 Getting Started with GitHub Enterprise

# Configuring GitHub for Collaboration and Compliance

Presented by @elstudio



# So you've bought GitHub Enterprise

It's so shiny!



GitHub Enterprise

# Loved by developers. Trusted by enterprises.

Innovate faster on the platform your team knows and loves—with the security your business demands.

[Start a free trial](#)[Contact Sales →](#)

## Contact Sales for pricing →

- ✓ SAML single sign-on
- ✓ Access provisioning
- ✓ Invoice billing
- ✓ Self-hosted or cloud-hosted
- ✓ FedRAMP and SOC 2 compliance
- ✓ Dependency insights
- ✓ Priority support
- ✓ Access to Actions and Packages
- ✓ Simplified account administration
- ✓ 99.95% uptime SLA for Enterprise Cloud





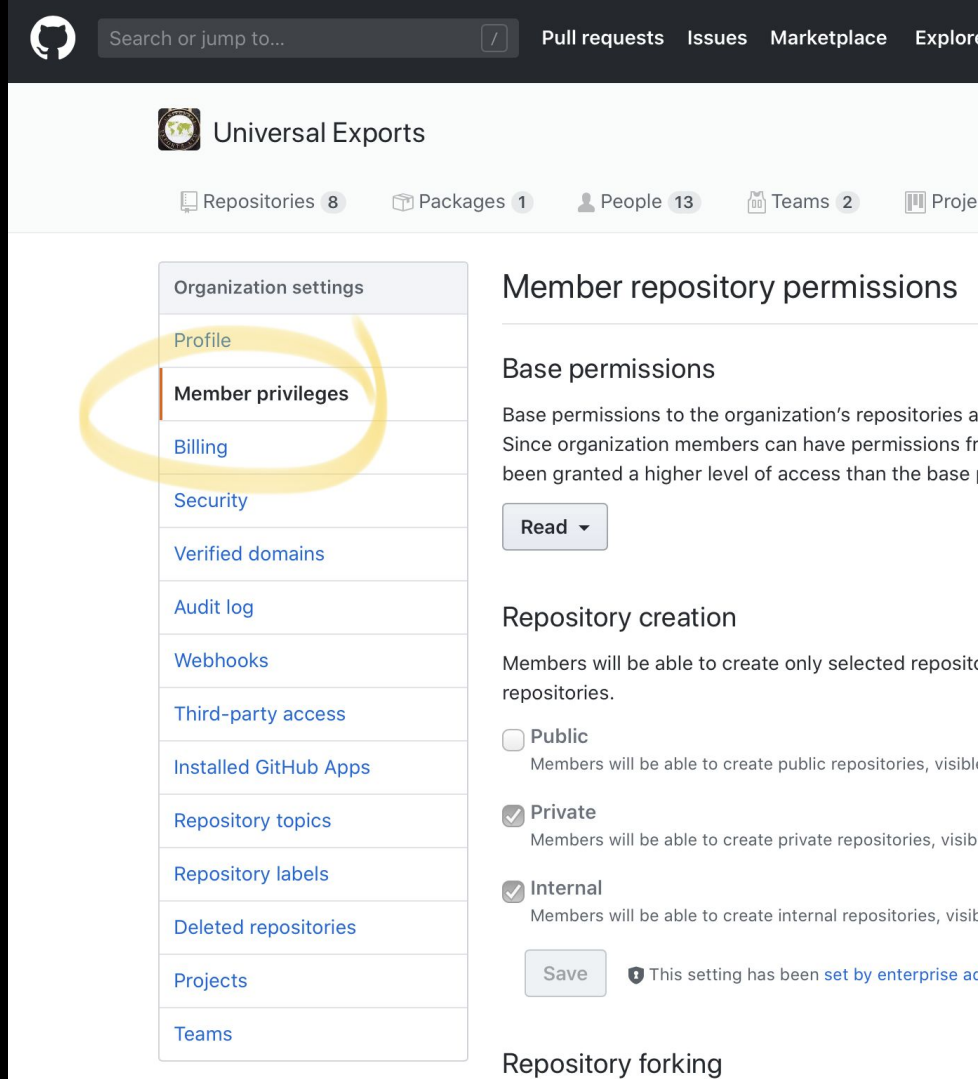
**Now what?**

# Configure GitHub for sharing

# Organization Settings

Set policy for repository creation and visibility:

- Who can create repositories
- How visible can these repositories be?
- Are users allowed to create teams?
- Invite external collaborators?



