

TOPICS

Introduction to Git ▼

IN THIS ARTICLE

Set up Git in VS Code ▼

(<https://vscode.dev/github/microsoft/vscode-docs/blob/main/docs/sourcecontrol/intro-to-git.md>)

Introduction to Git in VS Code

Want to easily manage your source code and collaborate with others? Git and GitHub are the tools you need! And with Visual Studio Code, you can set up and use them in a snap. Even if you're a beginner, VS Code's user-friendly interface guides you through common Git actions like pushing and pulling code, creating and merging branches, and committing code changes. And if you're a pro, you'll love the ability to perform Git actions directly within the editor, saving you time and effort compared to using the Git command line. Plus, the seamless workflow between VS Code and Git means you can stay in your editor and get more done.

With GitHub Copilot in VS Code, you can get suggestions for commit messages, pull requests, and review code changes before committing them. Learn more about [GitHub Copilot in VS Code](/docs/copilot/overview) (</docs/copilot/overview>).



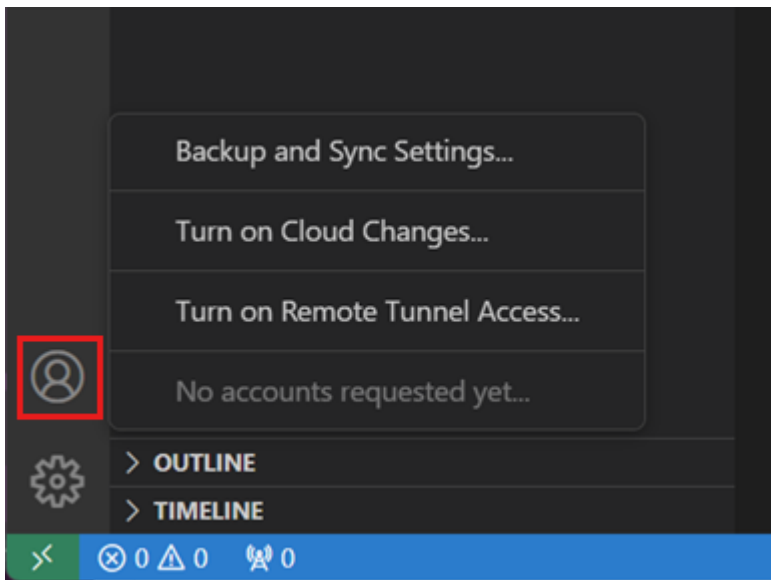
Tip

If you don't have a Copilot subscription yet, use Copilot for free by signing up for the [Copilot Free plan](https://github.com/github-copilot/signup) (<https://github.com/github-copilot/signup>). You'll get a monthly limit of completions and chat interactions.

Set up Git in VS Code

To use Git and GitHub in VS Code, first make sure you [have Git installed on your computer](https://git-scm.com/downloads) (<https://git-scm.com/downloads>). If Git is missing, the **Source Control** view shows instructions on how to install it. Make sure to restart VS Code afterwards.

Additionally you can sign into VS Code with your GitHub account in the **Accounts** menu in the lower right of the Activity bar to enable additional features like [Settings Sync](/docs/editor/settings-sync) (</docs/editor/settings-sync>), but also cloning and publishing repositories from GitHub.



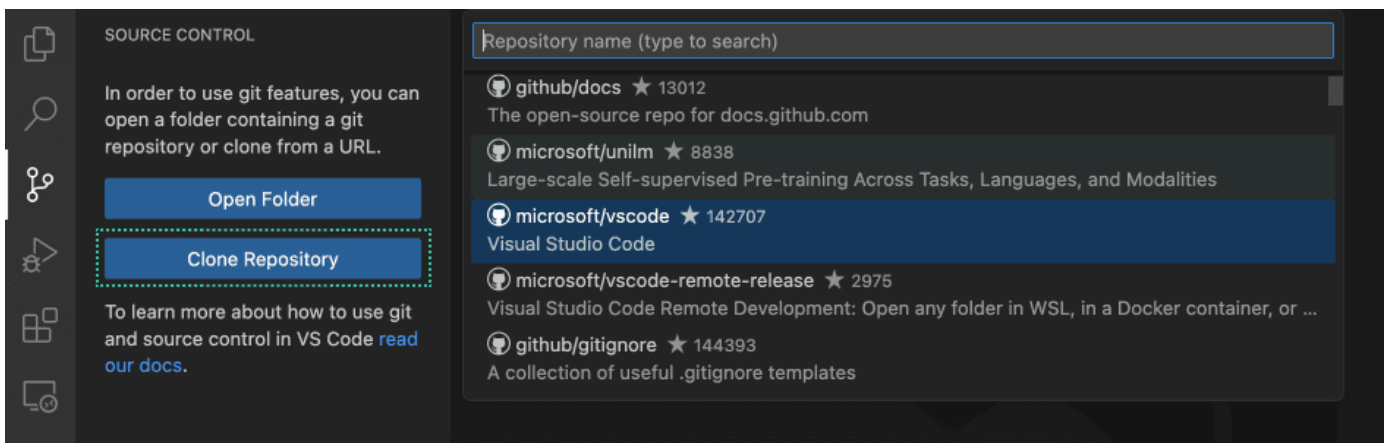
Open a Git repository

VS Code provides several ways to get started in a Git repository, from local to remote cloud-powered environments like [GitHub Codespaces](https://github.com/features/codespaces) (<https://github.com/features/codespaces>).

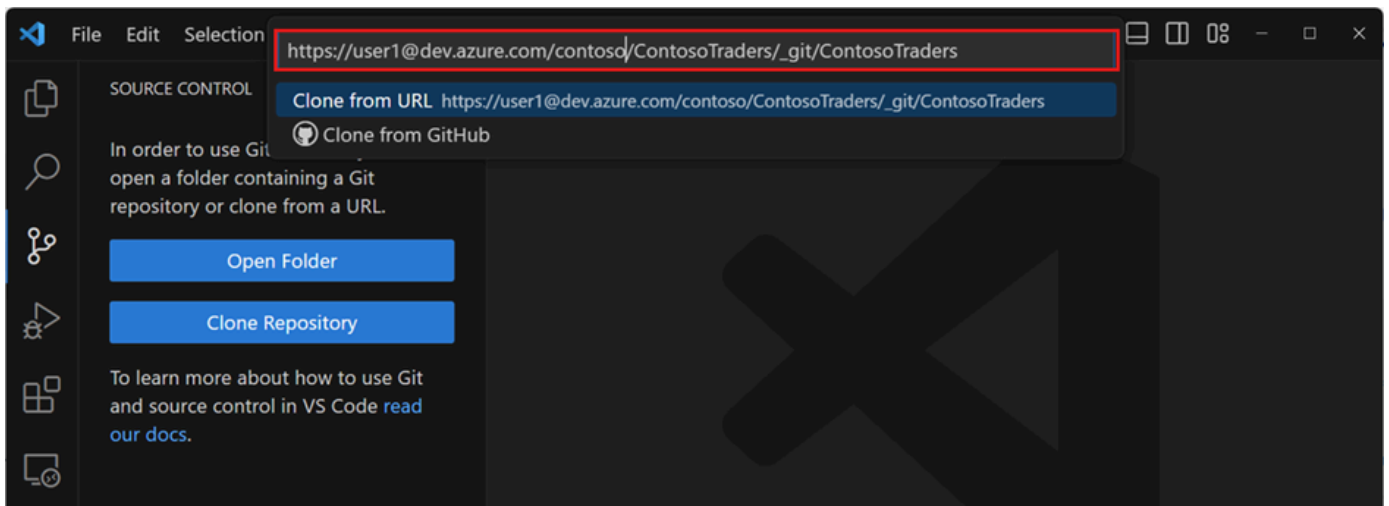
Clone a repository locally

To clone a repository, run the **Git: Clone** command in the Command Palette (`Ctrl+Shift+P`), or select the **Clone Repository** button in the **Source Control** view.

