

NOTES

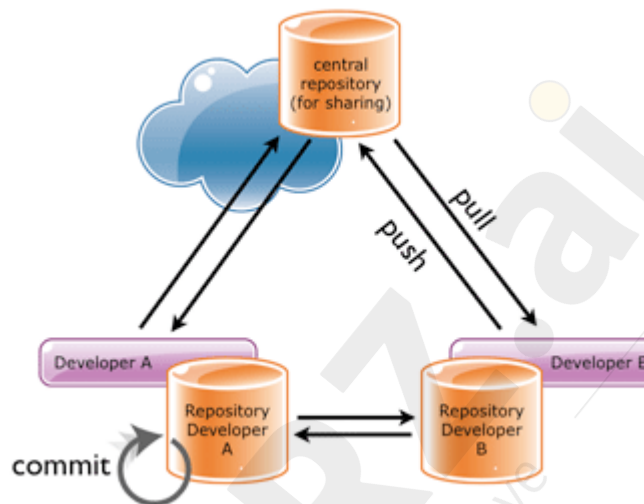
Git & GitHub



GIT & GITHUB NOTES

Definition and Purpose of Version Control (VCS)

Version Control (VCS) is a system that tracks changes to files over time, allowing multiple users to collaborate, revert to previous versions, and manage modifications efficiently. It helps developers keep a history of their work, manage code branches, and resolve conflicts when working in teams on shared projects.



Important Commands of GIT

1. Set-Up Git

- `git config --global user.name "Your User Name"`
Sets your Git username.
- `git config --global user.email "you@example.com"`
Sets your Git email.

```
nothing added to commit but untracked files present (use 'git add' to track)
rival@MARCOS_MARINE MINGW64 ~ (sandeepak)
$ git config --global user.name "rivalsolmons"
rival@MARCOS_MARINE MINGW64 ~ (sandeepak)
$ git config --global user.email "myemail@gmail.com"
rival@MARCOS_MARINE MINGW64 ~ (sandeepak)
$
```

2. Repository Setup

- git init

Initializes a new Git repository in the current directory.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git init
Reinitialized existing Git repository in C:/Users/rival/Work/Tech Hub/sir/.git/

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ |
```

- git clone <repository-url>

Copies an existing Git repository (e.g., from GitHub) to your local machine.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git clone https://github.com/rivalsolmons/testing.git
Cloning into 'testing'...
remote: Enumerating objects: 73, done.
remote: Counting objects: 100% (73/73), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 73 (delta 23), reused 48 (delta 11), pack-reused 0 (from 0)
Receiving objects: 100% (73/73), 9.58 KiB | 1.20 MiB/s, done.
Resolving deltas: 100% (23/23), done.

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ |
```

3. Basic File Operations

- git status

Shows the current state of the working directory (which files are staged, modified, or untracked).

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 2 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    20Sept.html
    a21FS.js
    bb.html
    testing/

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

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ |
```

- git add <filename>

Stages a file for a commit (moves it to the staging area).

- git add .

Stages all modified files for commit.

- `git rm <filename>`

Removes a file from the working directory and the Git index.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ rm nevfdbw.txt

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ ls
aimerz_army.txt  aimerz_fullstack.txt  aimerz_fullstack_army.txt  aimerz_new_file.txt

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$
```

- `git mv <old-filename> <new-filename>`

Renames a file.

4. Committing Changes

- `git commit -m "commit message"`

Commits staged changes with a descriptive message.

- `git commit --amend` Edits the last commit (for cases like adding missing changes or fixing a message).

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git commit -m 'my Message'
[main 5b196d1] my Message
2 files changed, 1 insertion(+), 2 deletions(-)
delete mode 100644 nevfdbw.txt
create mode 160000 testing

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$
```

5. Branching & Merging

- `git branch`

Lists all branches in the current repository.

- `git branch <branch-name>`

Creates a new branch.

- `git checkout <branch-name>`

Switches to a different branch.

- `git merge <branch-name>`

Merges the specified branch into the current branch.

- `git branch -d <branch-name>`

Deletes a branch.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git branch
  aimerz
* main
  sandeepak

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ |
```

6. Pushing & Pulling

- git pull origin <branch>

Fetches and merges changes from the remote repository into your current branch.

