



# End User Administration

# End User Administration

## **Section Goal:**

Administration of GitHub Enterprise.



## **Topics and Agenda:**

- **Create an Organization**
- **Create a Team**
- **Create a Repository**
- **Secure a Repository**



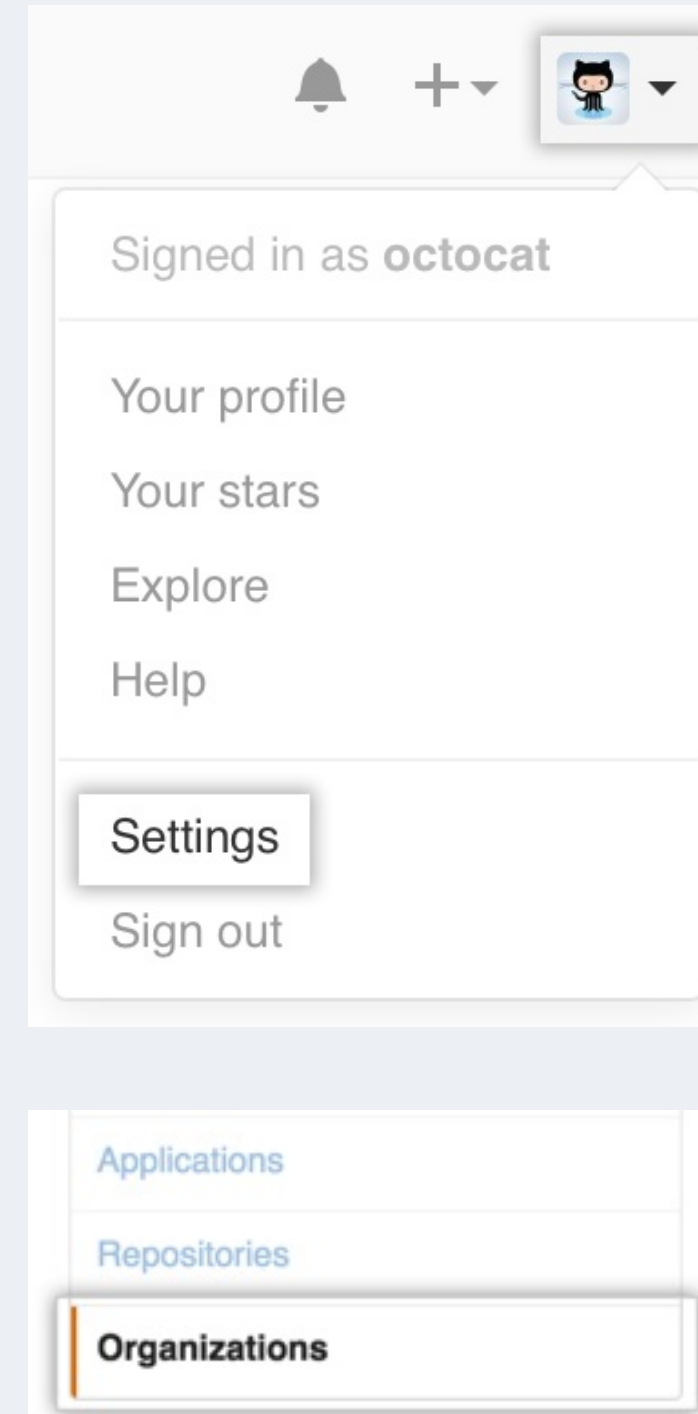
## **Topics and Agenda:**

- **Notifications and @ Mentions**
- **Git Large File Storage**
- **Benefits of git-lfs**
- **Enable git-lfs**

# Create an Organization

An [organization](#) is a collection of user accounts that owns repositories. To create an organization:

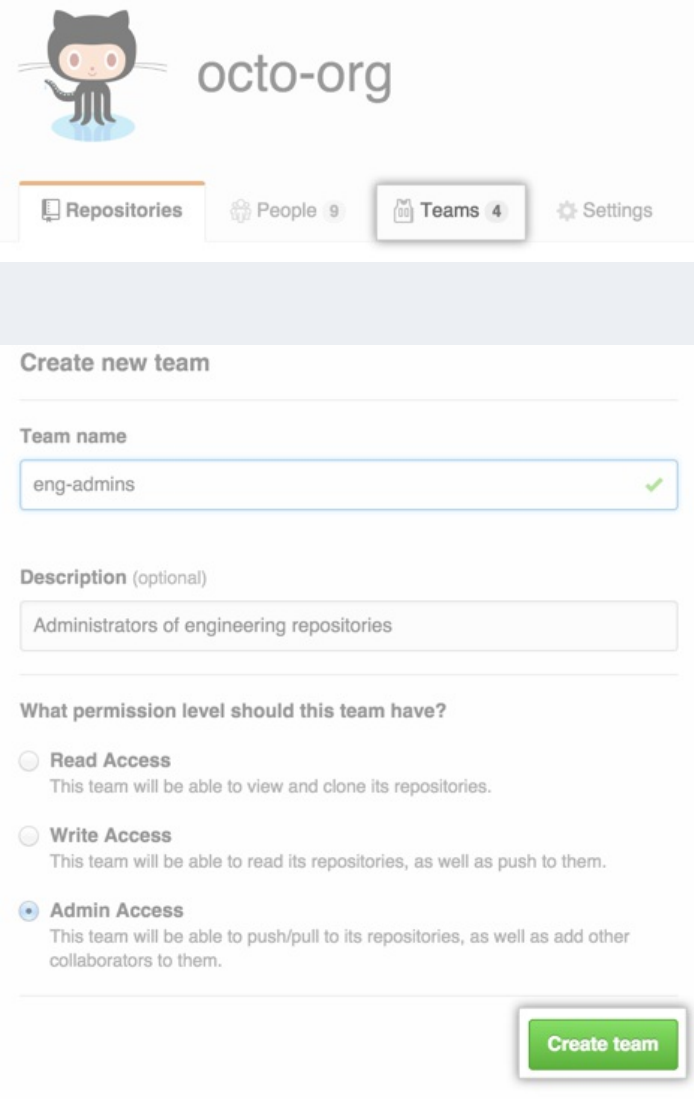
- **click your profile photo > 'Settings'**
- **settings sidebar, click 'Organizations'**
- **Organizations, click 'New organization'**
- **give the organization a name**
- **Enter the contact email**
- **Click 'Create organization'**