If you clone from GitHub, VS Code prompts you to authenticate with GitHub. Then, select a repository from the list to clone to your machine. The list contains both public and private repositories.



For other Git providers, enter the repository URL, select **Clone**, and pick a folder on your local machine to clone the files into. VS Code opens the folder once the repository is cloned on your local machine.



Initialize a repository in a local folder

To initialize a new local Git repository:

- 1 Pick an existing or new folder on your computer and open it in VS Code.
- 2 In the **Source Control** view, select the **Initialize Repository** button.

This creates a new Git repository in the current folder, allowing you to start tracking code changes.

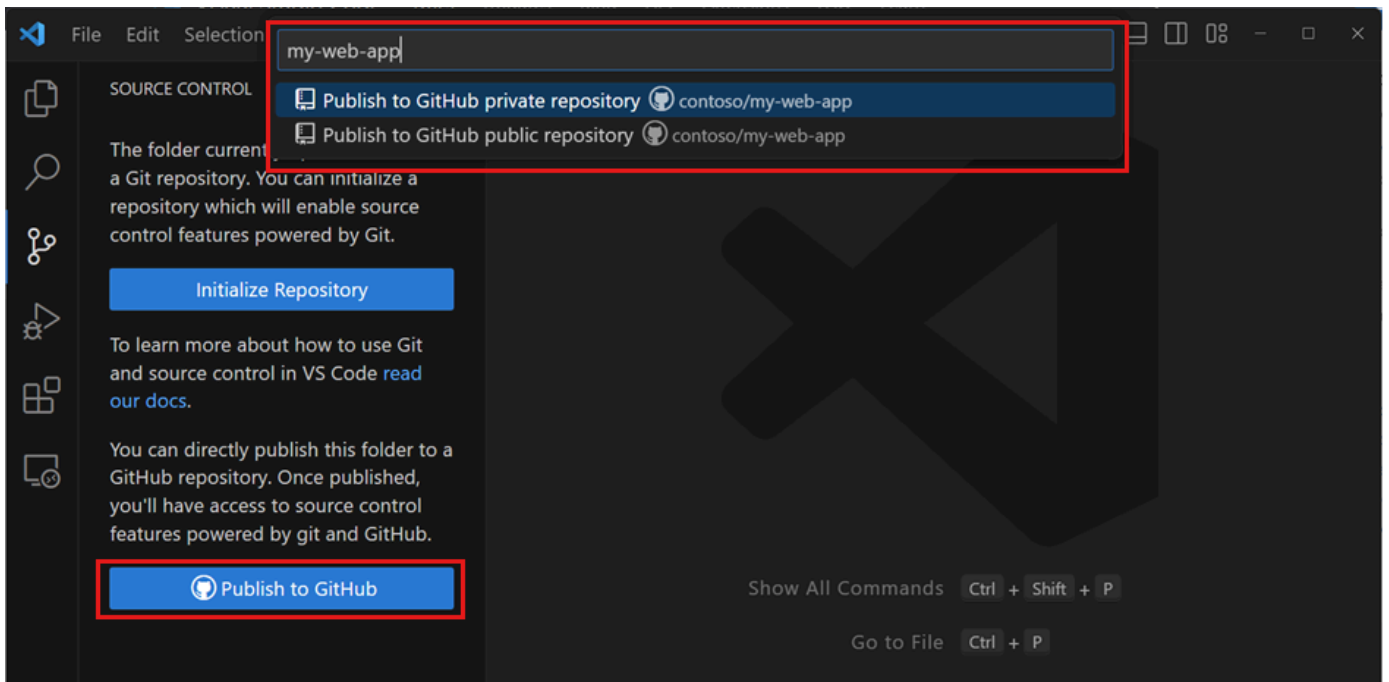
This action is equivalent to running `git init` on the command-line.



Publish local repository to GitHub

You can also initialize a local repository and publish it directly to GitHub. This creates a new repository on your GitHub account, and pushes your local code changes to the remote repository. Having your source code on a remote repository is a great way to back up your code, collaborate with others, and automate your workflow with [GitHub Actions](https://github.com/features/actions) (<https://github.com/features/actions>).

Use the **Publish to GitHub** command button in the **Source Control** view. You can then choose a name and description for the repository, and whether to make it public or private.



Once the repository has been created, VS Code pushes your local code to the remote repository. Your code is now backed up on GitHub, and you can start collaborating with others with commits and pull requests.

