Basic tools : Introduction to GitHub

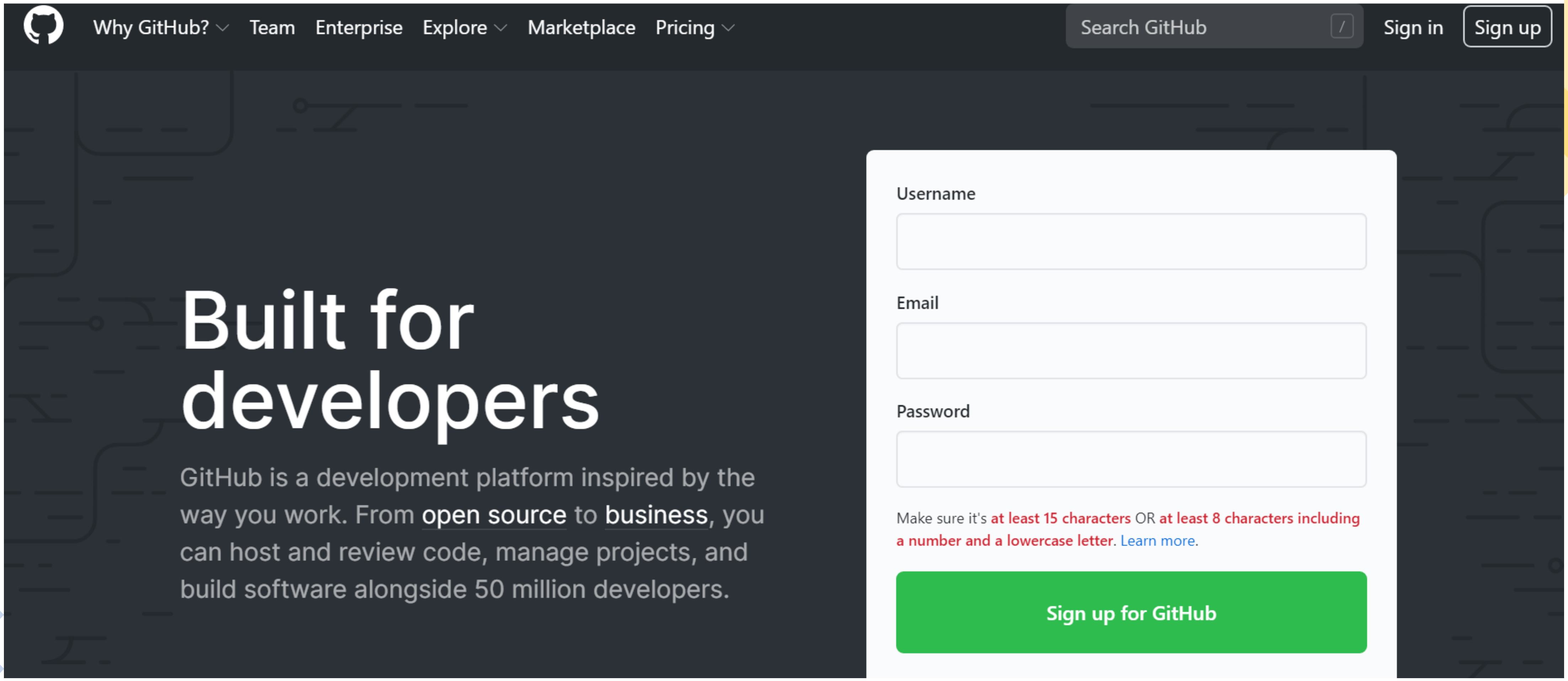
KASHTANOVA Victoriya

Inria, Epione

What is GitHub ?

- **GitHub**, Inc. is an American multinational corporation that provides hosting for software development and version control using **Git**.
- It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.
- GitHub offers its basic services free of charge and it is commonly used to host open-source projects.

Step 1 : Register on GitHub page (github.com)



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 50 million developers.

