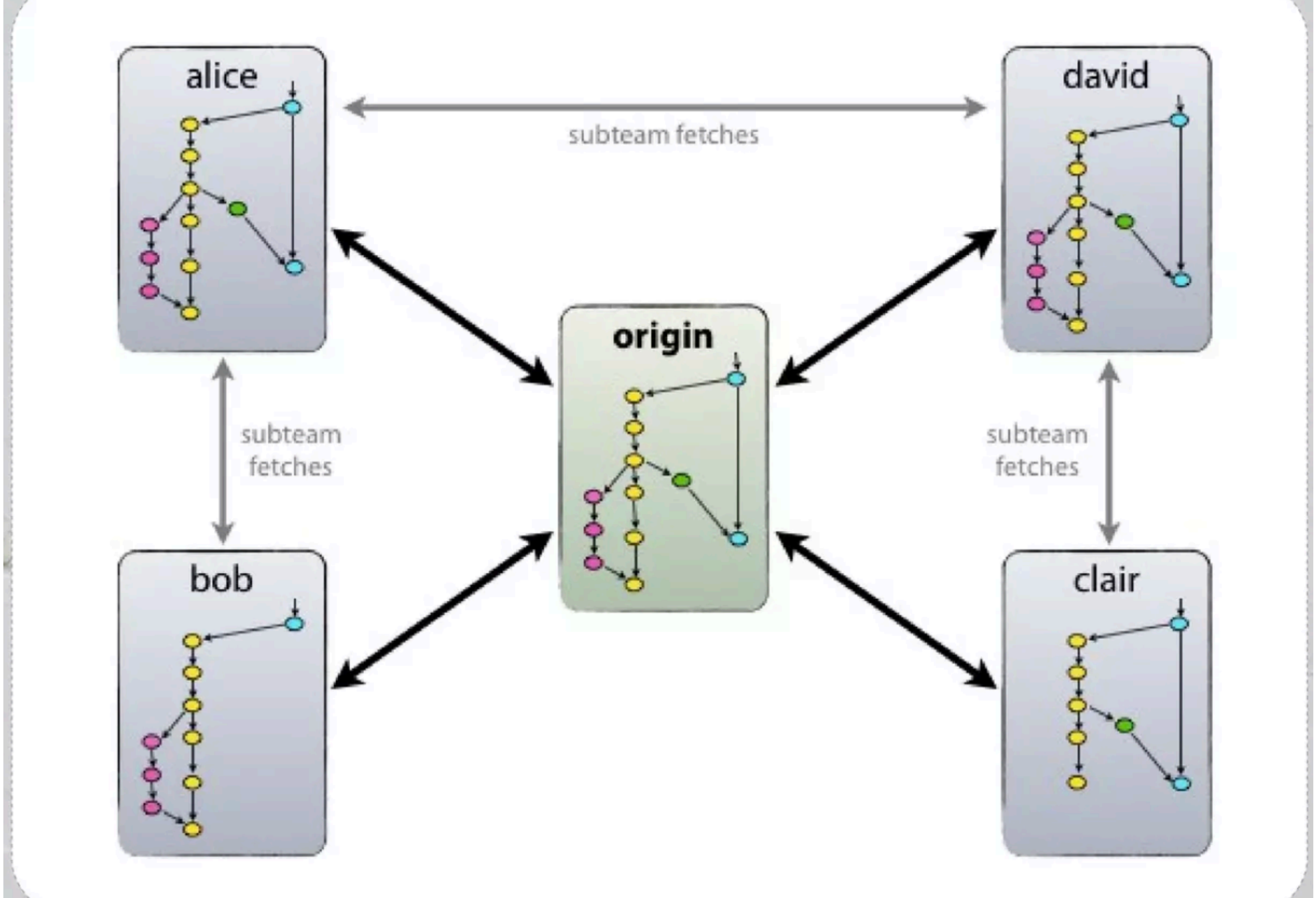
-Let’s say you have a large project, and you need to update the design on it.

-Branches allow you to work on different parts of a project without impacting the main branch.