- git push origin <branch>

Pushes your local changes to the remote repository.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git pull
Updating e2bc44d..a710ccd
Fast-forward
 file/version.txt          | 1 +
 production_services/bugs.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 file/version.txt
 create mode 100644 production_services/bugs.txt
```

7. Undoing Changes

- git checkout --<filename>

Discards changes in a working directory file (reverts it to the last commit).

- git reset <filename>

Unstages a file (removes it from the staging area but keeps changes in the working directory).

- git reset - -hard <commit>

Resets the working directory to a specific commit (removes all changes after that commit).

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git checkout aimerz_new_file.txt
Updated 0 paths from the index

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git reset

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$
```

8. Viewing History

- git log

Displays the commit history.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git log
commit 5b196d166d7d0a3cf2ac636631b5e4aed5d2176f (HEAD -> main)
Author: rivalso1mons <rivalso1mons@gmail.com>
Date: Mon Sep 23 09:07:14 2024 +0530

    my Message

commit a710ccd6bc4781446ad2ac494a539e771c3c35e9 (origin/main)
Author: Sandeepak <76931978+rivalso1mons@users.noreply.github.com>
Date: Sat Sep 14 20:01:33 2024 +0530

    Create version.txt

commit 63belca6d27f6474ec1aa2293e1105dec8affc0c
Author: Sandeepak <76931978+rivalso1mons@users.noreply.github.com>
Date: Sat Sep 14 19:57:20 2024 +0530

    Create bugs.txt

commit e2bc44d4eff9b236347501dfa51201a543d6dd93 (aimerz)
Author: rivalso1mons <rivalso1mons@gmail.com>
Date: Sat Sep 14 19:17:09 2024 +0530
```

- git diff

Shows the difference between the working directory and the last commit.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ rm main.txt

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git diff
diff --git a/main.txt b/main.txt
deleted file mode 100644
index 8313fdd..0000000
--- a/main.txt
+++ /dev/null
@@ -1,2 +0,0 @@
-
-New Added text by Aimerz

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$
```

10. Remote Repositories

- git remote -v

Lists the remote repositories associated with the local repo.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git remote -v
origin https://github.com/rivalso1mons/testing.git (fetch)
origin https://github.com/rivalso1mons/testing.git (push)

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$
```

- `git remote add <name> <url>`
Adds a new remote repository.
- `git fetch`
Retrieves the latest changes from the remote repository but doesn't merge them.

11. Tagging

- `git tag <tagname>`
Creates a tag for the current commit (used for versioning).
- `git tag`
Lists all tags in the repository.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git tag Aimerz_Army

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git tag FSD

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git tag
Aimerz_Army
FSD

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$
```

12. Conflict Resolution

- After a merge conflict, you'll manually edit the conflicting files and use:
 - `git add <filename>`
To mark the conflict as resolved.
 - `git commit`
To commit the merge resolution.

```
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git pull
Auto-merging aimerz_army.txt
CONFLICT (content): Merge conflict in aimerz_army.txt
Automatic merge failed; fix conflicts and then commit the result.

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main|MERGING)
$ git AC
```

Open the file in vimdiff editor and resolve conflict by rectifying the changes then thereafter push it to main.


```
[main 19c0791] Merge branch 'main' of https://github.com/rivalsolmons/testing
rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$ git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 861 bytes | 430.00 KiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/rivalsolmons/testing.git
  3017594..19c0791  main -> main
branch 'main' set up to track 'origin/main'.

rival@MARCOS_MARINE MINGW64 ~/Work/Tech Hub/sir (main)
$
```