Username

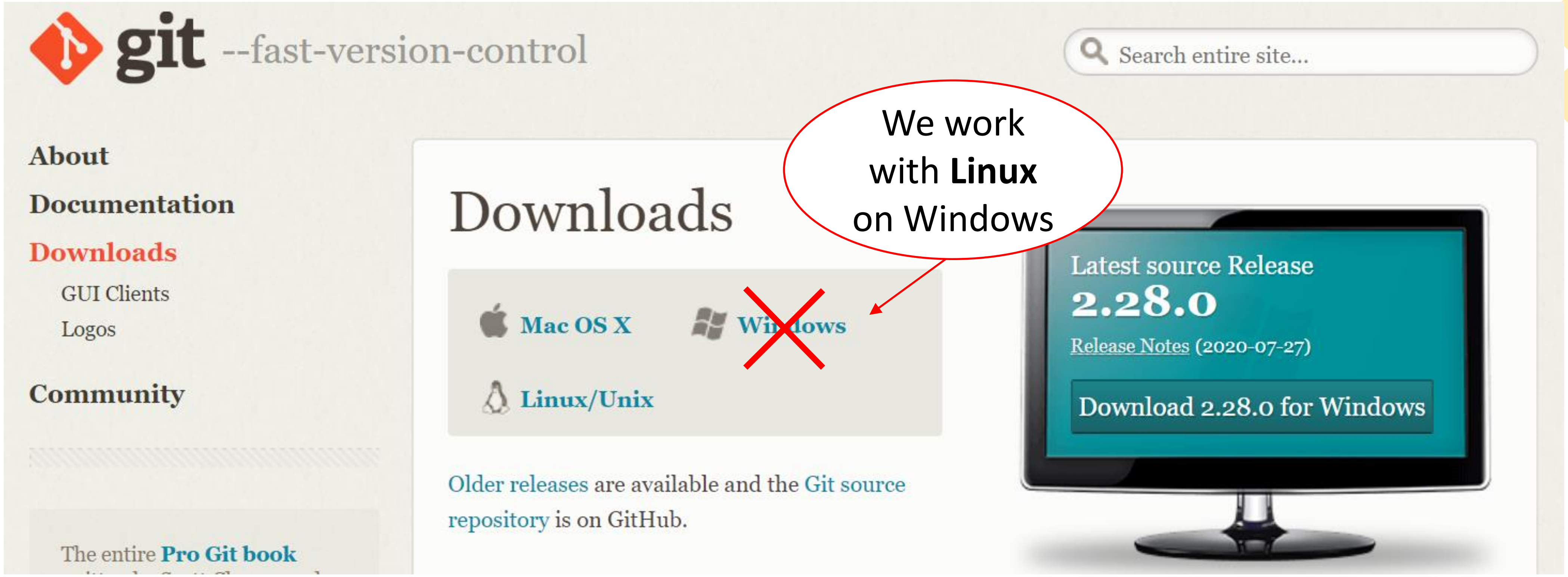
Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Sign up for GitHub

Step 2 : Install Git if needed (from https://git-scm.com/downloads)


git --fast-version-control

Search entire site...

About

Documentation

Downloads

GUI Clients

Logos

Community

The entire Pro Git book

Downloads

Mac OS X

~~Windows~~

Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release

2.28.0

Release Notes (2020-07-27)

Download 2.28.0 for Windows

We work with **Linux** on Windows

Step 3 : Link your computer to created GitHub account

Set your name and email in the global configurations :

1. git config --global user.name "My Name"
2. git config --global user.email myEmail@example.com

Create a SSH key for your computer :

1. Check if existing SSH keys are present with
   ls -al ~/.ssh
   - If you have some key pair (such as **id_rsa** and **id_rsa.pub**) in this folder you can use it and skip “Generating a new SSH key” step

2. Generate a new SSH key with
   ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

3. Add your SSH key to the ssh-agent with
   eval "$(ssh-agent -s)" (to start ssh-agent in the background)
   ssh-add ~/.ssh/id_rsa (to add your SSH private key to the ssh-agent)