-When the work is complete, a branch can be merged with the main project.

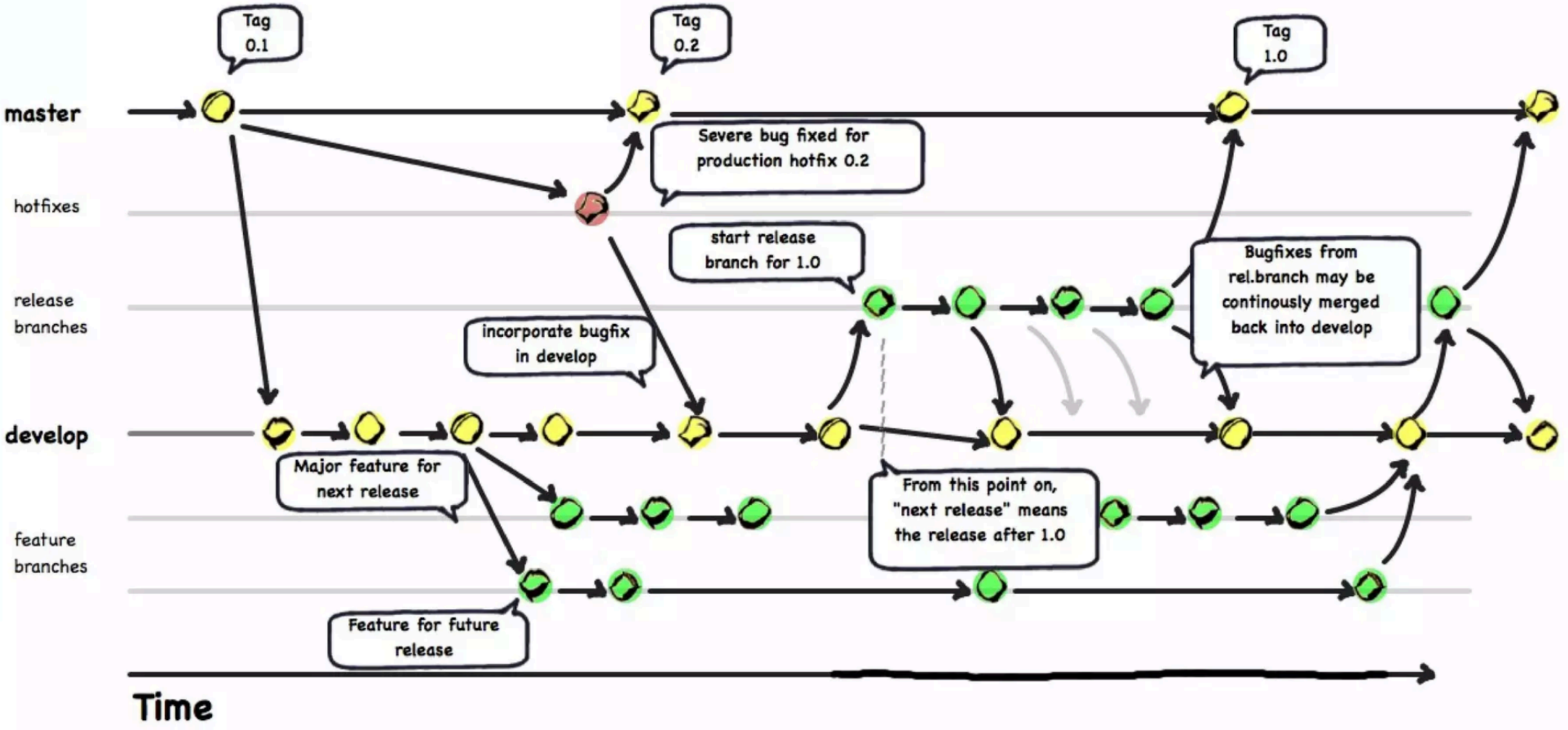
-You can even switch between branches and work on different projects without them interfering with each other.

