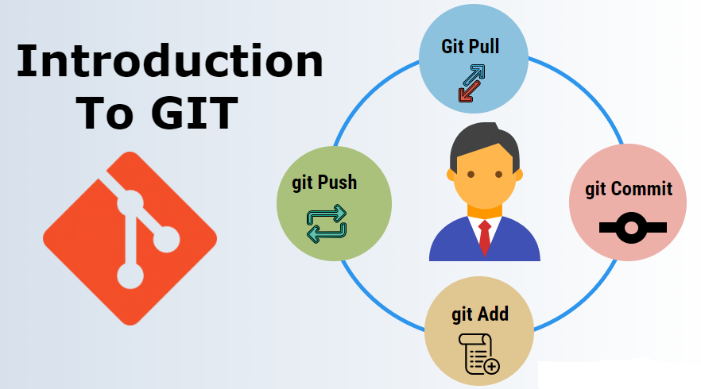
d. Git version control

Git is a distributed version control system, meaning that it allows developers to work on their own local copies of a project,

Git helps manage and track changes to code, but it does so in a decentralized way

This design makes Git fast, scalable, and highly resilient to issues like server failures

* Why uses Git?

Git becomes a well-known tool that developers can use to handle changes to their codes with ease. Given that it is a distributed system, every participant in this project will not only have access to the complete history of all its files but also flexibility is enhanced particularly during offline or remote-related tasks.

* When to use Git?

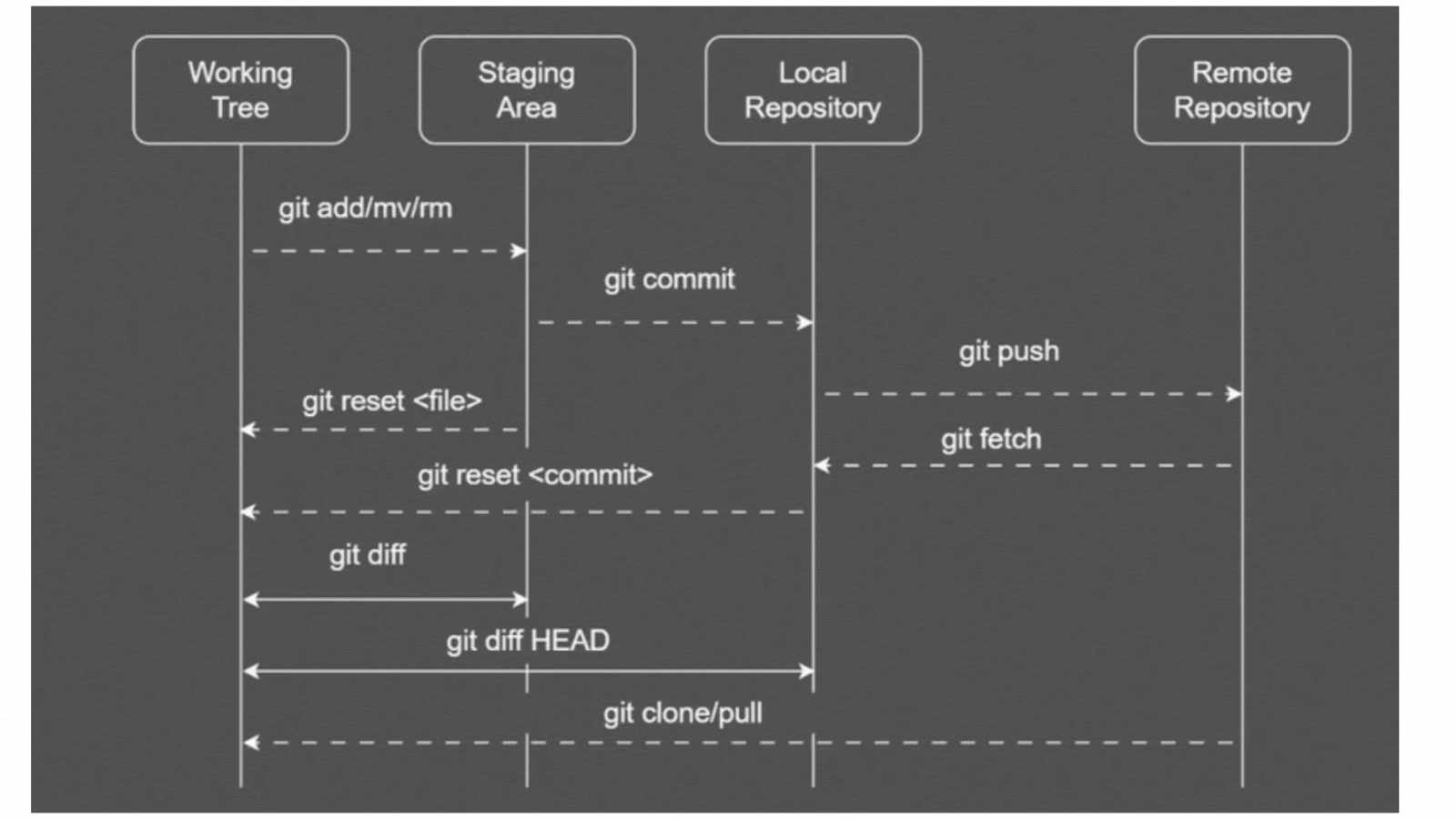
use Git if we want to collaborate with other developers on a coding project or work on own project.

displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git. Status output does not show you any information regarding the committed project history.

* Key Features of Git
* Version Tracking: Git follows all adjustments done in one record, letting you revert to old releases without trouble.
* Collaboration: Different programmers can work on a similar task at the same time without clash.
* Branching: You have the option to create distinct branches for new attributes, bug repairs or tests.
* Distributed System: Every programmer has an entire version of the project implying that it is decentralized software.
* Log of Commits: With this feature, Git maintains an account of all commit actions (changes), which makes understanding how a project has evolved over time much easier.
* **TERMS RELATED TO GIT**

There are so many terms related to git, some of them are the following:

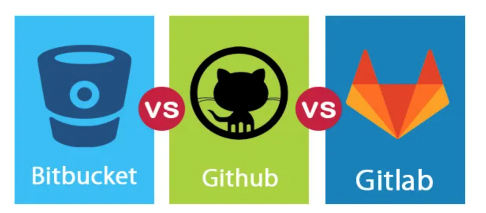
* **HOW GIT IS WORKING**

****

**There four main stages in using git as the following:**

1. **Working Directory:** It’s the workspace where the code can be modified and change without any restrictions.
2. **Staging Area:** It’s the space where the codes are reviewed before saving them as a new copy of the project and also here no modification can be made to the written code.
3. **Local repo:** It’s the space where the codes are saved as a new copy of the project and be ready to push to the remote repo.
4. **Remote Repo:** It’s a remote space where the main project is stored so that all team members can share their work with the rest of the team and get the latest update from the project.

* **THE CLOUD REPOSITORIES THAT USE GIT VERSION CONTROL**



|  |  |  |  |
| --- | --- | --- | --- |
| Features | Bitbucket | GitHub | Gitlab |
| Free private repositories | Yes | Yes | Yes |
| Free public repositories | Yes | Yes | Yes |
| Merge Request/Issue Templates | No | Yes | Yes |
| Integrated CI | yes | No | Yes |
| Open-source | No | No | Yes |
| File storage | Yes | Yes | Yes |
| Integrations | Yes | Yes | Yes |
| Analytics | Yes | No | Yes |

* **THE MOST COMMON COMMANDS USED IN GIT**

|  |  |
| --- | --- |
| Commands | Function |
| git config --global user.name “Zahra Developer” | set global user name for your PC |
| git config --global user.email “[zalhabsi1994@hotmail.com](mailto:zalhabsi1994@hotmail.com)” | link your PC with your account in remote repo using your email address |
| git init | initialize the git system in your project working directory |
| git clone | to get a project from remote repo |
| git add <file.name> | move one file to staging area |
| git add . | move all the files you have to the staging area |
| git commit -m “message/comment/updates” | move your work to local repo by commit it and give it a message |
| git status | list the files you have in working directory and staging area which are not committed yet |
| git log --online | list all the commit you have created in your project |
| git remote add | link your project with the remote repo you create for this project |
| git remote | list all link for remote repo you have in your PC |
| git push | upload work committed work to the remote repo |
| git pull | get or download the part you do not have in the main project which is in the remote repo |
|  |  |