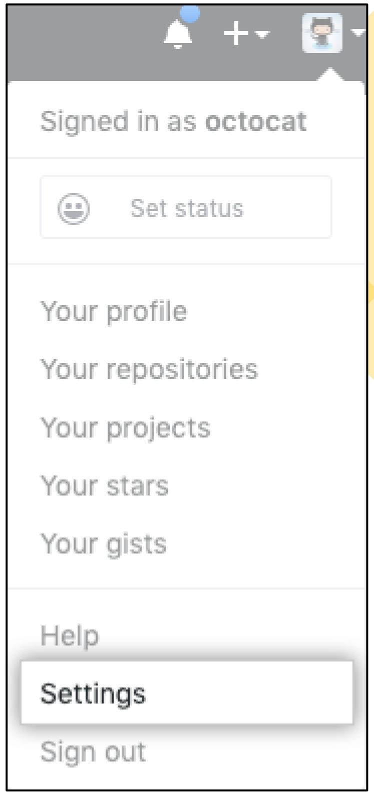
Step 3 : Link your computer to created GitHub account

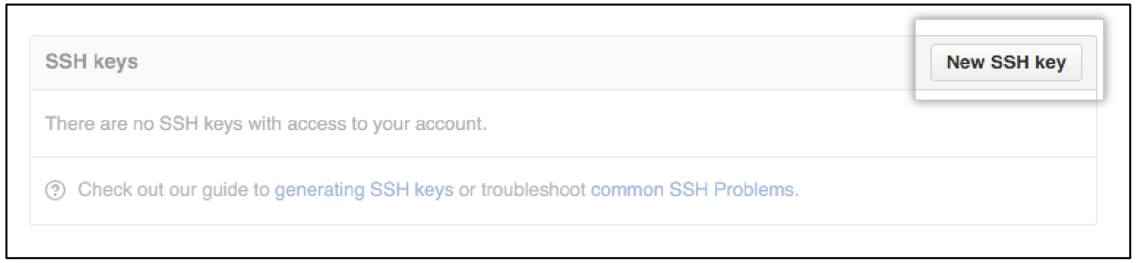
<u>Add a new SSH key to your GitHub account :</u>

1. Copy the SSH key to your clipboard with

   xclip -sel clip < ~/.ssh/id_rsa.pub

2. Open your GitHub page on browser, find **Account > Settings > SSH and GPG keys**
3. Click New SSH key or Add SSH key
4. In the "Title" field, add a descriptive label for the new key
5. Paste your key into the "Key" field
6. Click Add SSH key
7. If prompted, confirm your GitHub password



