

Set up a repository

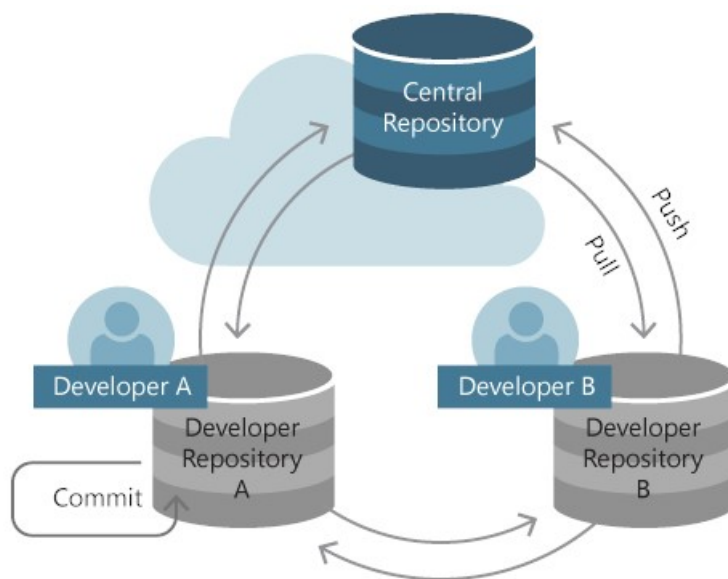
04/03/2017 • 3 minutes to read

By: Robert Outlaw

A Git repository, or repo, is a folder that you've told Git to help you track file changes in. You can have any number of repos on your computer, each stored in their own folder. Each Git repo on your system is independent, so changes saved in one Git repo don't affect the contents of another.

A Git repo contains every version of every file saved in the repo. This is different than other version control systems that store only the differences between files. Git stores the file versions in a hidden `.git` folder alongside other information it needs to manage your code. Git saves these files very efficiently, so having a large number of versions doesn't mean that it uses a lot of disk space. Storing each version of your files helps Git merge code better and makes working with multiple versions of your code quick and easy.

You work with Git through commands issued while working in your local repo on your computer. Even when you're sharing code or getting updates from your team, it's done from commands that update your local repo. This local-focused design is what makes Git a distributed version control system. Every repo is self-contained, and the owner of the repo is responsible for keeping it up to date with the changes from others.



Most teams will use a central repo hosted on a server everyone can access to coordinate their changes. The central repo is usually hosted in a source control management solution, like Team Foundation Server or Azure DevOps, that adds features that make working together easier.

Create a new Git repo

You have two options to create a Git repo—you can create a repo from the code in a folder on your computer, or you can clone the contents of an existing repo. Deciding which option you need is straightforward—if you're working with code that's just on your own computer, create a local repo using the code in that folder. Most of the time you're working with code already shared in a Git repo, so cloning the existing repo to your computer is the way to go.

From existing code

Use the `git init` command to create a new repo from an existing folder on your computer. From the command line, navigate to the root folder containing your code and run

```
> git init
```

to create the repo. You'll then need to add any files in the folder to your first commit with the following commands:

```
> git add --all
```

```
> git commit -m "Initial commit"
```

From a remote repository

Use the `git clone` command to copy the contents of an existing repo to a folder on your computer. From the command line, navigate to the folder you want to contain the cloned repo, then run:

```
> git clone https://fabrikam.visualstudio.com/DefaultCollection/Fabrikam/_git/FabrikamProject
```

Pass in the actual URL to the existing repo instead of the placeholder URL above. This URL, called the clone URL, points to a server where your team coordinates changes. You can get this URL from your team, or from the clone button on the web where you are browsing of the repo.

You don't need to add files or create an initial commit when you clone a repo since you copied all of the files and history of the existing repo during the clone.

[Azure Repos](#) provides unlimited free private Git repos. [Get started](#) today. Visual Studio user? Learn more about creating and cloning repos from Visual Studio in our [Team Services Git tutorial](#).



Get started with unlimited free private Git repos in [Azure Repos](#).



Robert is a content developer at Microsoft working on Azure DevOps and Team Foundation Server.