Open a GitHub repository in a codespace

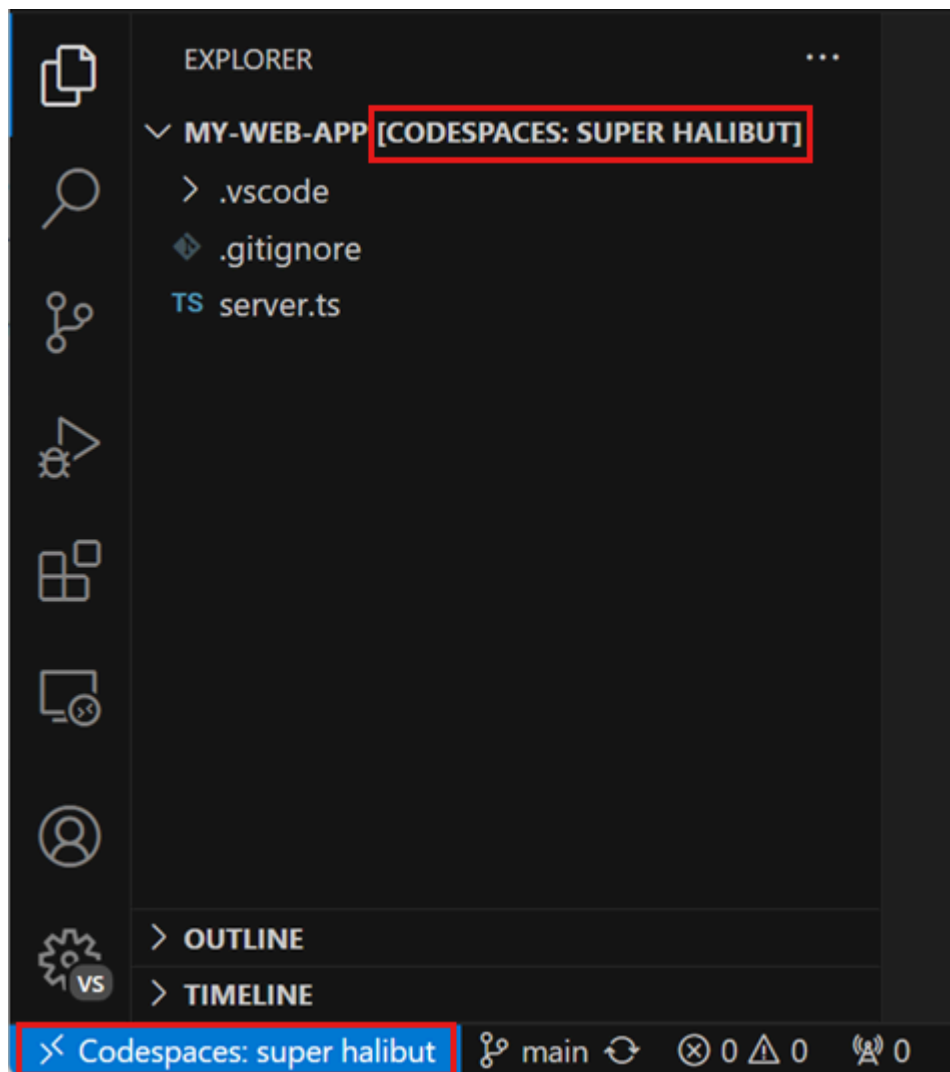
GitHub Codespaces (<https://github.com/features/codespaces>) lets you open a GitHub repository in a fully configured cloud-based development environment, enabling you to develop in a browser without having to install any software on your local computer. GitHub Codespaces allows free usage for individuals, which makes it easy to get started working on open source projects.

To create a codespace for your GitHub repository:

- 1 Install the [GitHub Codespaces](https://marketplace.visualstudio.com/items?itemName=GitHub.codespaces) (<https://marketplace.visualstudio.com/items?itemName=GitHub.codespaces>) extension in VS Code and sign in with your GitHub account.
- 2 Run the **Codespaces: Create New Codespace** command.
- 3 Select the repository and branch you want to open.

VS Code opens a new window, which is connected to the codespace. The source code, terminal, and running and debugging are hosted in the remote cloud-based development environment.

Notice that the File Explorer and Status Bar indicate that the workspace is opened in a codespace.



Alternatively, you can also start from a codespace template on the [GitHub Codespaces website](https://github.com/codespaces/templates) (<https://github.com/codespaces/templates>).

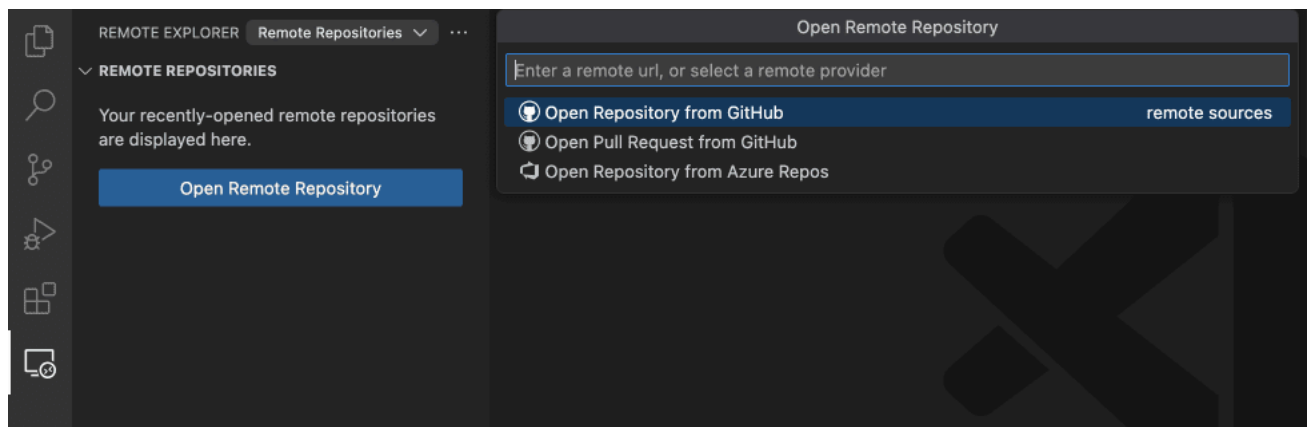
If you already have a codespace open in your browser, run the **Codespaces: Open in VS Code Desktop** command in the browser to connect to the codespace from your local VS Code Desktop.

You can learn more about GitHub Codespaces, including customization such as forwarding ports, in the [Developing in a codespace](https://docs.github.com/codespaces/developing-in-codespaces/developing-in-a-codespace?tool=vscode) (<https://docs.github.com/codespaces/developing-in-codespaces/developing-in-a-codespace?tool=vscode>), documentation.

Open a GitHub repository remotely

VS Code's remote repository support allows you to browse and edit a GitHub repository without cloning it to your local computer. This is useful for quickly making changes to a remote repository without having to clone the entire codebase to your machine.

- 1 First install the [GitHub Repositories](https://marketplace.visualstudio.com/items?itemName=GitHub.remotehub) (<https://marketplace.visualstudio.com/items?itemName=GitHub.remotehub>), extension.
- 2 Run the command **Remote Repositories: Open Remote Repository...** or use the **Open Remote Repository** button the Explorer view.
- 3 Search and select the GitHub repository that you want to open.



💡 Tip

If you need to execute code or run terminal commands, you can seamlessly switch from a remote repository to a codespace with the command **Continue Working on**.

