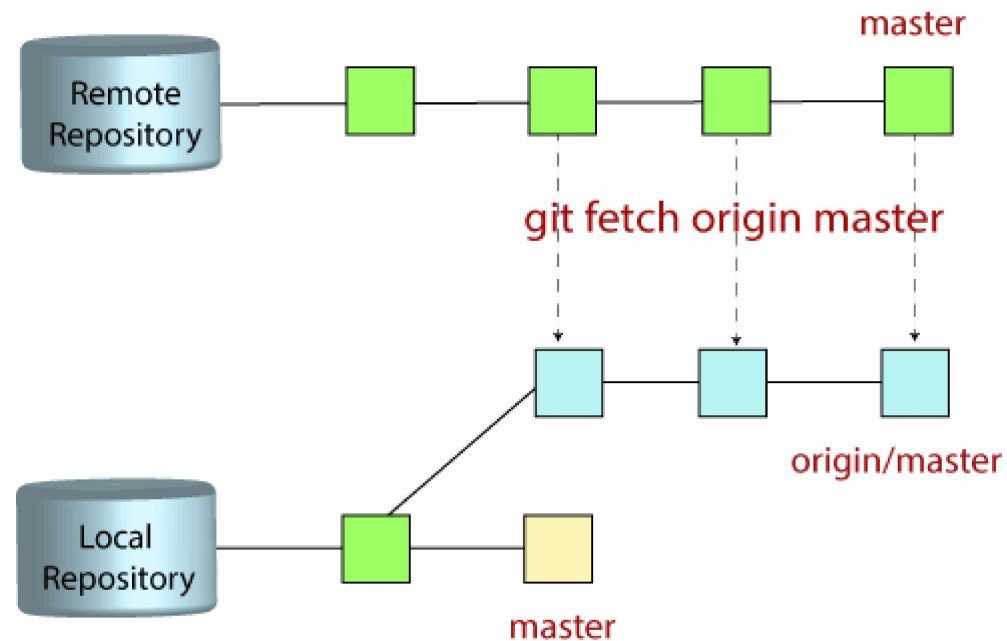


Git Fetch

Git "fetch" Downloads commits, objects and refs from another repository. It fetches branches and tags from one or more repositories. It holds repositories along with the objects that are necessary to complete their histories to keep updated remote-tracking branches.

ADVERTISEMENT



The "git fetch" command

The "**git fetch**" command is used to pull the updates from remote-tracking branches. Additionally, we can get the updates that have been pushed to our remote branches to our local machines. As we know, a branch is a variation of our repositories main code, so the remote-tracking branches are branches that have been set up to pull and push from remote repository.

How to fetch Git Repository

We can use fetch command with many arguments for a particular data fetch. See the below scenarios to understand the uses of fetch command.

Scenario 1: To fetch the remote repository:

We can fetch the complete repository with the help of fetch command from a repository URL like a pull command does. See the below output:

Syntax:

```
$ git fetch < repository Url>
```

Output:

ADVERTISEMENT

```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/Git-Example (master)
$ git fetch https://github.com/ImDwivedi1/Git-Example.git
warning: no common commits
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
From https://github.com/ImDwivedi1/Git-Example
 * branch                HEAD          -> FETCH_HEAD

HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/Git-Example (master)
$ |
```

In the above output, the complete repository has fetched from a remote URL.

Scenario 2: To fetch a specific branch:

We can fetch a specific branch from a repository. It will only access the element from a specific branch. See the below output:

Syntax:

```
$ git fetch <branch URL> <branch name>
```

Output:

```

HiManshu@HiManshu-PC MINGW64 ~/Desktop/Git-Example (master)
$ git fetch https://github.com/ImDwivedi1/Git-Example.git Test
warning: no common commits
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
From https://github.com/ImDwivedi1/Git-Example
 * branch                Test          -> FETCH_HEAD

HiManshu@HiManshu-PC MINGW64 ~/Desktop/Git-Example (master)
$ |

```

In the given output, the specific branch **test** has fetched from a remote URL.

Scenario 3: To fetch all the branches simultaneously:

The git fetch command allows to fetch all branches simultaneously from a remote repository. See the below example:

Syntax:

```
$ git fetch -all
```

Output:

```

HiManshu@HiManshu-PC MINGW64 ~/Desktop/Git-Example (master)
$ git fetch --all
Fetching origin
From https://github.com/ImDwivedi1/Git-Example
 * [new branch]      master    -> origin/master
 * [new branch]      Test      -> origin/Test

HiManshu@HiManshu-PC MINGW64 ~/Desktop/Git-Example (master)
$

```

In the above output, all the branches have fetched from the repository Git-Example.

Scenario 4: To synchronize the local repository:

Suppose, your team member has added some new features to your remote repository. So, to add these updates to your local repository, use the git fetch command. It is used as follows.

Syntax:

```
$ git fetch origin
```

Output:

```
HiManshU@HiManshU-PC MINGW64 ~/Desktop/Git-Example (master)
$ git fetch origin

HiManshU@HiManshU-PC MINGW64 ~/Desktop/Git-Example (master)
$ git fetch origin
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/ImDwivedi1/Git-Example
* [new branch]      test2      -> origin/test2

HiManshU@HiManshU-PC MINGW64 ~/Desktop/Git-Example (master)
$ |
```

In the above output, new features of the remote repository have updated to my local system. In this output, the branch **test2** and its objects are added to the local repository.

The git fetch can fetch from either a single named repository or URL or from several repositories at once. It can be considered as the safe version of the git pull commands.

The git fetch downloads the remote content but not update your local repo's working state. When no remote server is specified, by default, it will fetch the origin remote.

Differences between git fetch and git pull

To understand the differences between fetch and pull, let's know the similarities between both of these commands. Both commands are used to download the data from a remote repository. But both of these commands work differently. Like when you do a git pull, it gets all the changes from the remote or central repository and makes it available to your corresponding branch in your local repository. When you do a git fetch, it fetches all the changes from the remote repository and stores it in a separate branch in your local repository. You can reflect those changes in your corresponding branches by merging.

So basically,

```
git pull = git fetch + git merge
```

Git Fetch vs. Pull

Some of the key differences between both of these commands are as follows:

git fetch	git pull
Fetch downloads only new data from a remote repository.	Pull is used to update your current HEAD branch with the latest changes from the remote server.
Fetch is used to get a new view of all the things that happened in a remote repository.	Pull downloads new data and directly integrates it into your current working copy files.
Fetch never manipulates or spoils data.	Pull downloads the data and integrates it with the current working file.
It protects your code from merge conflict.	In git pull, there are more chances to create the merge conflict .
It is better to use git fetch command with git merge command on a pulled repository.	It is not an excellent choice to use git pull if you already pulled any repository.