Search or jump to... Pull requests Issues Marketplace Explore

Universal Exports

Repositories 8 Packages 1 People 13 Teams 2 Projects

Organization settings

- Profile
- Member privileges**
- Billing
- Security
- Verified domains
- Audit log
- Webhooks
- Third-party access
- Installed GitHub Apps
- Repository topics
- Repository labels
- Deleted repositories
- Projects
- Teams

### Member repository permissions

#### Base permissions

Base permissions to the organization's repositories and packages. Since organization members can have permissions from other organizations, they may have been granted a higher level of access than the base permissions.

Read ▾

#### Repository creation

Members will be able to create only selected repository types.

- ☐ Public  
Members will be able to create public repositories, visible to anyone on the internet.
- ☒ Private  
Members will be able to create private repositories, visible only to organization members.
- ☒ Internal  
Members will be able to create internal repositories, visible only to organization members.


Save ⓘ This setting has been set by enterprise administrator


#### Repository forking


# Organization Settings or Enterprise Policy?


Organization owners can set policies independently for each organization


Enterprise owners can override any of these — or delegate these choices to individual orgs

 Search or jump to... Pull requests Issues Marketplace Explore

 Octodemo

 Organizations

 People


 Policies

Repositories

[Projects](#)

[Teams](#)

[Organizations](#)


 Settings

## Repository policies

### Default permissions


Default permissions to the organizations' repositories apply to all organization members and collaborators. Since organization members can have permission to create repositories, and collaborators who have been granted a higher level of permission can create repositories, retain their higher permission privileges.

All organizations: No policy ▾

 [View your organizations' current configurations](#) without the ability to change them.

### Repository creation

If enabled, members will be able to create repositories. Organization members can choose whether to allow members to create repositories.

 [View your organizations' current configurations](#) without the ability to change them.

☐ No Policy

Organization members can choose whether to allow members to create repositories.

☐ Disabled

Members will not be able to create repositories.

☒ Members can create repositories

Members will only be allowed to create the specified types of repositories.

☐ Public

☒ Private

☒ Internal

Save

**Let's set some policies!**



# Organization Settings > Member privileges

Set policy for repository creation and visibility:

- Who can create repositories
- How visible can these repositories be?
- Are users allowed to create teams?
- Invite external collaborators

Search or jump to...

Pull requests Issues Marketplace Explore

Universal Exports

Repositories 8 Packages 1 People 13 Teams 2 Projects

Organization settings

- Profile
- Member privileges**
- Billing
- Security
- Verified domains
- Audit log
- Webhooks
- Third-party access
- Installed GitHub Apps
- Repository topics
- Repository labels
- Deleted repositories
- Projects
- Teams

## Member repository permissions

### Base permissions

Base permissions to the organization's repositories and packages. Since organization members can have permissions from other organizations, they may have been granted a higher level of access than the base permissions.

Read ▾

### Repository creation

Members will be able to create only selected repository types in this organization.

- ☐ Public  
Members will be able to create public repositories, visible to anyone on the internet.
- ☒ Private  
Members will be able to create private repositories, visible only to organization members.
- ☒ Internal  
Members will be able to create internal repositories, visible only to organization members.

Save ⓘ This setting has been set by enterprise administrator

### Repository forking

# Base permissions

## Default permissions for repositories

- Apply to organization members
- Overrides permissions on individual repositories

## What's best for your org?

- **Read** is best for innersource
  - Allows read of private repos
- **None** for private by default

ges 1

People 13

Teams 2

Projects

Insights

Settings

## Member repository permissions

### Base permissions

Base permissions to the organization's repositories apply to all members and excludes outside collaborators. Since organization members can have permissions from multiple sources, members and collaborators who have been granted a higher level of access than the base permissions will retain their higher permission privileges.