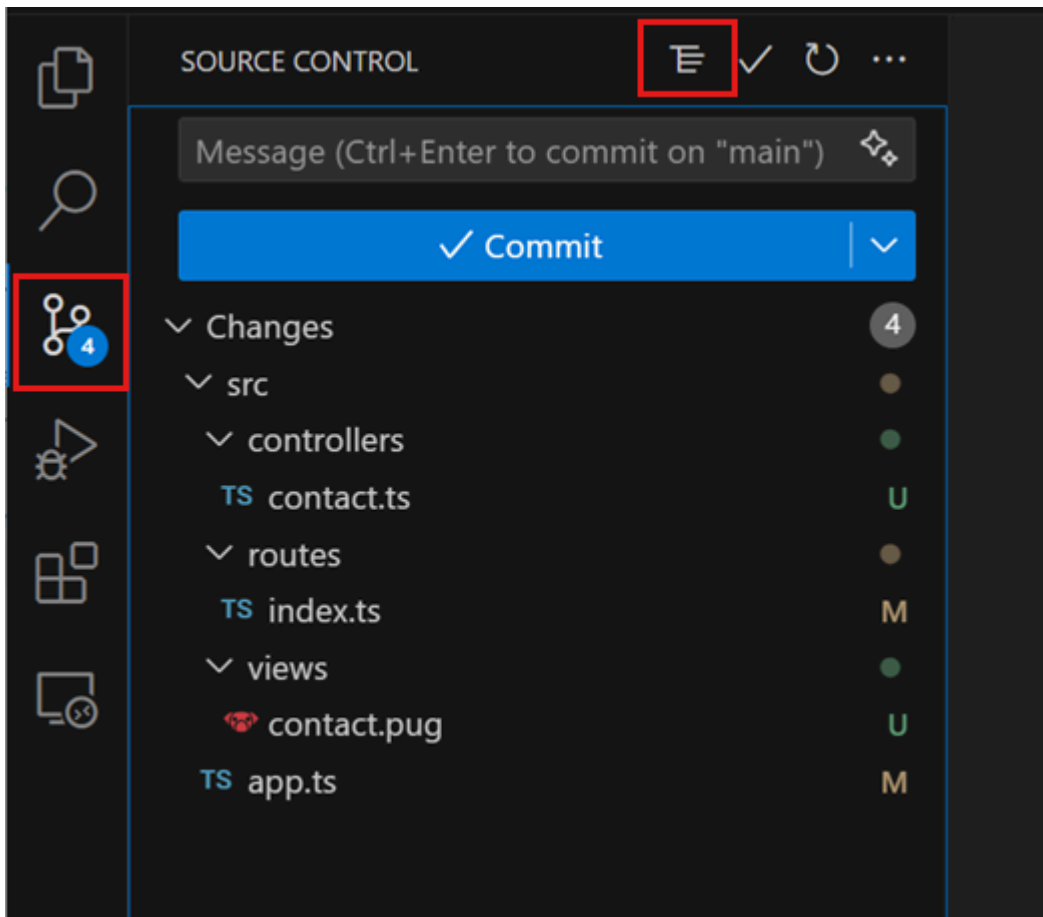
Staging and committing code changes

Once you have a Git repository set up, you can start tracking code changes by [staging and committing](https://git-scm.com/about/staging-area) (<https://git-scm.com/about/staging-area>), your newly created and edited code.

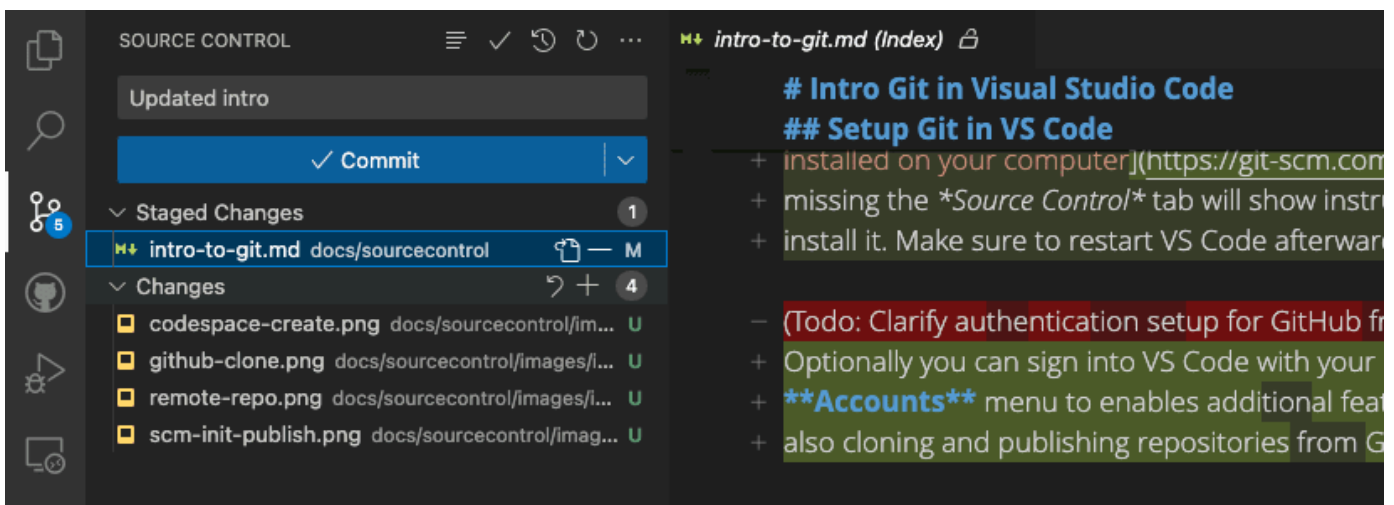
💡 Tip

Commit your changes early and often. This makes it easier to revert back to previous versions of your code if needed.

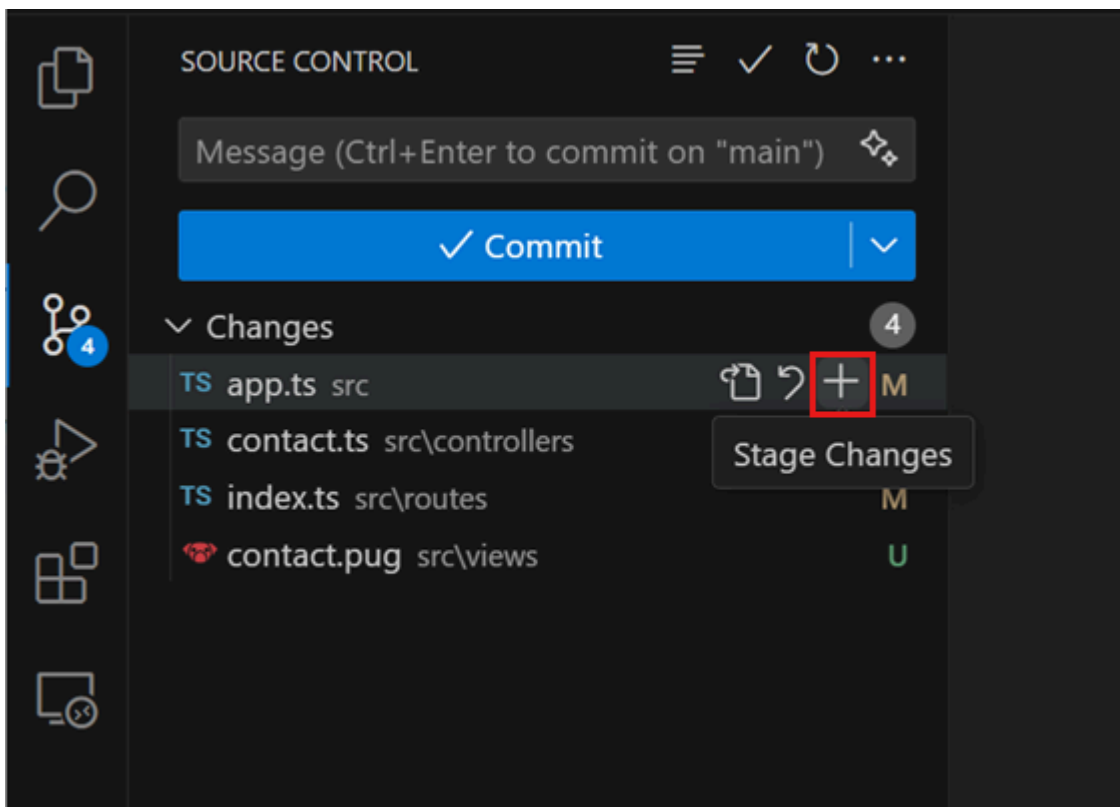
You can access the **Source Control** view from the Activity Bar to list all changed files in your workspace. You can toggle between a tree view or list view by using the tree/list icon in the Source Control view header.



