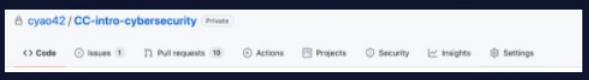
Managing a GitHub Repository

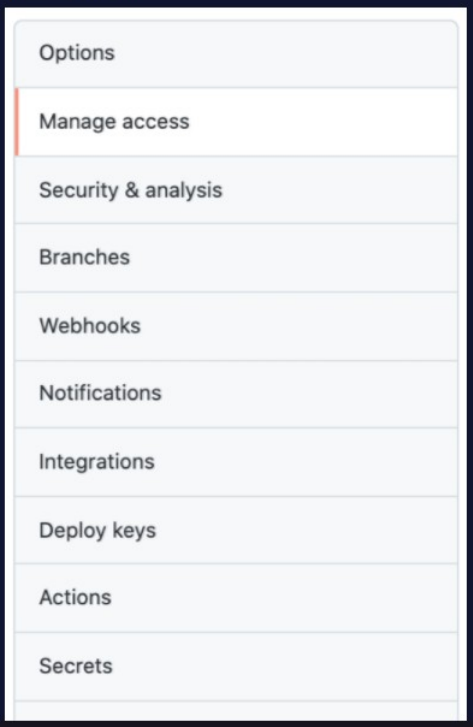
Learn how to manage GitHub repository settings and provide other users access to our repository!

GitHub Repository Settings



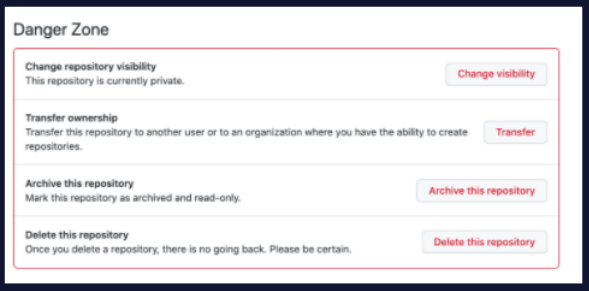
GitHub is a powerful code hosting platform for version control and collaboration. It allows multiple people to work on the same project from anywhere. However, that doesn't even begin to scratch the surface of what GitHub has to offer. In this article, we will explore settings to customize our GitHub repository as well as discuss features that will take our repository to the next level.

All GitHub settings can be accessed by clicking the **Settings** tab on the main page of our repository. You can follow along with this article by opening your own repository settings!



The Options tab allows us to change the basic repository information such as its name and social media banner. It's where we can enable or disable certain GitHub features like Wikis, Issues, Discussions, and more. We can also change merge options to only allow certain types of merges, or automatically delete head branches. Most importantly, there's the danger zone.

The Danger Zone



As the name suggests, the danger zone is where one should take caution when changing settings. We can make a repository private or public, transfer ownership to another user, and archive or delete the repository.

The Branches Tab

The settings' Branches tab is where we can set a default branch. The default branch is the branch against which all pull requests and code commits are automatically made (typically this is already set to the `main` branch). We can also protect our branches by adding rules that prevent branches from being deleted, disable force pushing, or require a pull request to be made before merging to a branch.

Multiple Choice Question

Which of the following settings is NOT possible to change in the Options tab of GitHub repository settings?

- Transfer repository ownership
- Change repository visibility to public
- Enable Discussions
- Change default branch

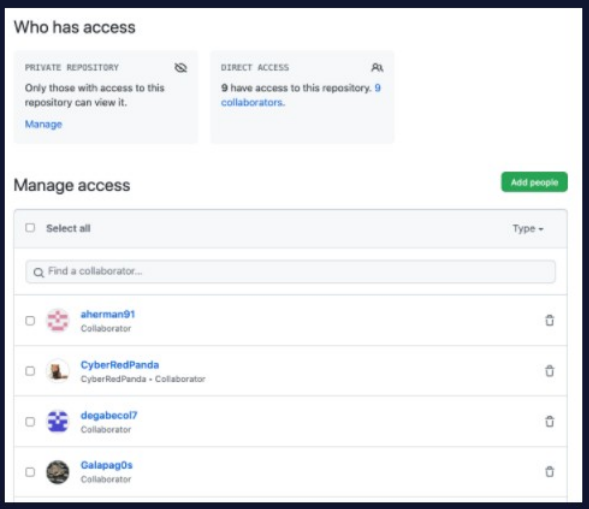
Managing Repository Access

By default, only the owner of the repository or the organization can configure the settings of a repository and access the tools. We have to go through the repository settings' Manage Access tab to grant other users or teams access. The specificity of permissions differs greatly between a user repository and an organization repository.

