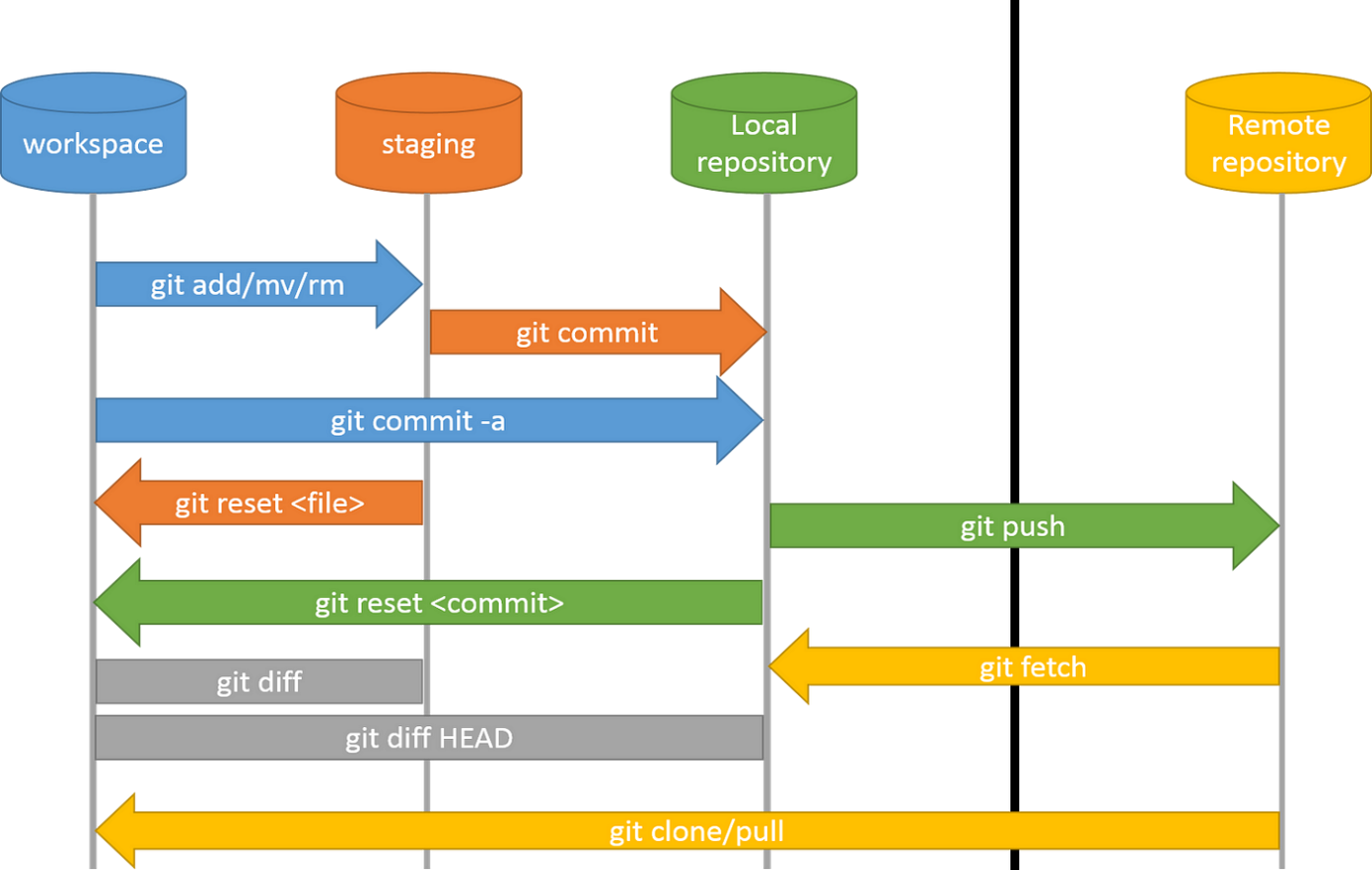
**Git Architecture**



**Key Features**

* [Git](https://www.simplilearn.com/tutorials/git-tutorial/git-tutorial-for-beginner) is a [DevOps tool](https://www.simplilearn.com/tutorials/devops-tutorial/devops-tools) used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently.
* Git uses a three-stage architecture - working directory, staging area, and local repository to optimize change tracking.
* Key concepts like committing, branching, merging, and remotes enable powerful version control workflows.
* Git maintains an extensive history and provides commands like git log and git diff to analyze changes over time.

**Git Architecture & Components**

While many version control systems use a two-tier architecture consisting of a repository and a working copy, Git distinguishes itself with a three-stage model optimized for tracking changes: the working directory, staging area, and local repository. Additionally, Git includes the concept of remote repositories for collaboration.

The Git workflow is divided into three states:

* Working directory - Modify files in your working directory
* Staging area (Index) - Stage the files and add snapshots of them to your staging area
* Git directory (Repository) - Perform a commit that stores the snapshots permanently to your Git directory. Checkout any existing version, make changes, stage them and commit.

**Git Commands**

[**Git**](https://www.geeksforgeeks.org/git-tutorial/)**commands are crucial for efficient collaboration and project management.**In this article, we’ll explore a list of important Git commands like git commands to push, git commit command, git pull command, and git push command, etc that will help to improve workflow and optimize productivity. These are a **Git Commands list**that can be used frequently on Git.

Frequently used commands for Git's [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) include:

* git init, which is used to create a git repository.
* git clone [URL], which clones, or duplicates, a git repository from an external URL.
* git add [file], which adds a file to git's working directory (files about to be committed).
* git remote add
* git commit -m [commit message], which commits the files from the current working directory (so they are now part of the repository's history).
* git config
* git help
* git config –-global user.name “ ”
* git config –-global user.email “ ”
* git status
* git log
* git push origin main
* git checkout -b
* git merge [URL]

## 