When you select a file in the Source Control view, the editor shows a diff view that highlights the file changes, compared to the previously committed file.

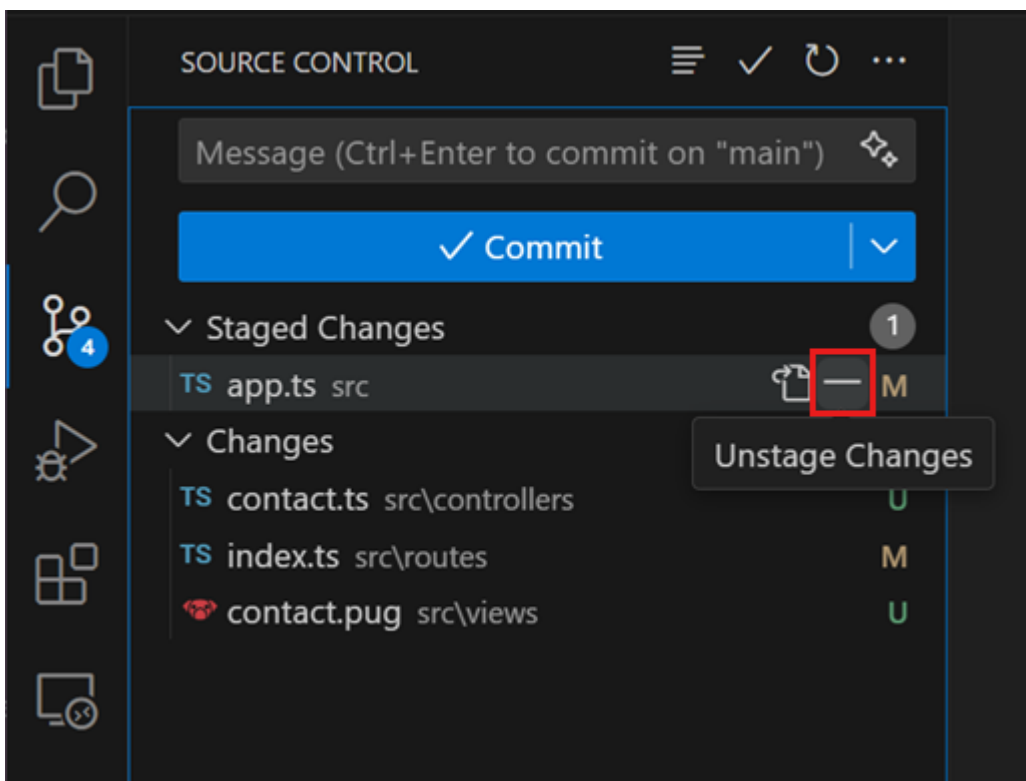


To stage a file, select the + (plus) icon next to the file in the **Source Control** view. This adds the file to the **Staged Changes** section, indicating that it will be included in the next commit.



You can also stage all pending changes at once by selecting the + (plus) icon next to **Changes** in the Source Control view.

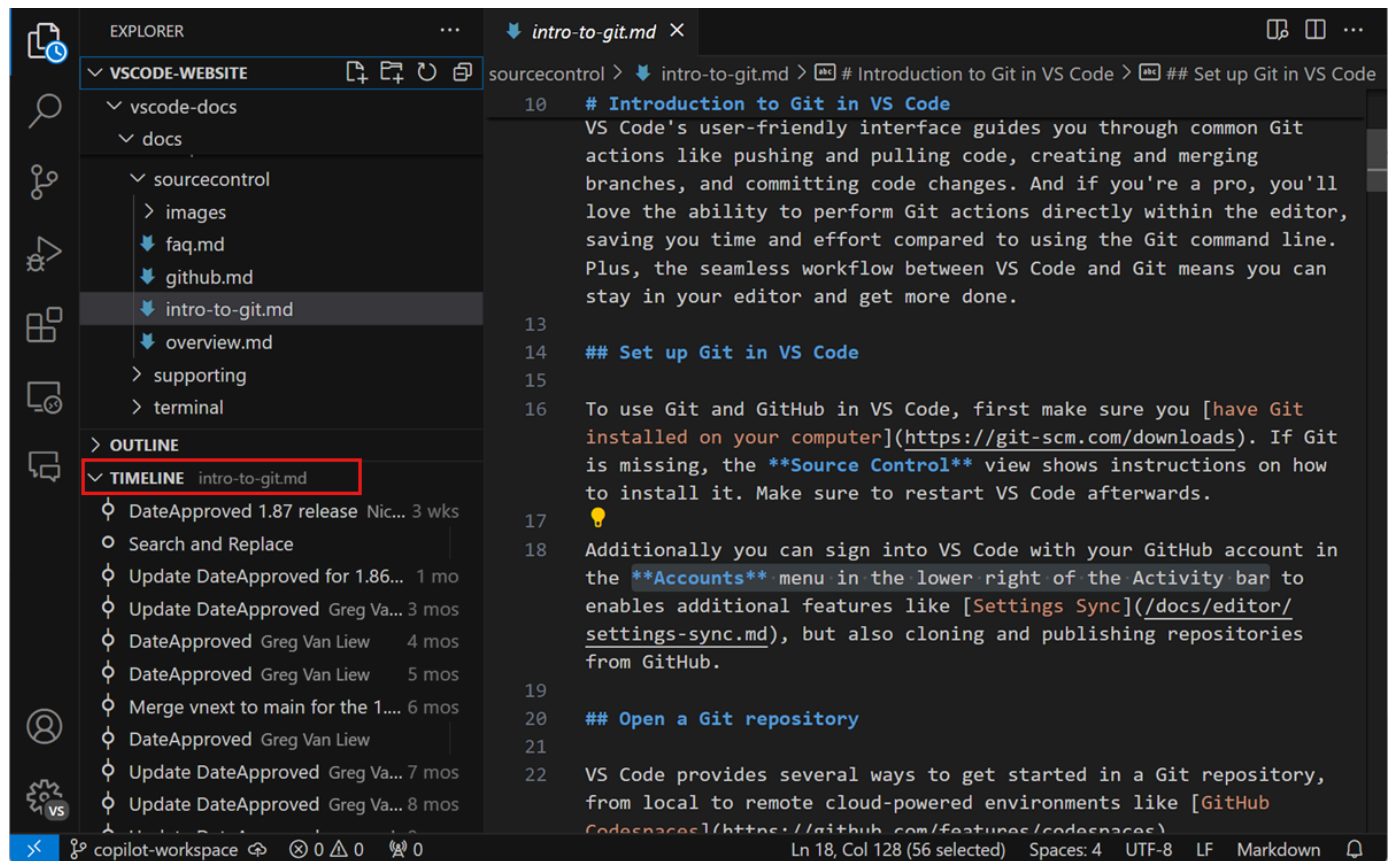
Staged changes can also be discarded by selecting the – (minus) icon next to the file. Similarly, you can discard all staged changes by selecting the – (minus) icon next to **Staged Changes** in the Source Control view.