As always, we should use caution when granting permissions especially to those outside of our immediate organization or team. They can make undesired changes to the code, host pages or packages on our behalf, or even leak private source code.

User Repository

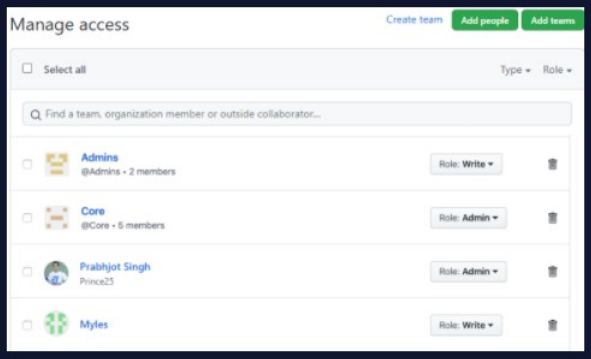
For a user repository, the owner can easily add another user by clicking the "Add people" button, searching their full name, email address, or username. The user will then have to accept the invitation. While the added user doesn't have the same permissions as the owner of the repository, the user can still rename a branch and publish packages among other things. For more information on the differences between the permission levels, check out GitHub Docs.



Organization Repository

GitHub Organizations provide its members a way to collaborate on multiple projects across multiple repositories. Owners or administrators of the organization can manage member access to the organization's repositories. Adding a member to a repository is no different from the process in a user repository with the exception of adding teams. In a GitHub Organization, admins can group members into teams and reference users by team names. You can read more about organization roles and their varying abilities on the GitHub

Docs.



Repository Roles

Businesses and institutions usually have a hierarchy that defines levels of authority and responsibilities. The same principles can be applied in a GitHub repository or organization; we don't want everyone to have admin privileges. Plus, it's strategic and secure to map levels of access.

You can check out GitHub Docs for a detailed breakdown of permissions for repository roles. You can also read about creating custom repository roles to have a configurable set of permissions with a role name of your choice.

