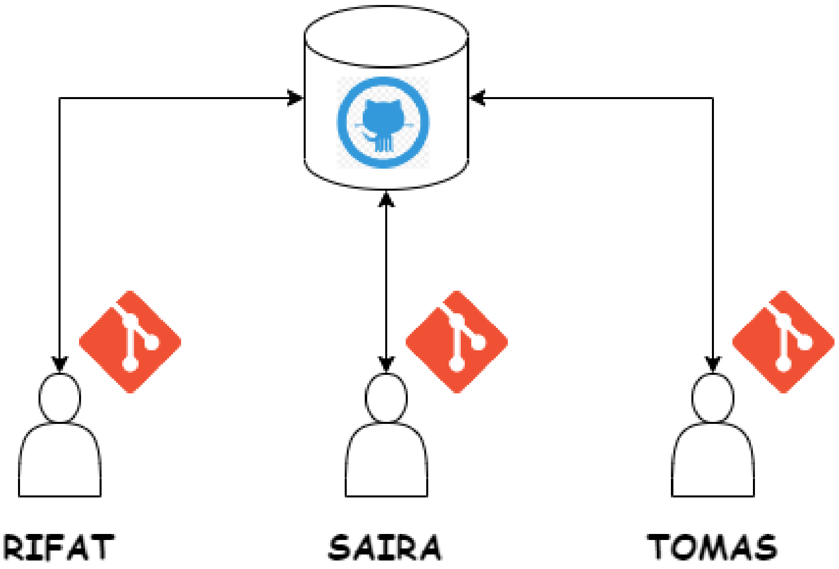
GIT is a Version Control System:

1. Tracking and controlling the differences between two versions of a same file
2. Remote Collaboration beyond physical boundaries.

Git vs GitHub [Local and Remote Repository]

1. A Scenario-based discussion



GITHUB	GIT
Cloud Service to Host git	Tool to Control and Track Version
Remote Connection Provider and Maintained on Cloud/Web	Installed and Maintained Locally
Provides a Graphical User Interface (GUI)	Generally, Command Line Interface (CLI), A Limited GUI

Git Setup and Configuration

[1] https://git-scm.com/downloads

[2] Open Git Bash

[3] Check the Git Version

A Demo Repository Creation:

[1] Create a New Folder

[2] Initialize the Git: `git init`

[3] Username and Email Setup:

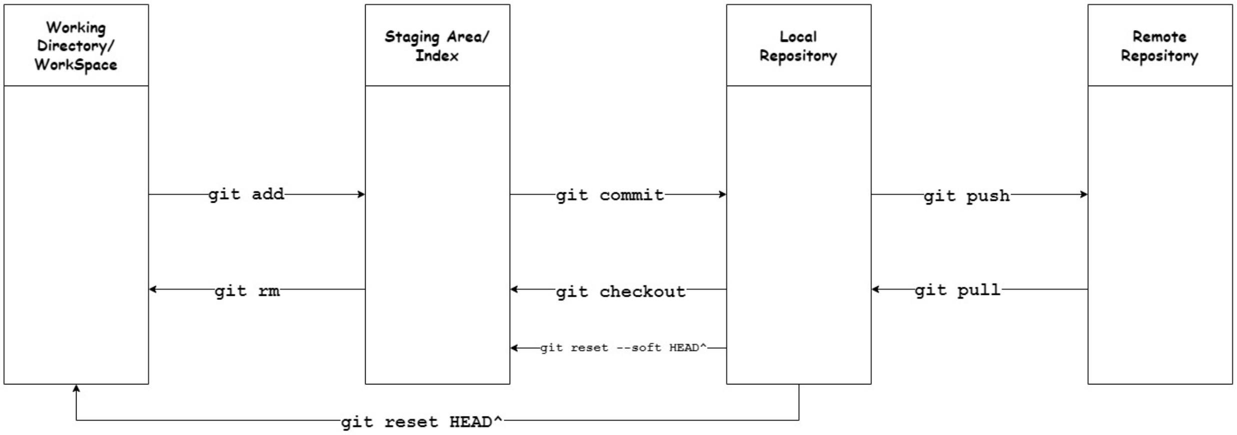
1. Check Current Config: `git config –list`
2. Username Set: `git config --global user.name "Akib Zaman"`
3. Email Set: `git config --global user.email "xxxx@gmail.com"`
4. Username Check: `git config user.name`
5. Username Check: `git config user.email`

[3] Add some new file to the Folder

[4] Check Status: `git status`

[5] Let’s add the new file to staging Area

Staging Area Concept:



Add, and Commit [Local Repo]:

Adding to the Staging Area:

1. Only a specific File: `git add < FILENAME> [git add hello.c]`
2. Only files of a folder but not the subfolders: `git add .`
3. All files and subfolders: `git add -A`
4. Only a particular type of file from directory: `git add *.c`
5. Only a particular type of file from directory & subdirector: `git add **/*.c`

Checking the Differences between previous and current version: `git diff`

Restore the earlier version of the file: `git restore < FILENAME>`

Removing from Staging Area: `git rm --cached <FILENAME>`

Committing to the Local Repository:

1. Only Commit: `git commit -m “first commit”`
2. Add and Commit: `git commit -am “first commit”`
3. See Commit History: `git log -m`
4. See in a Line: `git log -oneline`
5. Specific Commit: `git show COMMIT_ID`

Uncommitting from a Local Repository:

1. Local Repo to Staging Area: `git reset --soft HEAD^`
2. Local Repo to Staging Area (Multiple Commit): `git reset --soft HEAD~2`
3. Local Repo to Working Dir: `git reset HEAD^`
4. Total Deletion of a Commit: `git reset --hard HEAD^`

Checkout:

1. To a specific commit: `git checkout COMMIT_ID`
2. Move back: `git checkout master`

Git Collaboration

[1] Create a GitHub Repository

[2] Show Remote Repo Link: `git remote -v`

[3] Add a Remote Repo Link: `git remote add <NAME> <LINK> [git remote add origin link]`

[4] Clone a Remote Repo: `git clone <LINK>`

[5] Push: `git push -u origin`

[6] Push setting the branch upstream: `git push --set-upstream origin master`

Branch:

[1] Branch Show: `git branch`

[2] Branch Creation: `git branch <BRANCHNAME> [git branch new-feature]`

[3] Branch Select: `git checkout <BRANCHNAME> [git checkout new-feature]`