1. Git is a distributed version control system used to maintain a history of changes to program code or other files, collaborate on projects, and manage different versions of files. The basic actions and commands in Git include:

init: Initialize a new repository.

clone: Create a copy of a remote repository on your local computer.

add: Adding changed or new files to a staging area in preparation for a commit.

commit: Commits changes that have been added to a stage with a brief description of the changes (comment).

status: Checks the status of the working directory, indicating which files have been changed or added and which are not included in the commit.

log: View the history of the commit.

branch: Create, delete, and manage branches of code development.

merge: Merge branches together to combine changes from one branch to another.

pull: Update a local repository from a remote repository and merge changes.

push: Send commits from the local repository to the remote repository.

fetch: Retrieves change information from the remote repository, but does not merge it into the local repository.

checkout: Switch between branches or commits.

2. Commit in Git is an action that allows you to commit changes to a repository. During a commit, you save a set of changes to the internal history of the repository as an independent history item. Each commit has a unique identifier (hash), a short comment that explains the essence of the changes, and a link to the previous commit. Commit allows you to track changes in files, meaning you can see which files have been changed, their status at the time of each commit, and simplify the development process by knowing how and when changes were made.

Thanks to this version control system, development teams can work in parallel on a project, making changes and committing them through commits. This makes Git a powerful tool for collaboration, tracking changes, deploying releases, and resolving conflicts in projects large and small.