For more detail, check out the Atlassian Git Tutorials for a visual introduction to Git commands

Atlassian Support / Bitbucket 5.15 / Documentation / ... / Git resources

Basic Git commands

and workflows, including examples.

Here is a list of some basic Git commands to get you going with Git.

Cloud Server 5.15

Git resources

Related content

Download an archive from Bitbucket Server

Importing code from an existing project

Commit and push changes to Bitbucket

Server

Integrating Bitbucket Server with Atlassian applications

Still need help?

The Atlassian Community is here for you.

Ask the community

Git task	Notes	Git commands
Tell Git who you are	Configure the author name and email address to be used with your commits. Note that Git strips some characters (for example trailing periods) from user.name.	git configglobal user.name "Sam Smith" git configglobal user.email sam@example.com
Create a new local repository		git init
Check out a repository	Create a working copy of a local repository:	git clone /path/to/repository
	For a remote server, use:	git clone username@host:/path/to/repository

git add <filename> Add files Add one or more files to git add * staging (index): git commit -m "Commit message" Commit Commit changes to head (but not yet to the remote repository): git commit -a Commit any files you've added with git add, and also commit any files you've changed since then: git push origin master Send changes to **Push** the master branch of your remote repository: git status List the files **Status** you've changed and those you still need to add or commit:

Connect If you haven't git remote add origin <server> connected your to a local repository remote repository to a remote server, add the server to be able to push to List all currently git remote -v configured remote repositories: git checkout -b <branchname> **Branches** Create a new branch and switch to it: git checkout <branchname> Switch from one branch to another:

git branch List all the branches in your repo, and also tell you what branch you're currently in: git branch -d <branchname> Delete the feature branch: git push origin <branchname> Push the branch to your remote repository, so others can use git push --all origin Push all branches to your remote repository: git push origin :<branchname> Delete a branch on your remote repository:

git pull

Update

remote

from the

Fetch and

merge changes

on the remote

repository server to your working directory: git merge <branchname> To merge a different branch into your active branch: View all the git diff merge conflicts: git diff --base <filename> View the conflicts against git diff <sourcebranch> <targetbranch> the base file: Preview changes, before merging: git add <filename> After you have manually resolved any conflicts, you mark the changed file: git tag 1.0.0 <commitID> **Tags** You can use tagging to mark a significant changeset, such as a release: git log Commitld is the leading characters of the changeset ID, up to 10, but must be unique. Get the ID using: git push --tags origin Push all tags to remote repository: git checkout -- <filename> Undo If you mess up, local you can replace the changes in changes

your working tree with the last content in head: Changes already added to the index, as well as new files, will be kept. git fetch origin Instead, to drop all your local git reset --hard origin/master changes and commits, fetch the latest history from the server and point your local master branch at it, do this: Search the Search git grep "foo()" working directory for foo():

Last modified on Sep 21, 2017

Was this helpful? Yes No Provide feedback about this article

Powered by Confluence and Scroll Viewport.

A ATLASSIAN