1

Read

#### Organization member permissions

##### None

Members will only be able to clone and pull public repositories. To give a member additional access, you'll need to add them to teams or make them collaborators on individual repositories.

##### ✓ Read

Members will be able to clone and pull all repositories.

##### Write

Members will be able to clone, pull, and push all repositories.

##### Admin

Members will be able to clone, pull, push, and add new collaborators to all repositories.

If enabled, forking is allowed on private, internal, and public repositories. If disabled, forking is only allowed on public repositories. This setting is also configurable per-repository.

Save

# Other member repository permissions

1. What kinds of repositories can members create?
  - Uncheck Public if you require a review process
2. Forks are copies of repositories, for experimentation and bug fixes
3. Can repository admins can invite folks outside of the organization to collaborate?

1

## Repository creation

Members will be able to create only selected repository types. Outside collaborators can create public repositories.

☐ Public

Members will be able to create public repositories, visible to anyone.

☒ Private

Members will be able to create private repositories, visible to organization members with permissions.

☒ Internal

Members will be able to create internal repositories, visible to all [enterprise members](#).

Save

🔒 This setting has been [set by enterprise administrators](#).

2

## Repository forking

☐ Allow forking of private and internal repositories

If enabled, forking is allowed on private, internal, and public repositories. If disabled, forking is only allowed on public repositories. This setting is also configurable per-repository.

Save

3

## Repository invitations

☐ Allow members to invite outside collaborators to repositories for this organization

If disabled, only organization owners may invite collaborators to repositories.

Save

# Actions permissions

Actions are 3rd-party code for CI/CD or other automation

- Should Actions run at all?
- What sort of Actions should be allowed?
  - Local actions exist directly in the repository
  - Specific hand-picked actions from other sources

## Actions permissions

### Policies

Choose which repositories are permitted to use GitHub Actions.

All repositories ▾

☐ **Allow all actions**

Any action can be used, regardless of who authored it or where it is defined.

☐ **Allow local actions only**

Only actions defined in a repository within the enterprise can be used.

☒ **Allow select actions**

Only actions that match specified criteria can be used. [Learn more about allowing specific actions](#)

☐ Allow actions created by GitHub

☐ Allow Marketplace actions by [verified creators](#)

#### Allow specified actions

Enter a comma-separated list of actions

Wildcards, tags, and SHAs are allowed. Examples: monalisa/octocat@\*, monalisa/

# Admin repository permissions

These govern what repository administrators can do:

1. Can they change a repository from public to private?
2. Delete repositories or transfer them outside of the org?
3. Delete issues?
4. Show authors' full name?

## Admin repository permissions

1

### Repository visibility change

☒ **Allow members to change repository visibilities for this organization**

If enabled, members with admin permissions for the repository will be able to change repository visibility from **public** to **private**. If disabled, only organization owners can change repository visibilities.

Save

2

### Repository deletion and transfer

☐ **Allow members to delete or transfer repositories for this organization**

If enabled, members with admin permissions for the repository will be able to delete or transfer **public** and **private** repositories. If disabled, only organization owners can delete or transfer repositories.

Save

ⓘ This setting has been **disabled by enterprise administrators**.

3

### Issue deletion Beta

☐ **Allow members to delete issues for this organization**

If enabled, members with admin permissions for the repository will be able to delete issues.

Save

4

### Repository Comments

☐ **Allow members to see comment author's profile name in private repositories**

If enabled, members will be able to see comment author's profile name in issues and pull requests for private repositories.

Save

# Member team permissions

1. Can organization members create teams?
2. Can members view organization-wide insights?
  - Security notifications and licensing for open-source libraries in use

1

## Member team permissions

### Team creation rules

☒ **Allow members to create teams**

If enabled, any member of the organization will be able to create new teams. If disabled, only organization owners can create new teams.

Save

2

## Member organization permissions

### Dependency insights

☒ **Allow members to view dependency insights**

If disabled, only organization owners may view dependency insights, including aggregated information about security advisories and licenses in dependencies.