To commit your staged changes, type a commit message in the upper text box, and then select the **Commit** button. This saves your changes to the local Git repository, allowing you to revert to previous versions of your code if needed.

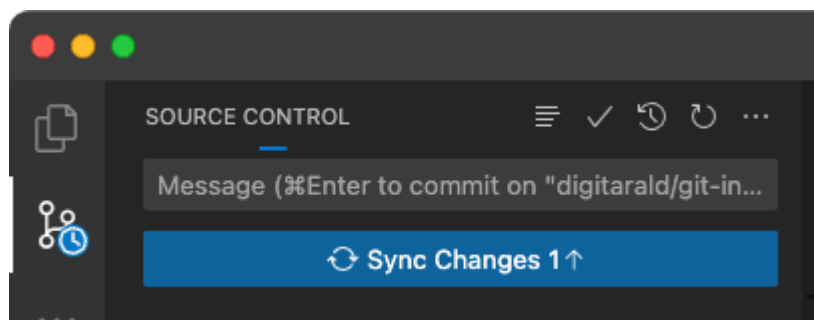
With GitHub Copilot in VS Code, you can generate a [commit message](#) ([/docs/copilot/overview#_productivity-improvements](#)) based on your code changes. Learn more about [generating commit messages with AI](#) ([/docs/sourcecontrol/overview#_generate-a-commit-message-with-ai](#)). Before you commit your changes, you can also let Copilot do a [code review of the uncommitted changes](#) ([/docs/sourcecontrol/overview#_review-uncommitted-code-changes-with-ai](#)).

You can navigate through and review all local file changes and commits in the **Timeline** view available in the bottom of the Explorer view.



Pushing and pulling remote changes

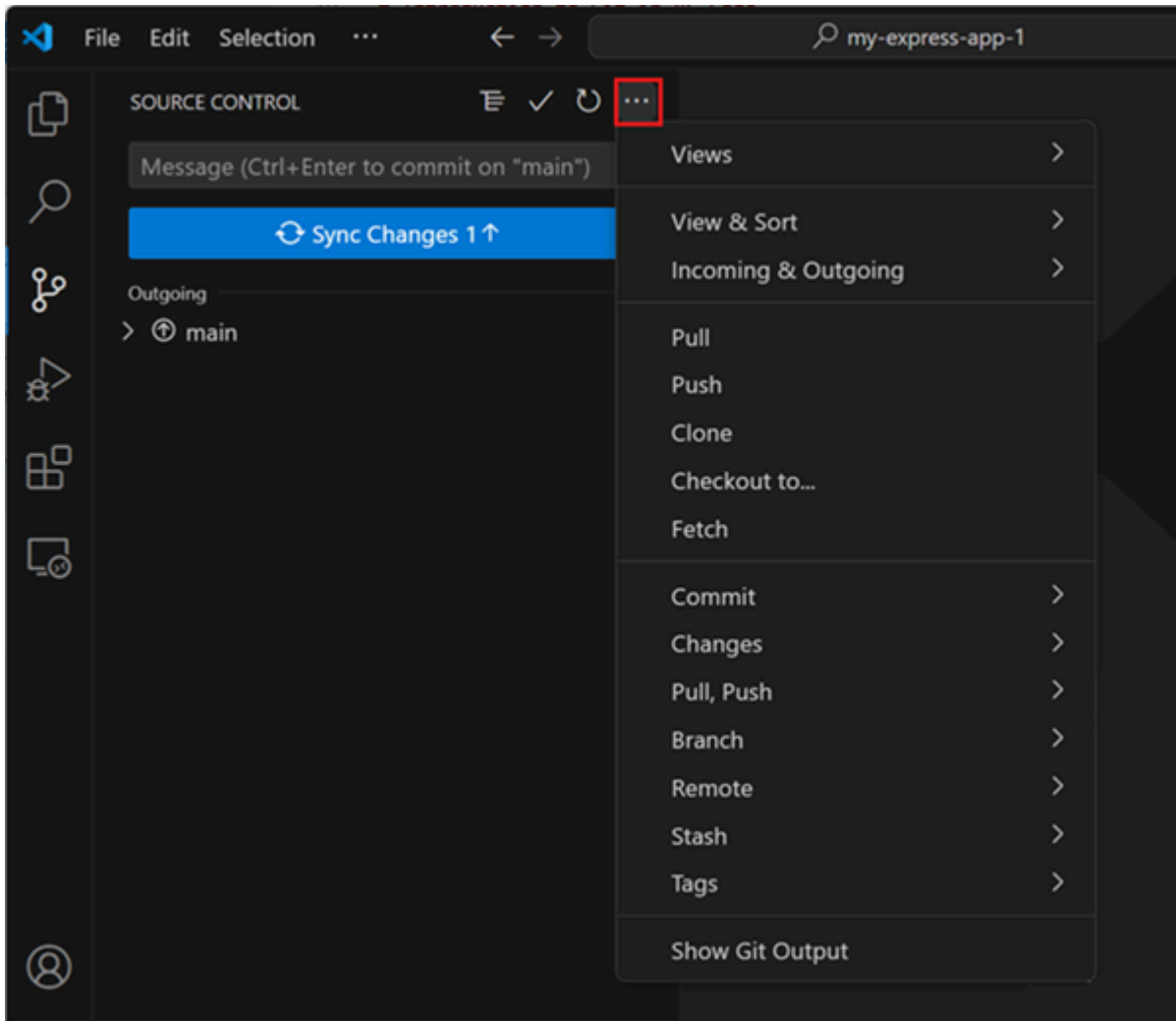
Once you have made commits to your local Git repository, you can push them to the remote repository. The **Sync Changes** button indicates how many commits are going to be pushed and pulled. Selecting the **Sync Changes** button downloads (pull) any new remote commits and uploads (push) new local commits to the remote repository.



 Tip

You can enable the **Git: Autofetch** setting (</docs/getstarted/settings>) to always get an up-to-date remote commit indicator.

Push and pull can also be performed individually by using their respective commands. You can access these commands from the Source Control menu.