-Branching in Git is very lightweight and fast!


alice
david
origin
bob
clair
subteam fetches
subteam fetches
subteam fetches

Branches

A successful Git branching model

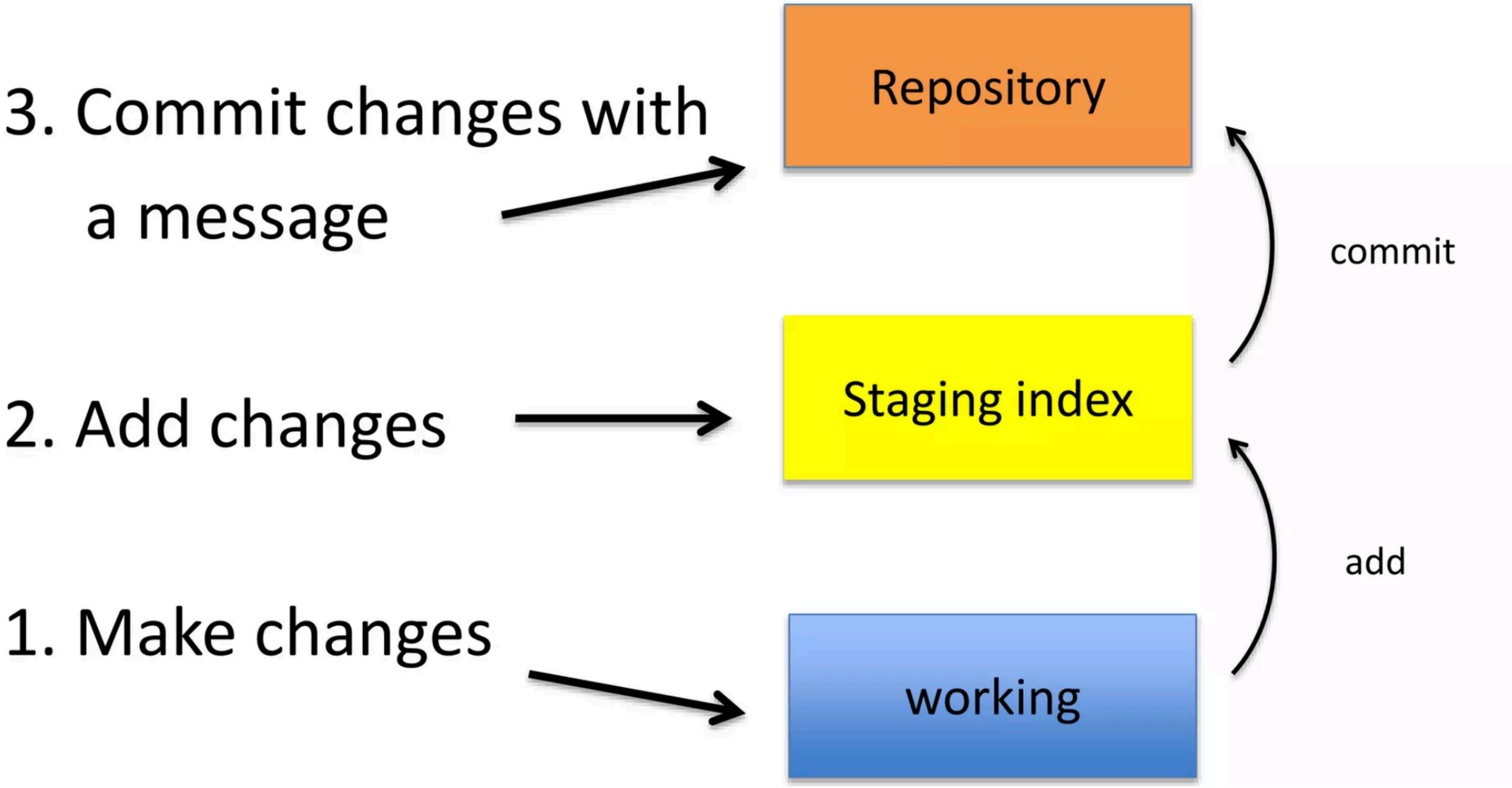

Tag 0.1
Tag 0.2
Tag 1.0
master
Severe bug fixed for production hotfix 0.2
hotfixes
start release branch for 1.0
release branches
Bugfixes from rel.branch may be continously merged back into develop
incorporate bugfix in develop
develop
Major feature for next release
From this point on, "next release" means the release after 1.0
feature branches
Feature for future release
Time

• Create a commit

A commit is a record of what files you have changed since the last commit.
Commits allow you to go back to the state of the project at any point in history.
You create a commit in order to add files to the Master.