Save

 This setting has been [enabled by enterprise administrators](#).

# Security settings

# Organization Settings > Security

Settings for access to GitHub  
Enterprise

- What's required for access?
- Settings for Two-factor Authentication, Single Sign-on, etc.

The screenshot shows the GitHub 'Universal Exports' interface. At the top, there are navigation links for Repositories (8), Packages (1), People (13), Teams (2), Projects, and Insights. The left sidebar contains a list of settings categories: Organization settings (selected), Profile, Member privileges, Billing, Security (highlighted with a yellow oval), Verified domains, Audit log, Webhooks, Third-party access, Installed GitHub Apps, Repository topics, Repository labels, Deleted repositories, Projects, and Teams. Below these are Developer settings and OAuth Apps. The main content area is titled 'Two-factor authentication' and includes a sub-header 'Requiring an additional authentication method adds another level of security to your organization.' A toggle switch for 'Require two-factor authentication for everyone in the Universal Exports organization' is currently turned off. A yellow warning box states: 'Members, billing managers, and outside collaborators who do not have two-factor authentication enabled for their personal account will be removed from the organization about the change. [Learn more.](#)' A 'Save' button is located below the warning box. The next section is 'SAML single sign-on', with a sub-header 'Manage your organization's membership while adding another level of security to your organization.' A toggle switch for 'Enable SAML authentication' is also turned off. A text box explains: 'Enable SAML authentication for your organization through an identity provider or your custom SAML 2.0 provider.' Another 'Save' button is present. The final section visible is 'Team synchronization', with a sub-header 'Team synchronization lets you manage team membership through your organization's SSO provider.'

Universal Exports

Repositories 8 Packages 1 People 13 Teams 2 Projects Insights

Organization settings

Profile

Member privileges

Billing

**Security**

Verified domains

Audit log

Webhooks

Third-party access

Installed GitHub Apps

Repository topics

Repository labels

Deleted repositories

Projects

Teams

Developer settings

OAuth Apps

## Two-factor authentication

Requiring an additional authentication method adds another level of security to your organization.

☐ Require two-factor authentication for everyone in the Universal Exports organization

Members, billing managers, and outside collaborators who do not have two-factor authentication enabled for their personal account will be removed from the organization about the change. [Learn more.](#)

Save

## SAML single sign-on

Manage your organization's membership while adding another level of security to your organization.

☐ Enable SAML authentication

Enable SAML authentication for your organization through an identity provider or your custom SAML 2.0 provider.

Save

## Team synchronization

Team synchronization lets you manage team membership through your organization's SSO provider.



# Sign-on settings

## Settings for access to GitHub Enterprise

1. Require 2FA for GitHub login?
  - Either SMS or TOTP code or U2F hardware key
  - For **outside collaborators** like contractors & members
  - SSO can have its own 2FA, for org members

1

### Two-factor authentication

Requiring an additional authentication method adds another level of security for your organization.

- ☐ **Require two-factor authentication for everyone in the Universal Exports organization.**

Members, billing managers, and outside collaborators who do not have two-factor authentication enabled for their personal account will be removed from the organization and will receive an email notifying them about the change. [Learn more.](#)

Save

### SAML single sign-on

Manage your organization's membership while adding another level of security with SAML. [Learn more](#)

- ☐ **Enable SAML authentication**

Enable SAML authentication for your organization through an identity provider like Azure, Okta, Onelogin, Ping, or your custom SAML 2.0 provider.

Save

# SAML SSO settings

GitHub Enterprise Cloud supports [SAML 2.0 for Single Sign-on](#)

1. Your SAML IdP will provide values for these settings
2. Update the signature & digest method to match
3. Test before saving
4. And test with actual users
  - You can require SSO for members — not outside collaborators

## SAML single sign-on

Manage your organization's membership while adding another level of security with SAML. [Learn more](#)

### ☒ Enable SAML authentication

Enable SAML authentication for your organization through an identity provider like Azure, Okta, Onelogin, Ping Identity or your custom SAML 2.0 provider.