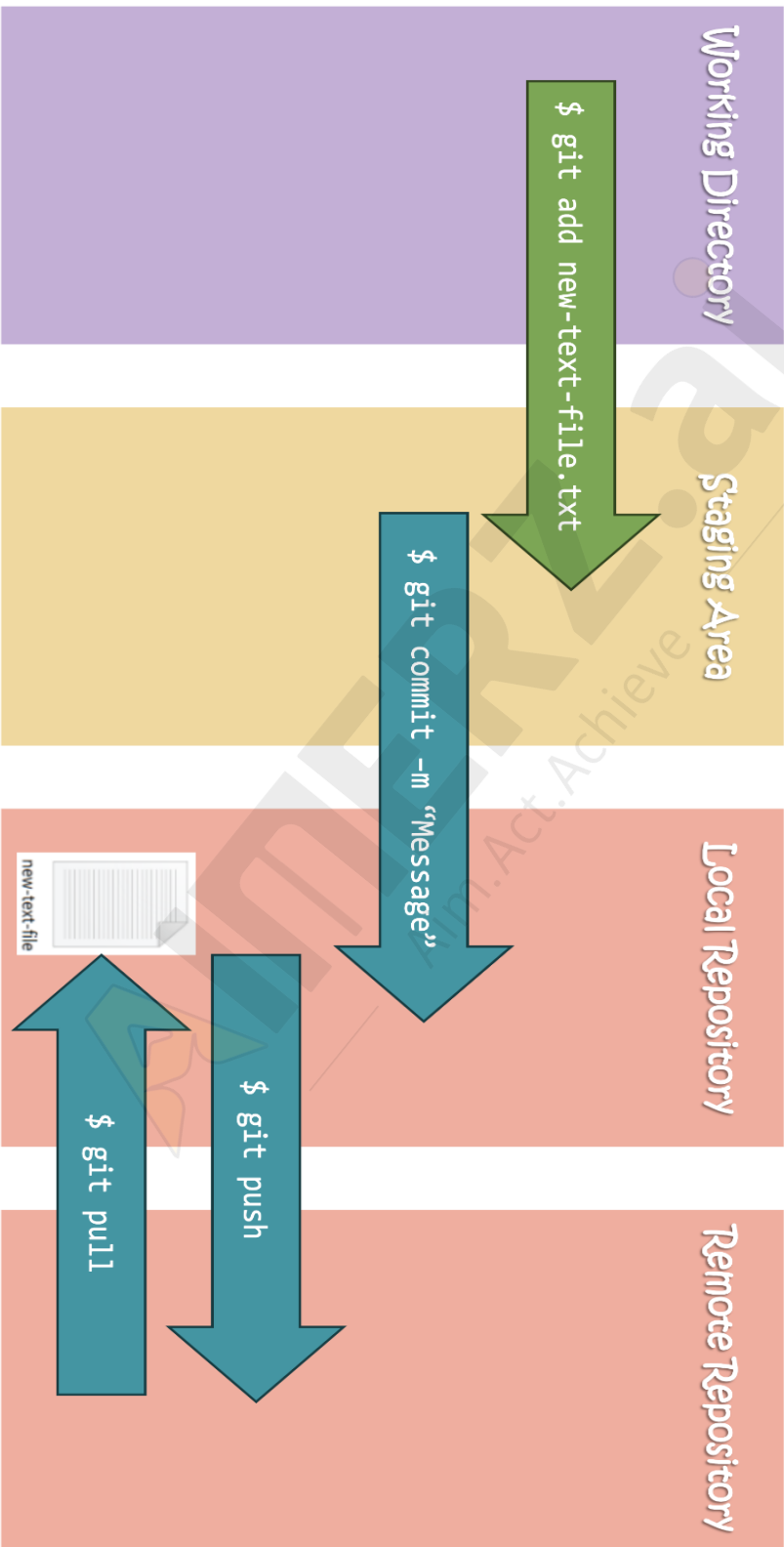
Difference between GIT and GITHUB

Git is a local, distributed version control system for tracking changes in code, managing branches, and collaborating. GitHub is a cloud-based platform built on Git, offering hosting for repositories, collaboration tools, pull requests, and additional features like issue tracking, project management, and code reviews for remote teamwork.

Some more VCS other then GIT

Additional VCS include CVS, Fossil, Plastic SCM, Darcs, Rational Synergy, and Vesta, supporting both centralized and distributed workflows.

MIND MAP FOR VISUALIZATION



THANK YOU

