Table of Contents

[Main guide 2](#__RefHeading___Toc301_376676067)

[Members and outside collaborators 6](#__RefHeading___Toc303_376676067)

[About Teams 7](#__RefHeading___Toc305_376676067)

[Permissions per repo 8](#__RefHeading___Toc307_376676067)

[Necessary permissions 8](#__RefHeading___Toc309_376676067)

[Roles per organization 10](#__RefHeading___Toc311_376676067)

# Main guide

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Repository Visibility** | **User role** | **Contributor's Access Level** | **Participation Method** | **Owner's Actions Required** | **Description** |
| Public | Unauthenticated User | None | Unauthenticated users cannot participate. They need to create a GitHub account to contribute. | Unauthenticated users cannot participate. Encourage them to create a GitHub account to contribute. | Unauthenticated users cannot directly interact with the repository. They need to create a GitHub account to participate. |
| Public | Outside collaborators and organization members | Read Access | * Request permission to fork the repository. * Fork Repository & Submit Pull Request | No specific action required by the owner. Contributors can fork the repository, make changes, and submit pull requests from their forks. Owners should monitor and review incoming pull requests. [GitHub Docs](https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-a-pull-request-from-a-fork?utm_source=chatgpt.com) | Contributors can fork the repository to their own account, make changes in their fork, and then submit a pull request to propose merging their changes into the original repository. GitHub |
| Public |  | Write Access | * Request Write access to the owner * Create Branch & Submit Pull Request | Grant the contributor Write access to the repository: 1. Navigate to the repository's Settings. 2. Click on Manage access. 3. Click Invite a collaborator and enter the contributor's username or email. 4. Once accepted, assign the Write role to the contributor. [GitHub Docs](https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles?utm_source=chatgpt.com) | Contributors can create a new branch within the repository, make changes, and submit a pull request for review. GitHub Docs |
| Public |  | Admin Access | * Request Admin access to the owner Direct Push to main or Create Branch & Submit Pull Request | Grant the contributor Admin access to the repository: 1. Navigate to the repository's Settings. 2. Click on Manage access. 3. Click Invite a collaborator and enter the contributor's username or email. 4. Once accepted, assign the Admin role to the contributor. [GitHub Docs](https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles?utm_source=chatgpt.com) | Administrators can push changes directly to the repository or follow the pull request process. It's recommended to use pull requests to ensure code quality and maintain a clear history. GitHub Docs |
| Private |  | Outside Collaborator (Read) | Not Applicable | Grant the contributor Read access as an outside collaborator: 1. Navigate to the repository's Settings. 2. Click on Manage access. 3. Click Invite a collaborator and enter the contributor's username or email. 4. Once accepted, assign the Read role to the contributor. [GitHub Docs](https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles?utm_source=chatgpt.com) | Read access alone does not permit creating branches or submitting pull requests. Contributors need at least write access to participate. GitHub Docs |
| Private |  | Outside Collaborator (Write) | If granted access, create Branch & Submit Pull Request | Grant the contributor Write access as an outside collaborator: 1. Navigate to the repository's Settings. 2. Click on Manage access. 3. Click Invite a collaborator and enter the contributor's username or email. 4. Once accepted, assign the Write role to the contributor. [GitHub Docs](https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles?utm_source=chatgpt.com) | Contributors can create branches and submit pull requests within the repository. GitHub Docs |
| Private |  | Organization Member (Read) | Not Applicable | Add the contributor as an Organization Member with Read access: 1. Navigate to the organization's People tab. 2. Click Invite member and enter the contributor's username or email. 3. Assign the Read role to the contributor. [GitHub Docs](https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles?utm_source=chatgpt.com) | Read access alone does not permit creating branches or submitting pull requests. Members need at least write access to participate. GitHub Docs |
| Private |  | Organization Member (Write) | If granted access, create Branch & Submit Pull Request | Add the contributor as an Organization Member with Write access: 1. Navigate to the organization's People tab. 2. Click Invite member and enter the contributor's username or email. 3. Assign the Write role to the contributor. [GitHub Docs](https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles?utm_source=chatgpt.com) | Members can create branches and submit pull requests within the repository. GitHub Docs |
| Private |  | Organization Member (Admin) | If granted access, direct Push or Create Branch & Submit Pull Request | Add the contributor as an Organization Member with Admin access: 1. Navigate to the organization's People tab. 2. Click Invite member and enter the contributor's username or email. 3. Assign the Admin role to the contributor. [GitHub Docs](https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles?utm_source=chatgpt.com) | Administrators can push changes directly to the repository or follow the pull request process. It's recommended to use pull requests to ensure code quality and maintain a clear history. GitHub Docs |

# **Members and outside collaborators**

Organization Members:

* Broader Access: Members can be part of multiple teams, granting them access to various repositories within the organization.
* Team Inclusion: They can be assigned to teams, facilitating streamlined access management across multiple repositories.
* Additional Permissions: Depending on their role, members may have permissions beyond repository access, such as creating repositories or managing teams.
* Visibility: Members can view other members and teams within the organization, enhancing collaboration.

Outside Collaborators:

* Limited Access: Collaborators have access only to specific repositories they've been invited to, without broader organizational permissions.
* No Team Inclusion: They cannot be added to teams; access must be managed individually for each repository.
* Restricted Permissions: Their permissions are confined to the repositories they have access to, without additional organizational capabilities.
* Limited Visibility: Collaborators do not have visibility into the organization's member list or team structures.