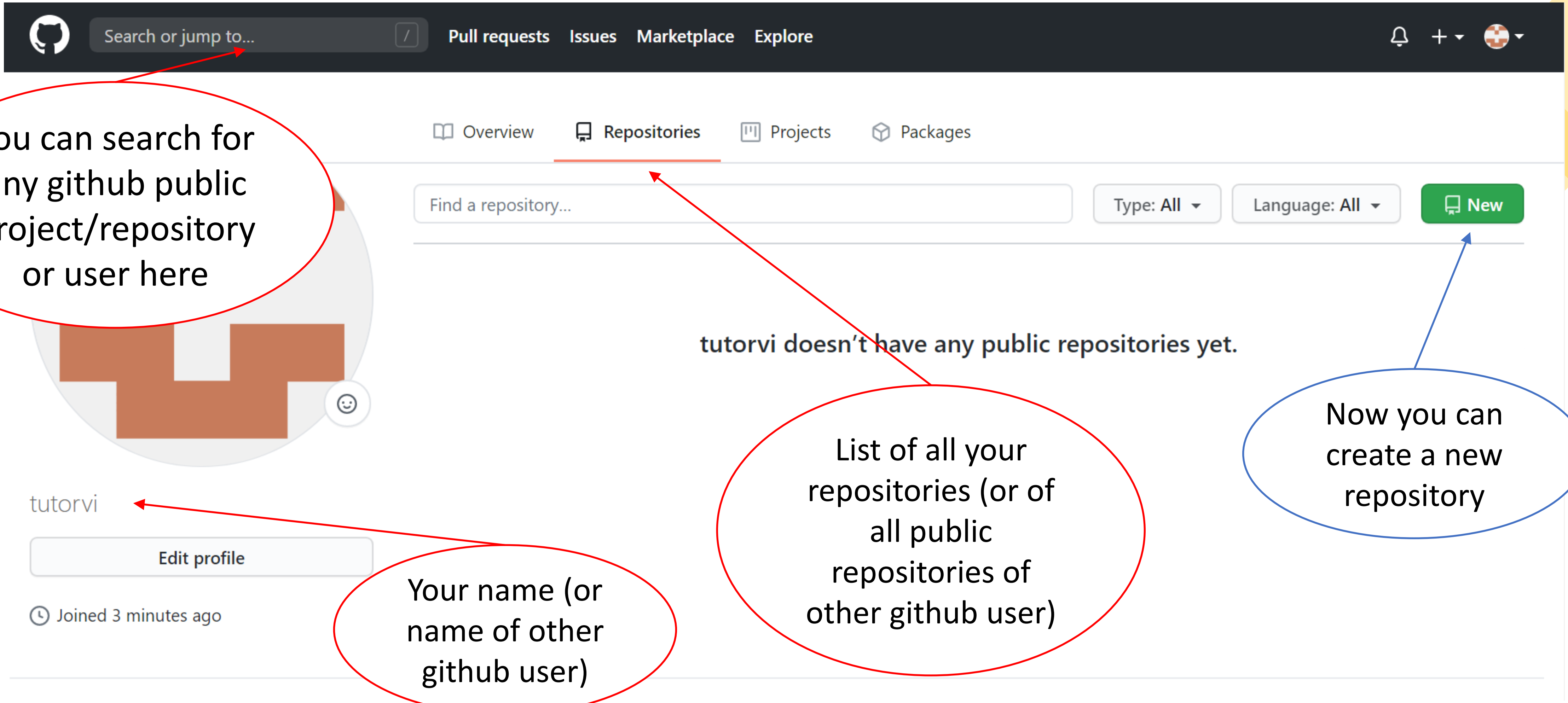

SSH keys

New SSH key

There are no SSH keys with access to your account.

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

Step 4 : Use GitHub



Search or jump to...

Pull requests Issues Marketplace Explore

Overview Repositories Projects Packages

Find a repository... Type: All Language: All New

tutorvi doesn't have any public repositories yet.

