

Bitbucket Server ▾

Bitbucket Server Latest ▾

[Git resources](#)

Bitbucket Server was previously known as Atlassian Stash. [Learn more.](#)

Basic Git commands

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

For more detail, check out the [Atlassian Git Tutorials](#) for a visual introduction to Git commands and workflows, including examples.

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 <code>user.name</code> .	<code>git config --global user.name "Sam Smith"</code> <code>git config --global user.email sam@example.com</code>
Create a new local repository		<code>git init</code>
Check out a	Create a working copy of a local	<code>git clone /path/to/repository</code>

repository	repository:	
	For a remote server, use:	<code>git clone username@host:/path/to/repository</code>
Add files	Add one or more files to staging (index):	<code>git add <filename></code> <code>git add *</code>
	Commit changes to head (but not yet to the remote repository):	<code>git commit -m "Commit message"</code>
Commit	Commit any files you've added with <code>git add</code> , and also commit any files you've changed since then:	<code>git commit -a</code>
Push	Send changes to the master branch of your remote repository:	<code>git push origin master</code>
Status	List the files you've changed and those you still need to add or commit:	<code>git status</code>
Connect to a remote repository	If you haven't connected your local repository to a remote server, add the server to be able to push to it:	<code>git remote add origin <server></code>

	List all currently configured remote repositories:	<code>git remote -v</code>
Branches	Create a new branch and switch to it:	<code>git checkout -b <branchname></code>
	Switch from one branch to another:	<code>git checkout <branchname></code>
	List all the branches in your repo, and also tell you what branch you're currently in:	<code>git branch</code>
	Delete the feature branch:	<code>git branch -d <branchname></code>
	Push the branch to your remote repository, so others can use it:	<code>git push origin <branchname></code>
	Push all branches to your remote repository:	<code>git push --all origin</code>
	Delete a branch on your remote repository:	<code>git push origin :<branchname></code>
Update from the remote repository	Fetch and merge changes on the remote server to your working directory:	<code>git pull</code>
	To merge a different branch	<code>git merge <branchname></code>

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

content in head:

Changes already
added to the
index, as well as
new files, will be
kept.

Instead, to drop
all your local
changes and
commits, fetch the
latest history from
the server and
point your local
master branch at
it, do this:



```
git fetch origin
```

```
git reset --hard origin/master
```

Search

Search the
working directory
for `foo()`:

```
git grep "foo()"
```

Was this helpful?  [Yes](#)  [No](#)