Commit

- A **commit** object mainly contains three things:
  - A set of **changes** the **commit** introduces
  - **Commit message** describing the changes
  - A **hash**, a 40-character string that uniquely identifies the commit object

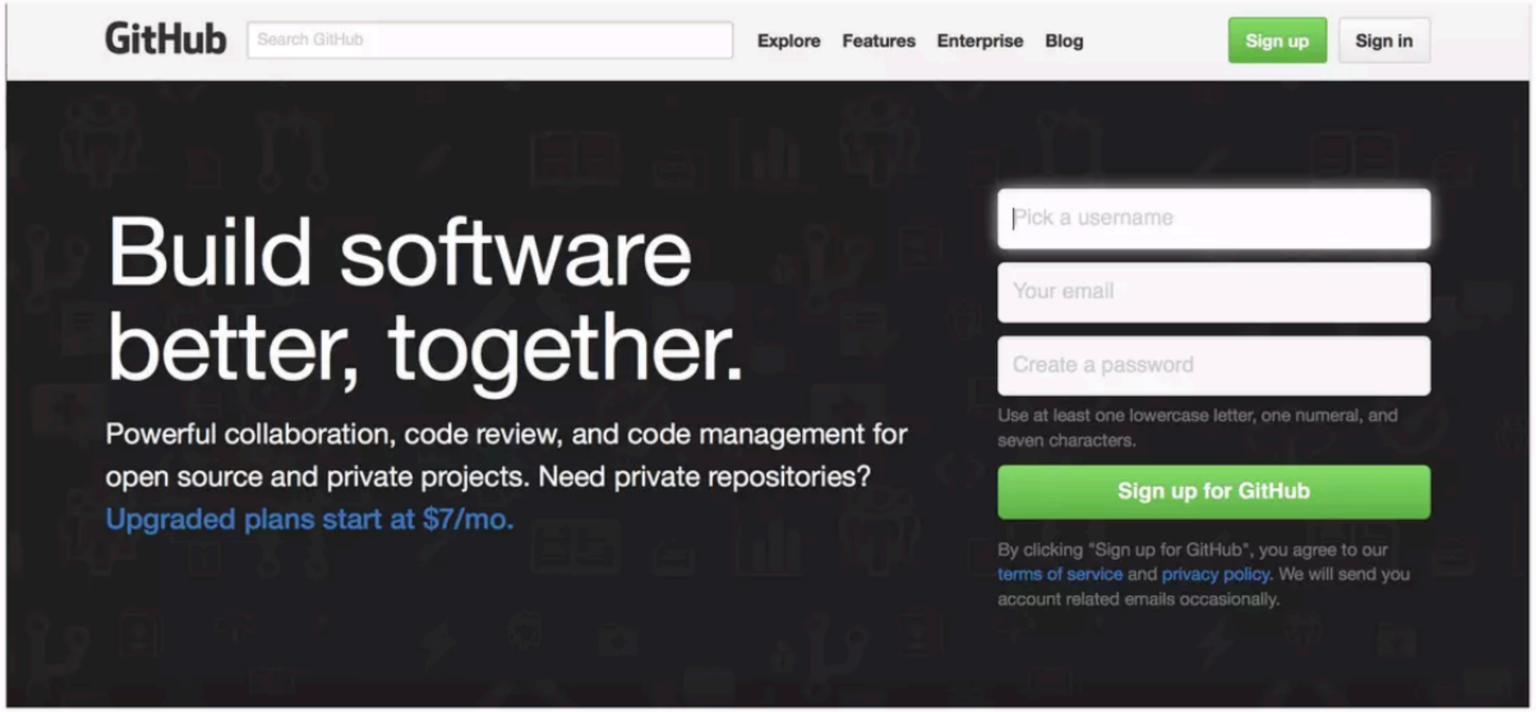

3. Commit changes with a message
Repository
commit
2. Add changes
Staging index
add
1. Make changes
working

Install git

- **Linux (Debian)**
  - Command: `sudo apt-get install git`
- **Linux (Fedora)**
  - Command: `sudo yum install git`
- **Mac**
  - http://git-scm.com/download/mac
- **Windows**
  - http://git-scm.com/download/win