tutorvi

Edit profile

Joined 3 minutes ago

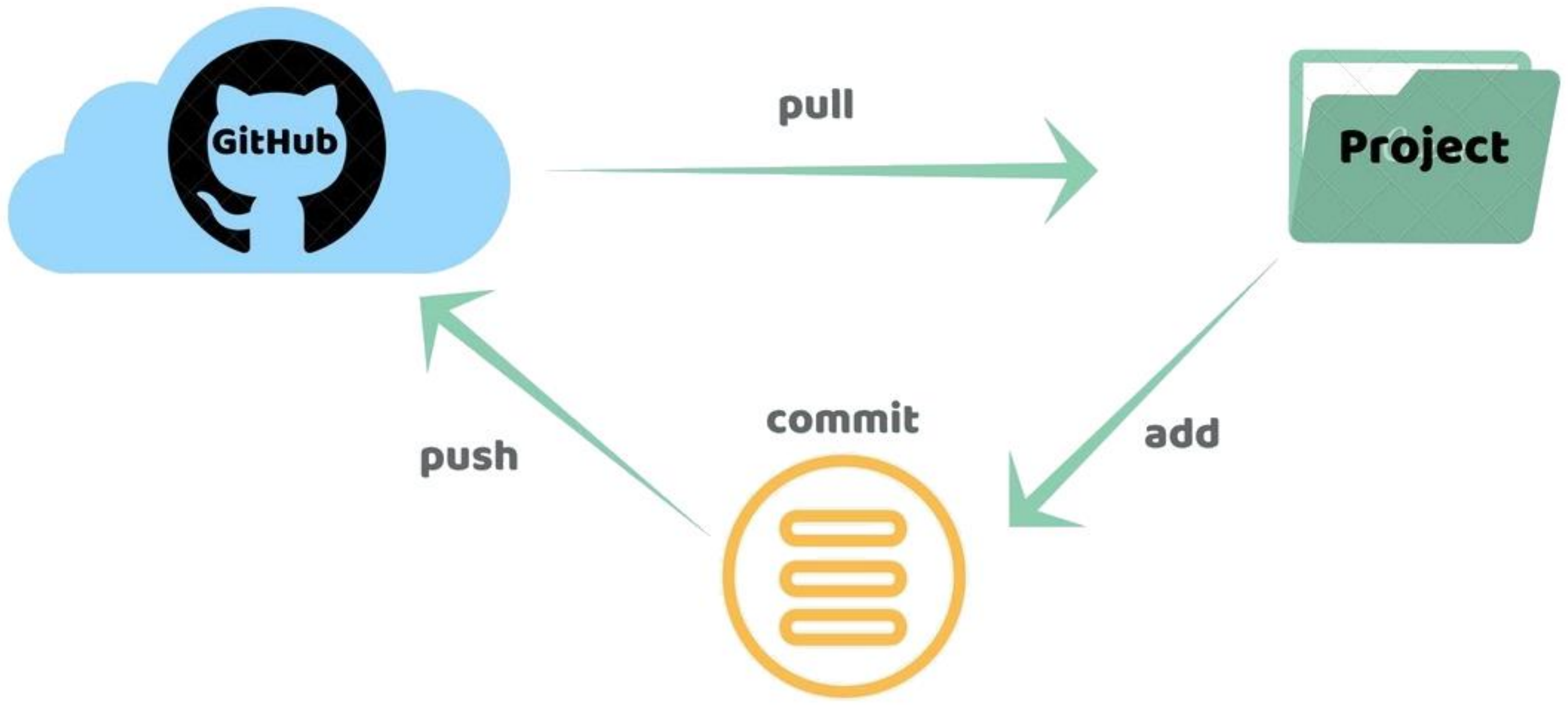
You can search for any github public project/repository or user here

List of all your repositories (or of all public repositories of other github user)