Fill in the Code

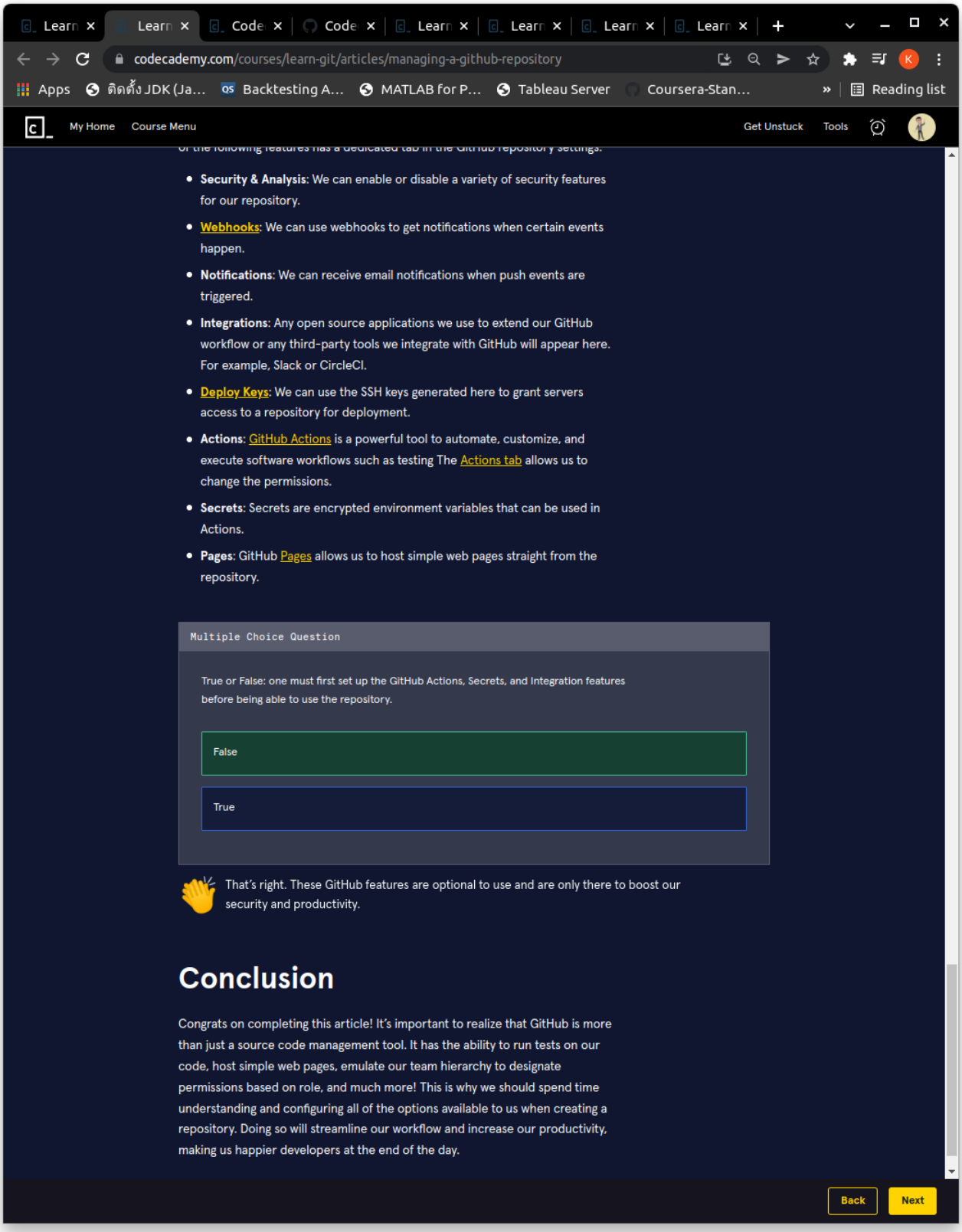
Complete the sentence below.

In the `organization` repository settings, users and `teams` can be added as collaborators.

You got it!

Other Repository Features and Tools

So far, we've already discussed the GitHub features we will be using most of the time. However, GitHub has a ton of advanced tools and settings we should definitely take advantage of. While we will not be diving deep into these, we still want you to be aware of them so you can revisit them when the need arises. Each of the following features has a dedicated tab in the GitHub repository settings:


codecademy.com/courses/learn-git/articles/managing-a-github-repository
My Home
Course Menu
Get Unstuck
Tools
Security & Analysis: We can enable or disable a variety of security features for our repository.
Webhooks: We can use webhooks to get notifications when certain events happen.
Notifications: We can receive email notifications when push events are triggered.
Integrations: Any open source applications we use to extend our GitHub workflow or any third-party tools we integrate with GitHub will appear here. For example, Slack or CircleCI.
Deploy Keys: We can use the SSH keys generated here to grant servers access to a repository for deployment.
Actions: GitHub Actions is a powerful tool to automate, customize, and execute software workflows such as testing The Actions tab allows us to change the permissions.
Secrets: Secrets are encrypted environment variables that can be used in Actions.
Pages: GitHub Pages allows us to host simple web pages straight from the repository.
Multiple Choice Question
True or False: one must first set up the GitHub Actions, Secrets, and Integration features before being able to use the repository.
False
True
That's right. These GitHub features are optional to use and are only there to boost our security and productivity.
Conclusion
Congrats on completing this article! It's important to realize that GitHub is more than just a source code management tool. It has the ability to run tests on our code, host simple web pages, emulate our team hierarchy to designate permissions based on role, and much more! This is why we should spend time understanding and configuring all of the options available to us when creating a repository. Doing so will streamline our workflow and increase our productivity, making us happier developers at the end of the day.
Back
Next