Create Github account

- www.github.com
- Free for public repositories



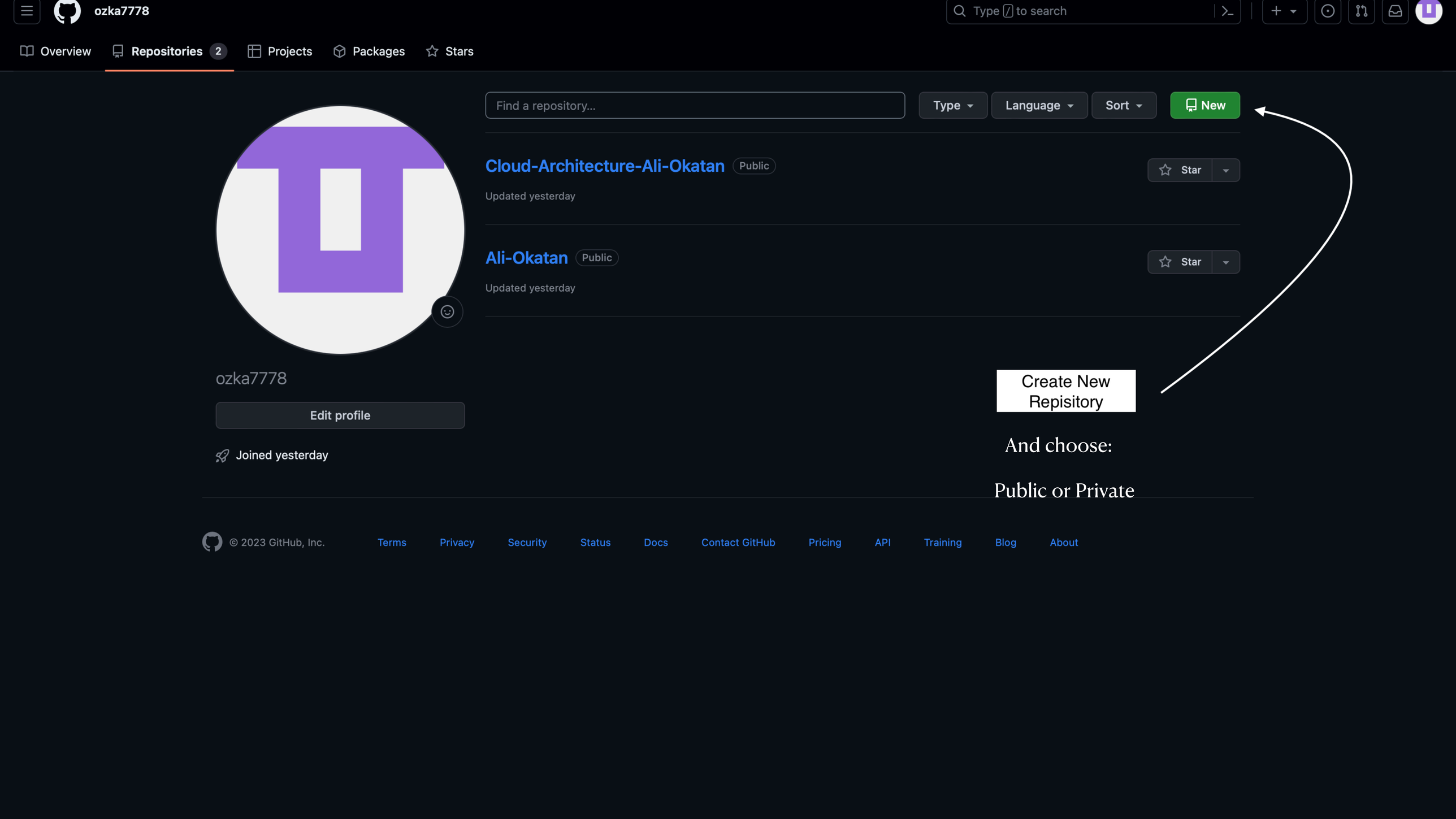
Before starting to use git

- Setup your name and email so others can know who committed changes.

- git config - -global user.name ”name”
- git config - -global user.email ”email”
  Note: set for all repositories on your computer

- git config - -local user.email ”email”
  Note: can set differently for each repository

```
Venkats-MBP:learning_git Venkat$ git config user.name
Venkat Malladi
Venkats-MBP:learning_git Venkat$ git config user.email
vsmalladi@gmail.com
```


ozka7778
Type / to search
Overview
Repositories 2
Projects
Packages
Stars
Find a repository...
Type
Language
Sort
New
Cloud-Architecture-Ali-Okatan
Public
Updated yesterday
Star
Ali-Okatan
Public
Updated yesterday
Star
ozka7778
Edit profile
Joined yesterday
Create New Repisitory
And choose:
Public or Private
© 2023 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

For uploading something to our repository we will use these codes in terminal

1. Initialize a new project in a directory:

git init

```
[ dolanmi L02029756  ~/Desktop ]$ mkdir new_project
[ dolanmi L02029756  ~/Desktop ]$ cd new_project/
[ dolanmi L02029756  ~/Desktop/new_project ]$ git init
Initialized empty Git repository in /Users/dolanmi/Desktop/new_project/.git/
