Git Cheat Sheet

Brought to you by the Teamfu Team

August 2023

Configuration

3

git config --global user.name "Your Name"
Set your global username.

git config --global user.email "youremail@example.com"
Set your global email.

git config --list
List your local git configurations

Basic Commands

6

git init
Initializes a new Git repository in the current directory.

git add .
Stage changes for the next commit.

git status
Show the current status of the repository (modified files, etc.).

git commit -m "My message"
Commit staged changes with a descriptive message.

git log
Show commit history.

git clone [repository URL]
Clones an existing remote repository onto your local machine.

Basic Concepts

7

Repository
A project's version controlled directory, including all files and commit history.

Clone
Copying a repository from a remote source to your local machine.

Commit
Saving changes to the repository with a message.

Branch
A separate line of development to isolate changes and work on features independently.

Merge
Combining changes from one branch into another.

Pull
Fetching changes from a remote repository and merging them into your local branch.

Push
Sending your committed changes to a remote repository.

Branching

5

git branch
List all branches. The current branch is marked with an asterisk.

git checkout <branch-name>
Switch to an existing branch.

git checkout -b <branch-name>
Create and switch to a new branch.

git merge <branch-name>
Merge changes from a specified branch into the current branch.

git branch -d [branch-name]
Deletes the specified branch if it's already merged into the current branch.

Remote repos

5

git remote add origin <remote-url>
Link your local repository to a remote repository.

git pull origin <branch-name>
Pull changes from a remote branch to your current branch.

git push origin <branch-name>
Push your committed changes to a remote branch.

git push origin <branch-name>
Push your committed changes to a remote branch.

git fetch origin
This fetches all the changes from the remote repository named "origin."

Resolving conflicts

5

Conflicts occur when Git can't automatically merge changes. You must resolve them manually.

git diff
Show the differences between conflicting files.

Open the conflicting file, search for conflict markers, and make necessary changes.

git add <filename>
Stage the resolved file.

git commit -m "Merge conflict resolved"
Commit the resolved changes.

Other useful commands

git pull --rebase
Fetch and reapply local changes before the pull.

git reset --hard HEAD
Discard all local changes and return to the last commit.

git stash
Temporarily save changes to a stack for later retrieval.

git clean -f
Remove untracked files from the working directory.

git branch -m [new-branch-name]
Renames the current branch to the new name.

git log --graph --oneline --all
Displays a compact, graphical representation of the commit history, showing branches and their relationships.

git branch --merged
Lists all the branches that have been merged into the current branch.

Branching strategy

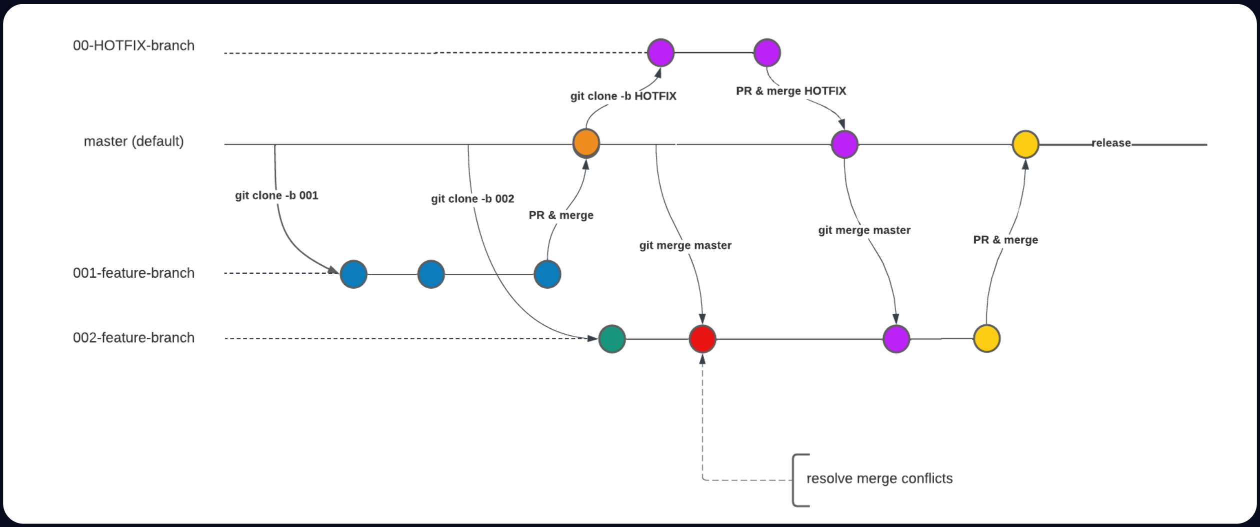
Branches represent isolated work, ensuring each feature or bug fix is developed separately (e.g., Feature-A, Feature-B, Hotfix).

The master branch reflects the latest stable release, always deployable to production. Feature branches enable parallel development, speeding up progress.

Completed features merge back into master for inclusion in the next release.

Hotfixes address critical issues separately, branching from master and merging back promptly.

Thorough testing and code reviews precede merging to maintain code quality and minimise bugs.



Generating SSH keys

To download source code from a repository, you can use a compressed zip file, HTTPS, or SSH. SSH is recommended for stability, performance, and automatically signing your commits.

To set up SSH, generate a key pair with ***ssh-keygen -t ed25519 -C "<your-email-address>"***, preferably with an empty passphrase to speed up interactions. Copy the public key to your clipboard, and paste your public key into your GitHub account settings (SSH and GPG keys) to associate it.

Test your key by cloning a repository using **ssh -T *git@github.com***

Now you can successfully clone your repo with ***git clone git@github.com:<user-name>/<repo-name>.git***

The cloned code will be placed in a new folder in your current directory.