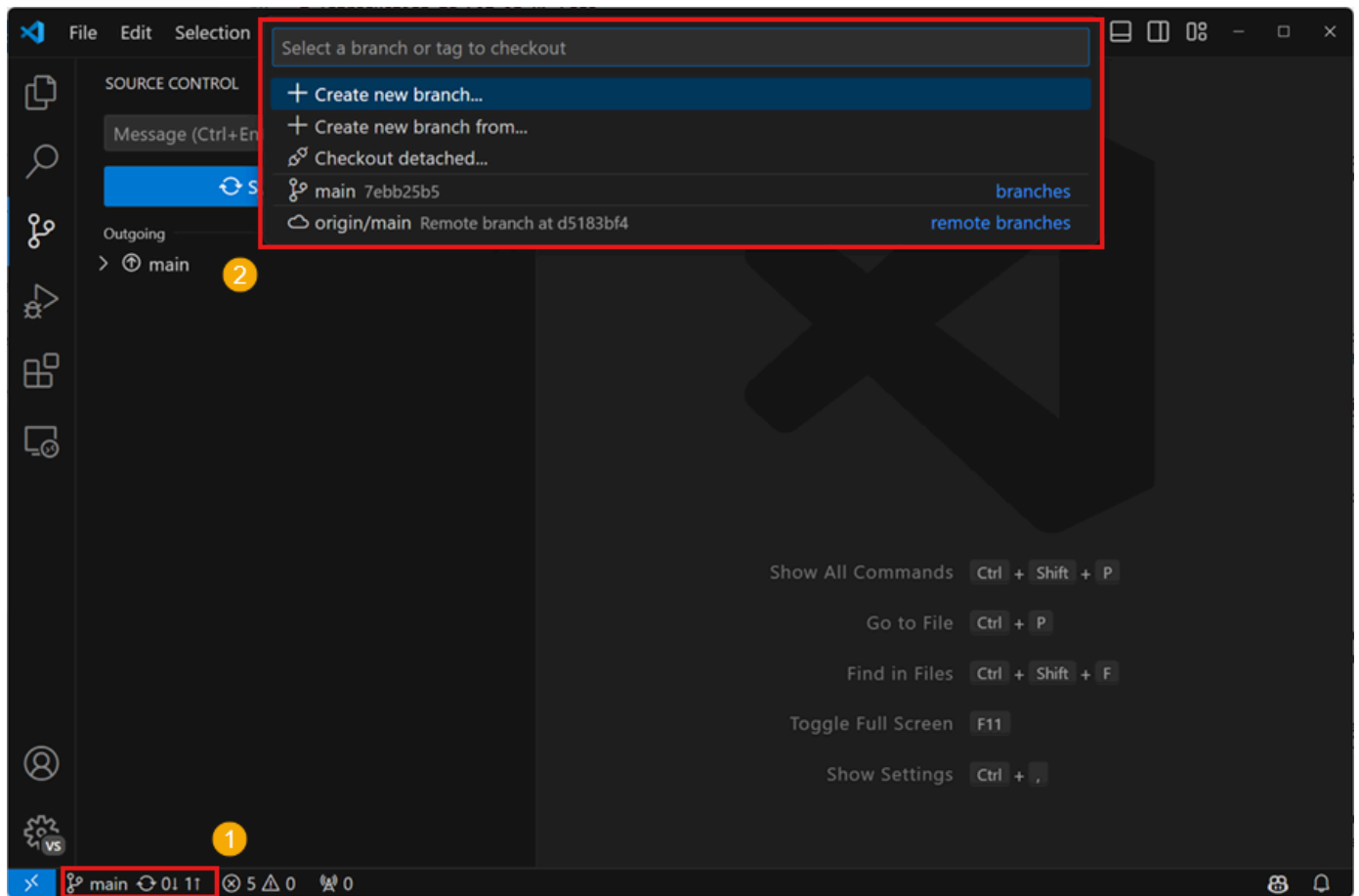
Using branches

In Git, [branches](https://docs.github.com/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches) (<https://docs.github.com/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches>) allow you to work on multiple versions of your codebase simultaneously. This is useful for experimenting with new features or making large code changes without affecting the main codebase.

The branch indicator in the Status bar shows the current branch and lets you switch to new and existing branches.



To create a new branch, select the branch indicator and choose to create it from the current branch or another local one. Type a name for the new branch, and confirm. VS Code creates a new branch and switches to it, allowing you to make changes to your code without affecting the main branch.



💡 Tip

If you use the [GitHub Pull Requests and Issues](https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github) (<https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github>) extension, you can create a branch directly from an issue, which gets you started working in a new local branch and automatically prefills the pull request for you.

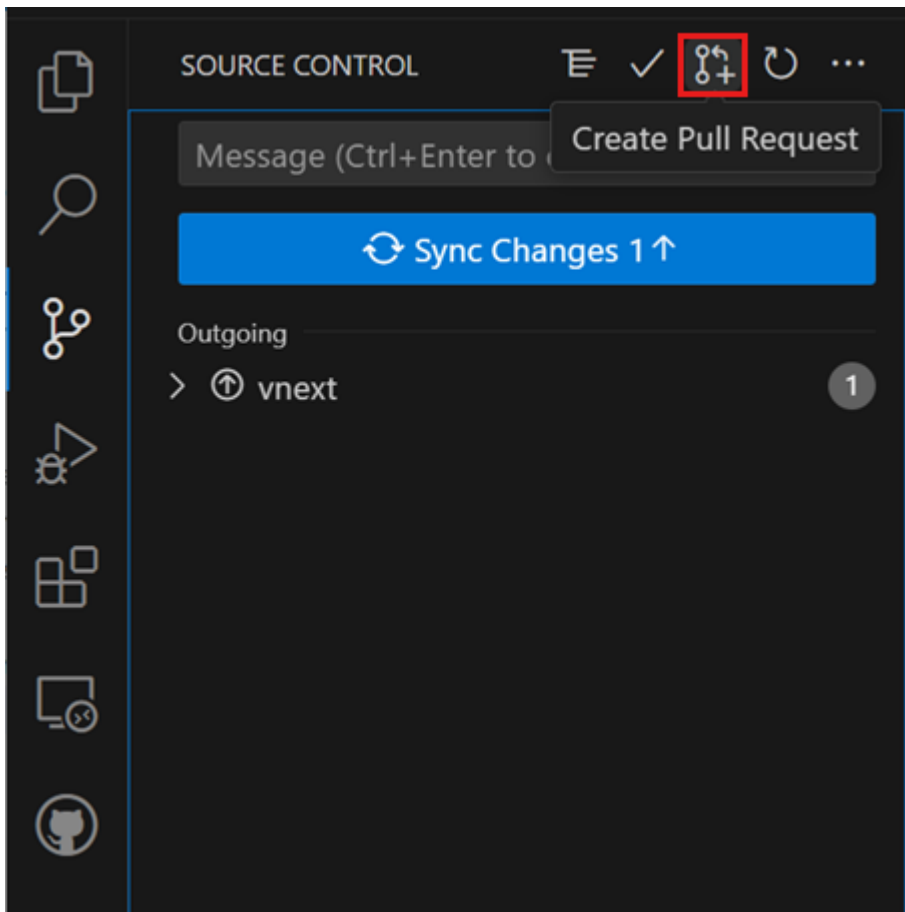
After you create a local branch, you can push the branch to the remote repository by selecting **Publish Branch** in the **Source Control** view. This creates a new branch on the remote repository, allowing you to collaborate with others in that branch.