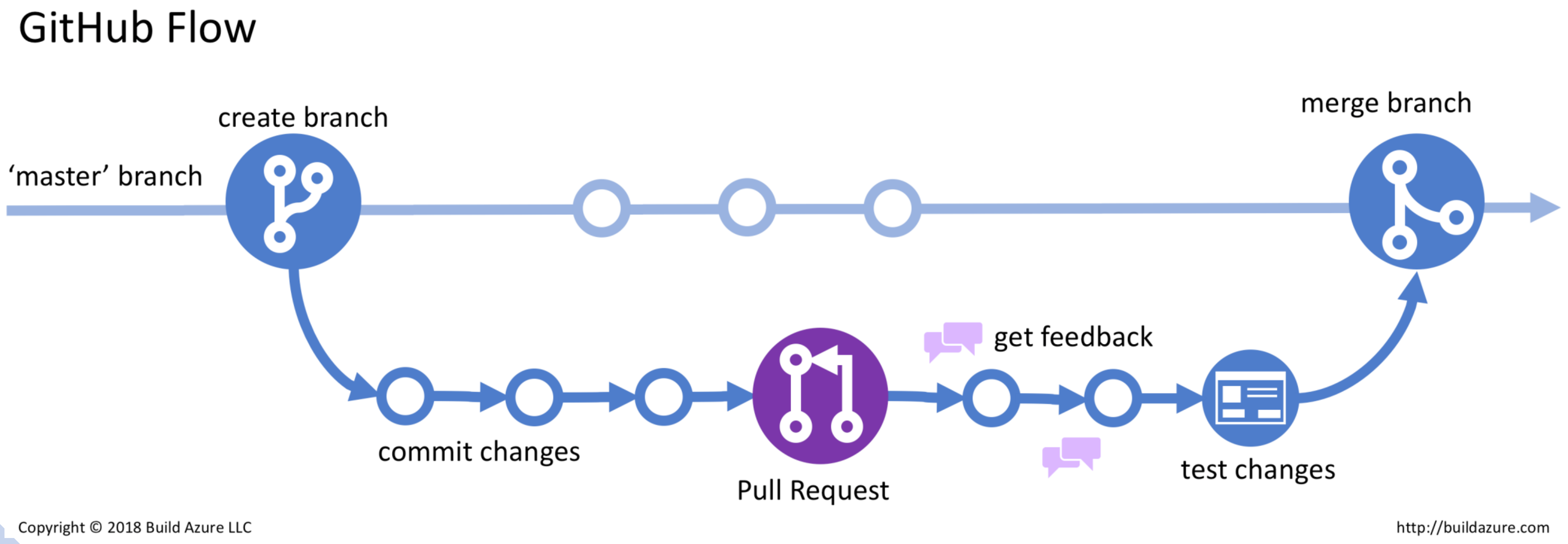
Now you can create a new repository

Your name (or name of other github user)

Local GitHub working flow (personal repository)



Global GitHub working flow (team repository)



Basic Git command

- **git init** - initializes a brand new Git repository and begins tracking an existing directory. It adds a hidden subfolder within the existing directory that houses the internal data structure required for version control.
- **git clone** - creates a local copy of a project that already exists remotely. The clone includes all the project's files, history, and branches.
- **git add** - stages a change. Git tracks changes to a developer's codebase, but it's necessary to stage and take a snapshot of the changes to include them in the project's history.
- **git commit** - saves the snapshot to the project history and completes the change-tracking process. In short, a commit functions like taking a photo. **Anything that's been staged with git add will become a part of the snapshot with git commit.**
- **git status** - shows the status of changes as untracked, modified, or staged.
- **git pull** - updates the local line of development with updates from its remote counterpart. Developers use this command if a teammate has made commits to a branch on a remote, and they would like to reflect those changes in their local environment.
- **git push -** updates the remote repository with any commits made locally to a branch.

Basic Git command (useful today)

- **git clone** - creates a local copy of a project that already exists remotely. The clone includes all the project's files, history, and branches.
- **git add** - stages a change. Git tracks changes to a developer's codebase, but it's necessary to stage and take a snapshot of the changes to include them in the project's history.
- **git commit** - saves the snapshot to the project history and completes the change-tracking process. In short, a commit functions like taking a photo. **Anything that's been staged with git add will become a part of the snapshot with git commit.**
- **git push -** updates the remote repository with any commits made locally to a branch.
- **git pull** - updates the local line of development with updates from its remote counterpart. Developers use this command if a teammate has made commits to a branch on a remote, and they would like to reflect those changes in their local environment.

Exact commands for practical work

When you add some files into directory :

- **git add** (**name of new file** or **.**)
- **git commit -m “line of commit"**
- **git push** (-u origin master)

When your collaborator changed smth and added it to remote :

- **git pull**

Example : Recommendations for creating a repository from GitHub

Create a new repository on the command line :

- echo "# test" >> README.md
- git init
- git add README.md
- git commit -m "first commit"
- git branch -M master
- git remote add origin https://github.com/**your_github_name/name_of_repository**.git
- git push -u origin master

Push an existing repository from the command line :

- git remote add origin https://github.com/**your_github_name/name_of_repository**.git
- git branch -M master
- git push -u origin master