1

#### Sign on URL

Members will be forwarded here when signing in to your organization

#### Issuer

Typically a unique URL generated by your SAML identity provider

#### Public certificate

Your SAML provider is using the RSA-SHA256 Signature Method and the SHA256 Digest Method. [✎](#)

2

The assertion consumer service URL is <https://github.com/orgs/universal-exports-ltd/saml/consume>.

3

#### Test SAML configuration

Before enabling test your SAML SSO configuration

4

- ☐ Require SAML SSO authentication for all members of the Universal Exports organization.  
Requiring SAML SSO will remove all members (excluding outside collaborators) who have not authenticated their accounts. Members will receive an email notifying them about the change. Leaving this option unchecked will allow you to test before requiring.

ⓘ You must [single sign-on](#) before you can require SAML SSO for all members of the Universal Exports organization.

**Who gets superpowers?**  
**Organization owners**



## Universal Exports



Repositories 8



Packages 1

**People 13**

Teams 2



Projects



Insights



Settings



role:owner

**Members**

Outside collaborators

**Invite member**

Team members

283 licenses left — [Buy more](#) ?

2FA ▾

Role ▾

**Eric Johnson**  
elstudio

No verified domain email

2FA ✓



Filter by role

Everyone

✓ **Owners**

Members

# What can these folks do?

Pretty much everything!

- View and set repository settings
- Invite members to the organization
- Set permissions for the org
  - Unless these are overridden at the Enterprise
- [Permission levels for an organization](#)

Limit org admin to 3-5 folks

## Permission levels for an organization

Organization members can have *owner*, *billing manager*, or *member* roles:

- **Owners** have complete administrative access to your organization. This role should be limited to a few people in your organization. For more information, see "[Changing a person's role to owner](#)."
- **Billing managers** allow a person to manage billing settings. For more information, see "[Adding a billing manager to your organization](#)".
- **Members** are the default role for everyone else.

Organization action	Owners	Members	Billing managers
Create repositories (see " <a href="#">Restricting repository creation in your organization</a> " for details)	X	X	
View and edit billing information	X		X
Invite people to join the organization	X		
Edit and cancel invitations to join the organization	X		
Remove members from the organization	X		
Reinstate former members to the organization	X		

# Repository permissions

🔒 universal-exports-ltd / **reading-time-demo** Internal

👁 Watch ▾

5

★ Star

0

🍴 Fork

1

<> Code

🔔 Issues 1

🔗 Pull requests 2

🎬 Actions

📁 Projects 1

📖 Wiki

🛡 Security

📊 Insights

⚙ Settings

Options

Manage access

Branches

Webhooks

Notifications

Integrations & services

Deploy keys

Autolink references

Secrets

Actions

## Who has access

Beta

[Learn more](#) or [give us feedback](#)

### INTERNAL REPOSITORY



Members of any organization belonging to Octodemo can see this repository.

[Manage](#)

### BASE ROLE

Read

All 13 Octodemo members can access this repository.

[Manage](#)

### DIRECT ACCESS



6 have access to this repository. [2 members](#). [2 outside collaborators](#). [2 teams](#).

## Manage access

[Create team](#)

[Invite teams or people](#)

☐ Select all

Type ▾ Role ▾

# Repository access

**Who has access** is an overview of access (from Internal visibility, etc)  
Repository admins can invite others or change permissions:

1. Teams
  - Non-members invited to individual repositories
2. Individual members
  - Perfect for contractors or others who can't SSO
3. Outside collaborators