[ dolanmi L02029756  ~/Desktop/new_project ]$
```

2. Add a file using a text editor to the directory
3. Add every change that has been made to the directory:

git add .

4. Commit the change to the repo:

git commit –m “important message here”

```
[ dolanmi L02029756  ~/Desktop/new_project ]$ git add .
[ dolanmi L02029756  ~/Desktop/new_project ]$ git commit -m "Add message to file.txt"
[master (root-commit) 1a7e4a5] Add message to file.txt
 1 file changed, 1 insertion(+)
```

• Create a new branch

Since you're on the master branch already, run the git checkout -b command and name your branch. The command will:

Automatically create a new branch, using the name you specify

Immediately check the branch out to you

Move you to the new branch

- **Create a new branch**

```
git checkout -b <branch name>
```

• Create a new branch

Confirm that your branch was created:

```
git status
```

Switch back to the master branch.

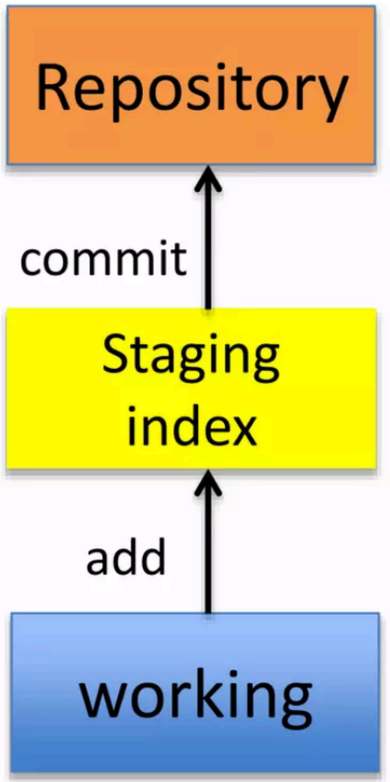
```
git checkout master
```

Deleting files from the repo

git rm filename.txt

- moves deleted file change to staging area

- It is not enough to delete the file in your working directory. You must commit the change.



git init

git status

git log

git add

git commit

git diff

git rm

git mv