Creating and reviewing GitHub pull requests

In Git and GitHub, [pull requests \(PRs\)](https://docs.github.com/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests) (<https://docs.github.com/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>) are a way for collaborators to review and merge code changes from separate branches into the main branch. This enables teams to review and approve code changes before they are incorporated into the main codebase, ensuring that only high-quality changes are merged.

To use pull requests in VS Code, you need to install the [GitHub Pull Requests and Issues](https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github) (<https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github>) extension. This extension adds PR and issue tracking functionality to VS Code, allowing you to create, review, and merge PRs from within the editor.

To create a PR, make sure you are on a separate branch from the main branch, and push your code changes to the remote repository. In the **Source Control** view, select the **Create Pull Request** button. This opens the PR creation form, where you can enter a title and description for the PR, and choose which branch to merge the changes into. Select **Create** to create the PR.



Tip

Use GitHub Copilot to generate a PR title and description, based on the commits that are included in the PR. Select the *sparkle* icon next to the PR title field to generate a PR title and description. You can also let Copilot perform a code review of the PR() before you create it. Select the **Copilot Code Review** button in the GitHub Pull Request view.

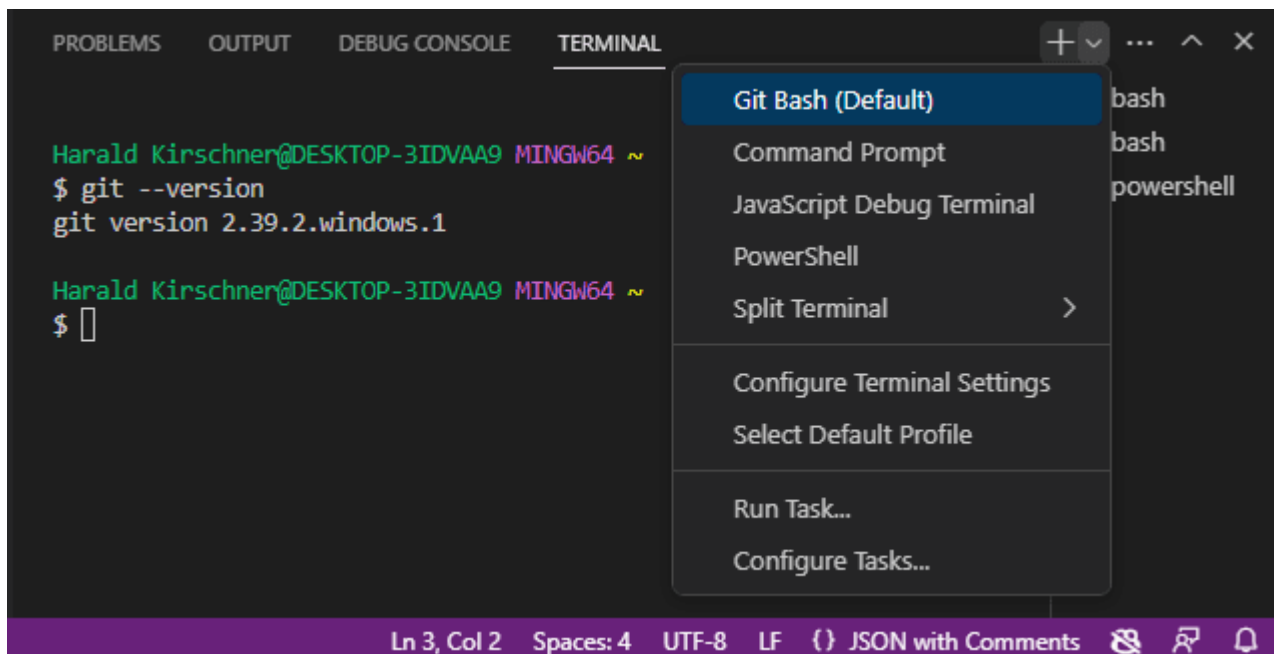
Learn more about pull requests in [VS Code's GitHub documentation \(/docs/sourcecontrol/github/\)](/docs/sourcecontrol/github/).

Using Git in the built-in terminal

As all Git state is kept in the local repository, you can easily switch between VS Code's UI, the [built-in terminal \(/docs/terminal/basics/\)](/docs/terminal/basics/), or external tools like [GitHub Desktop \(https://desktop.github.com\)](https://desktop.github.com). You can also set up [VS Code as your default Git editor \(/docs/sourcecontrol/overview#_vs-code-as-git-editor\)](/docs/sourcecontrol/overview#_vs-code-as-git-editor), allowing you to use VS Code to edit commit messages and other Git-related files.

Git Bash on Windows

Git Bash is a popular shell environment for Windows that provides a Unix-like command-line interface for working with Git and other command-line tools. Visual Studio Code's integrated terminal supports Git Bash as a shell, allowing you to seamlessly integrate Git Bash into your development workflow. Installing Git on your Windows machine also installs Git Bash, if it wasn't deselected during the installation steps.



Start by opening **View > Terminal** (`Ctrl+``). Select the dropdown arrow next to the `+` icon in the terminal panel to pick a new shell to open. If Git Bash is installed, it's shown in the list of terminals and shells. You can toggle between different terminals and shells in the Terminal sidebar. With Git Bash configured in Visual Studio Code, you can now use all of your favorite Git commands directly from the terminal in your code editor.

If you want to set Git Bash as your default shell, open the Terminal dropdown (next to the `+` icon) and select **Select Default Profile**. This opens a list of available shells, including Git Bash. Selecting *Git Bash* sets it as your default shell, and all future terminals will be opened with Git Bash. More advanced terminal tips are available in the [terminal documentation](/docs/terminal/basics) (</docs/terminal/basics>).

Was this documentation helpful?

Yes No

12/11/2024

 [Subscribe\(/feed.xml\)](/feed.xml)  [Ask questions\(https://stackoverflow.com/questions/tagged/vscode\)](https://stackoverflow.com/questions/tagged/vscode)

 [Follow @code\(https://go.microsoft.com/fwlink/?LinkID=533687\)](https://go.microsoft.com/fwlink/?LinkID=533687)

 [Request features\(https://go.microsoft.com/fwlink/?LinkID=533482\)](https://go.microsoft.com/fwlink/?LinkID=533482)

 [Report issues\(https://www.github.com/Microsoft/vscode/issues\)](https://www.github.com/Microsoft/vscode/issues)

 [Watch videos\(https://www.youtube.com/channel/UCs5Y5_7XK8HLDX0SLNwkd3w\)](https://www.youtube.com/channel/UCs5Y5_7XK8HLDX0SLNwkd3w)



(<https://www.microsoft.com>)

(<https://go.microsoft.com/fwlink/?LinkID=533687>)



(<https://github.com/microsoft/vscode>)



(<https://www.youtube.com/@code>)

Support (<https://support.serviceshub.microsoft.com/supportforbusiness/create?sapId=d66407ed-3967-b000-4cfb-2c318cad363d>)

Privacy (<https://go.microsoft.com/fwlink/?LinkId=521839>)

Terms of Use (<https://www.microsoft.com/legal/terms-of-use>) License (/License)