## Repository

A repository is a location for your code managed by version control. Azure Repos supports both [Git](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#git) and [TFVC](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#team-foundation-version-control-tfvc).

**Clone**

Create a complete local copy of an existing Git repo by cloning it. Cloning a repo downloads all [commits](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#commit) and [branches](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#branch) in the repo and sets up a named relationship with the existing repo that you cloned. Use this relationship to interact with the existing repo, [pushing](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#push) and [pulling](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#pull) changes to share code with your team.

## Branch

Branches are lightweight references that keep a history of commits and provide a way to isolate changes for a feature or a bug fix from your main branch and other work.

Committing changes to a branch doesn't affect other branches. You can push and share branches with other people on your team without having to merge the changes into main.

Switching between branches is quick and easy. Git doesn't create multiple copies of your source code when you're working with branches - it uses the history information stored in commits to re-create the files in a branch when you start working on it.

## Commit

A commit is a group of changes saved to your local repository. You can share these changes to the remote repository by [pushing](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#push).

## Fork

A fork is a complete copy of a repository, including all files, commits, and (optionally) branches.

## Pull

A pull command updates the code in your local repository with the changes from other members of your team that are in the remote repository.

## Push

Share changes made in commits and branches by using the push command.

When you push, Git uploads the saved commits in your checked branch to the remote repository. If the branch exists on the remote repository, Git takes the [commits](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#commit) and adds them to that branch on the remote repository. If that branch doesn't exist, Git creates a new branch with the same commits as your local branch.

## Git workflow

Version control has a general workflow that most developers use when writing code and sharing it with the team. These steps are:

1. Get a local copy of code if they don't have one yet.
2. Make changes to code to fix bugs or add new features.
3. When the code is ready, make it available for your team to review.
4. After the code is reviewed, merge it into the team's shared codebase.

Git has a version of this workflow that uses terminology and commands such as repositories, branches, commits, and pull requests. These terms might sound familiar if you've used a version control system like Team Foundation Version Control (TFVC) or Subversion, but they behave differently in Git.

1. [Create a branch](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#branch) for the changes you plan to make and give it a name, such as users/jamal/fix-bug-3214 or features/cool-feature.
2. [Commit changes](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#commit) to your branch. People often have multiple commits for a bug fix or feature.
3. [Push your branch](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#push) to the remote repository.
4. [Create a pull request](https://docs.microsoft.com/en-us/azure/devops/repos/get-started/key-concepts-repos?view=azure-devops#pull-request) so other people can review your changes. To incorporate feedback, you might need to make more commits and push more changes. When the code is ready, complete the pull request and merge your code into the target branch, such as main.

## Public projects

A project created within an Azure DevOps Services organization that is visible to the whole world. Everyone in the world can discover them and perform limited operations. You can use the [Azure DevOps CLI to discover a list of projects](https://docs.microsoft.com/en-us/azure/devops/organizations/projects/create-project?view=azure-devops#list-and-connect-to-projects). Administrators can control who gets to fully contribute. Administrators can switch a project from private to public, and vice-versa.

## Projects

A project, which was previously known as a team project, provides a repository for source code. A project provides a place where a group of people can plan, track progress, and collaborate on building software solutions. A project is defined for an Azure DevOps Services organization You can use it to focus on those objects defined within the project.