The screenshot shows the GitHub interface for the 'ng-time-demo' repository. At the top, there are tabs for 'Watch', 'Star', and 'Fork'. Below these are navigation links for 'equests 2', 'Actions', 'Projects 1', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Who has access' section is highlighted, showing three categories: 'INTERNAL REPOSITORY' (Members of any organization belonging to Octodemo can see this repository. [Manage](#)), 'BASE ROLE' (All 13 Octodemo members can access this repository. [Manage](#)), and 'DIRECT ACCESS' (6 have access to this repository. 2 members. 2 outside collaborators. 2 teams. [Learn more or give us feedback](#)). Below this is the 'Manage access' section, which includes a search bar and a list of access entries. The list is numbered 1, 2, and 3, corresponding to the list in the text. Entry 1 is '00-team' (6 members, Role: Write). Entry 2 is 'Azure Specialists' (7 members, Role: Write). Entry 3 is 'Froilan Irizarry Rivera' (Role: Write). Entry 4 is 'Isaac Cohen' (Outside Collaborator, Role: Read).

ng-time-demo Internal

Watch 5 Star 0 Fork 1

equests 2 Actions Projects 1 Wiki Security Insights Settings

### Who has access

[Beta](#) [Learn more or give us feedback](#)

INTERNAL REPOSITORY

Members of any organization belonging to Octodemo can see this repository.

[Manage](#)

BASE ROLE

All 13 Octodemo members can access this repository.

[Manage](#)

DIRECT ACCESS

6 have access to this repository. 2 members. 2 outside collaborators. 2 teams.

### Manage access

[Create team](#) [Invite teams or people](#)

Select all Type Role

Find a team, organization member or outside collaborator...

1

00-team

@00-team • 6 members

Role: Write

2

Azure Specialists

@azure-specialists • 7 members

Role: Write

3

Froilan Irizarry Rivera

froi

Role: Write

Isaac Cohen

issc29 • Outside Collaborator

Role: Read



# Branch protections

Enforce workflow and branching strategy:

1. Branch name (\* is a wildcard)
2. Require code reviews?
3. Require status checks (from automated testing or tools)?

Create [rules for branches](#) as needed

The screenshot shows the GitHub interface for configuring branch protection rules. The repository is 'universal-exports-ltd / reading-time-demo'. The left sidebar contains a menu with 'Options', 'Manage access', 'Branches' (highlighted with a yellow box and a '1'), 'Webhooks', 'Notifications', 'Integrations & services', 'Deploy keys', 'Autolink references', 'Secrets', 'Actions' (highlighted with a yellow box and a '2'), 'Moderation', and 'Reported content'. The main content area is titled 'Branch protection rule' and shows the configuration for the 'master' branch. The 'Branch name pattern' is set to 'master'. Under 'Protect matching branches', three rules are enabled: 'Require pull request reviews before merging' (with a dropdown for 'Required approving reviews: 1'), 'Require status checks to pass before merging', and 'Require branches to be up to date before merging'. A yellow box with a '3' highlights the 'Require status checks to pass before merging' section.

universal-exports-ltd / reading-time-demo Internal

Watch 5 Star

<> Code Issues 1 Pull requests 2 Actions Projects 1 Wiki Security Insights

Options

Manage access

Branches 1

Webhooks

Notifications

Integrations & services

Deploy keys

Autolink references

Secrets

Actions 2

Moderation

Reported content

### Branch protection rule

Branch name pattern

master

Applies to 1 branch

master

#### Protect matching branches

- ☒ **Require pull request reviews before merging**  
When enabled, all commits must be made to a non-protected branch and submitted via a pull request. When enabled, all commits must be made to a non-protected branch and submitted via a pull request. When enabled, all commits must be made to a non-protected branch and submitted via a pull request.  
Required approving reviews: 1
- ☐ **Dismiss stale pull request approvals when new commits are pushed**  
New reviewable commits pushed to a matching branch will dismiss pull request review approvals.
- ☐ **Require review from Code Owners**  
Require an approved review in pull requests including files with a designated code owner.
- ☐ **Restrict who can dismiss pull request reviews**  
Specify people or teams allowed to dismiss pull request reviews.
- ☒ **Require status checks to pass before merging**  
Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to the branch that matches this rule after status checks have passed.
- ☒ **Require branches to be up to date before merging**  
This ensures pull requests targeting a matching branch have been tested with the latest code. This ensures pull requests targeting a matching branch have been tested with the latest code. This ensures pull requests targeting a matching branch have been tested with the latest code.  
Status checks found in the last week for this repository