In summary, while both organization members and outside collaborators can have similar access levels to repositories, organization members benefit from broader access, team inclusion, and additional organizational permissions, whereas outside collaborators have more restricted, repository-specific access.

# **About Teams**

In GitHub, Teams are a feature within organizations that facilitate efficient collaboration and streamlined access management among members. Here's an overview:

Purpose of Teams:

* Organizational Structure: Teams represent groups within an organization, such as departments or project-based squads, allowing for a clear organizational hierarchy.
* Access Management: By assigning repositories to teams, you can manage permissions collectively, simplifying the process of granting or modifying access.

Key Features:

* Hierarchical Organization: Teams can have parent-child relationships, enabling the creation of subteams that inherit permissions from parent teams.
* Unified Communication: Teams can be integrated with communication platforms like Microsoft Teams, allowing members to receive updates and collaborate seamlessly.
* Granular Permissions: Assign specific roles (read, write, admin) to teams for each repository, ensuring members have appropriate access levels.

Benefits:

* Simplified Access Control: Managing permissions at the team level reduces administrative overhead compared to individual assignments.
* Enhanced Collaboration: Teams facilitate better coordination among members working on shared projects or within the same department.
* Scalability: As organizations grow, teams provide a scalable method to manage access and collaboration across multiple projects and repositories.

Integration with Communication Tools:

GitHub's integration with platforms like Microsoft Teams enables:

* Real-Time Notifications: Receive updates on repository activities directly within your communication channels.
* Actionable Messages: Perform GitHub actions, such as commenting on issues or merging pull requests, from within the communication platform.
* Contextual Discussions: Share GitHub links that expand to show relevant details, facilitating informed discussions.

For more information on integrating GitHub with Microsoft Teams, refer to the [GitHub + Microsoft Teams Integration](https://github.com/integrations/microsoft-teams).

By leveraging GitHub Teams, organizations can effectively manage member access, foster collaboration, and maintain organized workflows across their projects.

# Permissions per repo

To manage your GitHub organization's repositories effectively as you transition from solo development to collaborative work, consider the following steps:

1. Understanding Your Current Setup

Currently, you're able to push changes directly to the main branch of your repositories. This capability is standard in Git when no branch protection rules are in place. Since you're using Ubuntu and not GitHub Desktop, your workflow likely involves using Git via the command line.

2. Restricting Direct Pushes to the Main Branch

To ensure that only you can push directly to the main branch while requiring others to use pull requests, you can set up branch protection rules:

- Navigate to Branch Protection Settings:

1. Go to your repository on GitHub.
2. Click on the "Settings" tab.
3. In the left sidebar, select "Branches."
4. Under "Branch protection rules," click "Add rule."

- Configure the Protection Rule:

1. In "Branch name pattern," enter main (or the name of your default branch).
2. Enable "Require a pull request before merging" to mandate pull requests for changes.
3. To specify who can bypass these rules:
   * Enable "Allow specified actors to bypass required pull requests."
   * Search for and add your GitHub username.

This setup ensures that only you can push directly to the main branch, while others must use pull requests.

3. Maintaining Your Direct Push Access

With the above configuration, your direct push access remains intact. However, ensure that:

- Your username is correctly added to the list of actors allowed to bypass pull request requirements.

* You have the necessary permissions (admin or write access) for the repository.

By implementing these branch protection rules, you can maintain your workflow while facilitating collaborative development through pull requests for other contributors.

## Necessary permissions

In GitHub, permissions determine the actions a user can perform within a repository or organization. These permissions are structured into roles, each encompassing a set of capabilities.

Repository Roles in an Organization

GitHub provides several predefined roles for repositories within an organization, each with specific permissions:

* Read: Allows viewing and forking repositories.
* Triage: Enables managing issues and pull requests without write access.
* Write: Permits pushing changes to branches.
* Maintain: Grants permissions to manage repository settings and users without full administrative access.
* Admin: Provides full control over the repository, including sensitive actions like deleting the repository.

For a detailed breakdown of permissions associated with each role, refer to GitHub's documentation on [repository roles for an organization](https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories/managing-repository-roles/repository-roles-for-an-organization).

Ensuring Appropriate Permissions

To ensure that only you can push directly to the main branch:

1. Assign Roles Appropriately: Grant other contributors roles like Triage or Read, which do not include push permissions.
2. Implement Branch Protection Rules: Configure branch protection to restrict who can push to the main branch, allowing exceptions for specific users as needed.

By carefully managing roles and branch protections, you can control access and maintain the integrity of your project's main branch.

# Roles per organization

​To view the owners of your GitHub organization, follow these steps:

1. Access Your Organization's People Page:

* In the upper-right corner of GitHub, click your profile photo, then select "Your organizations."
* Click the name of your organization.
* Under your organization’s name, click "People."

2. Filter by Role:

* On the "People" page, locate the "Role" dropdown menu.
* Select "Owner" from the dropdown to filter the list and display only organization owners.

This will provide you with a list of all members who have owner privileges in your organization.

Note: Only organization members can view people's roles within the organization.

If you are not a member of the organization and need to contact its owners, GitHub does not provide a direct method to view or contact organization owners. In such cases, you might consider reaching out to GitHub Support for assistance.