# Create a Team

**Teams** give organizations the ability to create groups of members and control access to repositories. Team members can be granted read, write, or admin permissions to specific repositories. To create a team:

- **click your profile photo > 'Your profile'**
- **Organizations, click your org icon**
- **click 'Teams' under your org**
- **click 'New team'**
- **decide if it will be 'Visible'**
- **click 'Create team'**



The screenshot shows the GitHub interface for the 'octo-org' organization. At the top, there's a navigation bar with 'Repositories', 'People 9', 'Teams 4' (highlighted), and 'Settings'. Below this is the 'Create new team' form. The 'Team name' field contains 'eng-admins' with a green checkmark. The 'Description (optional)' field contains 'Administrators of engineering repositories'. Under 'What permission level should this team have?', the 'Admin Access' option is selected with a radio button. The other options are 'Read Access' and 'Write Access'. At the bottom right of the form is a green 'Create team' button.

octo-org

Repositories People 9 Teams 4 Settings

Create new team

Team name

eng-admins ✓

Description (optional)

Administrators of engineering repositories

What permission level should this team have?

☐ Read Access  
This team will be able to view and clone its repositories.

☐ Write Access  
This team will be able to read its repositories, as well as push to them.

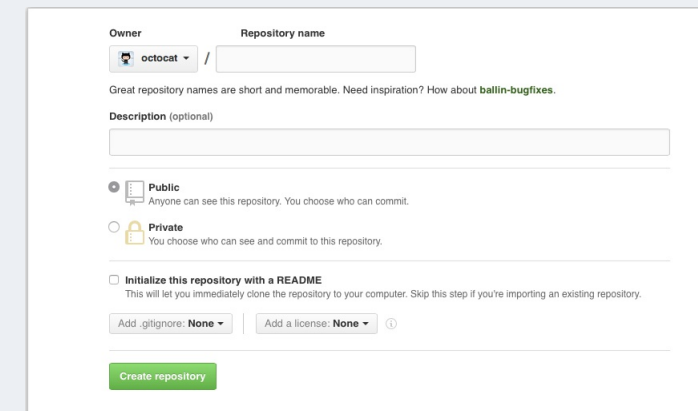
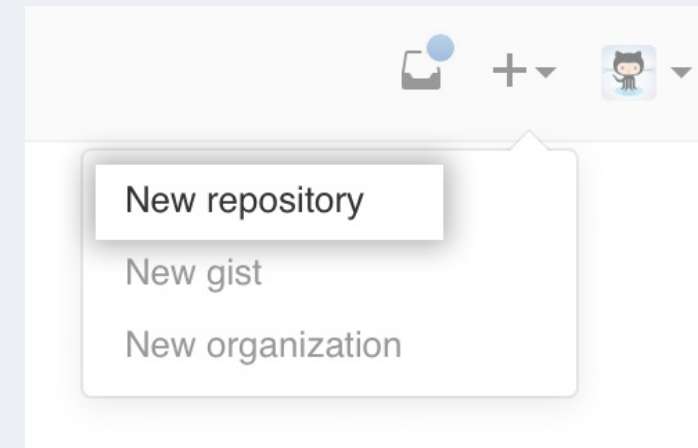
☒ Admin Access  
This team will be able to push/pull to its repositories, as well as add other collaborators to them.

Create team

# Create a Repository

You can [create a new repository](#) on your personal account or any organization where you have sufficient permissions. To create a repository:

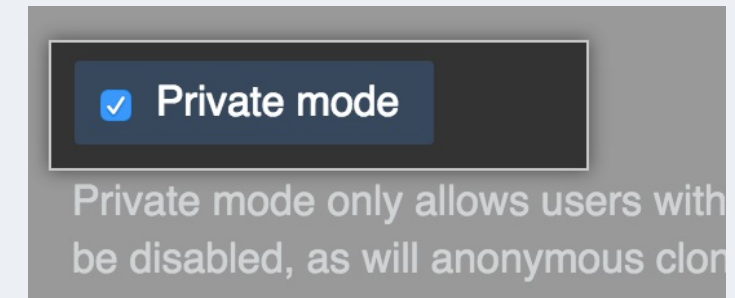
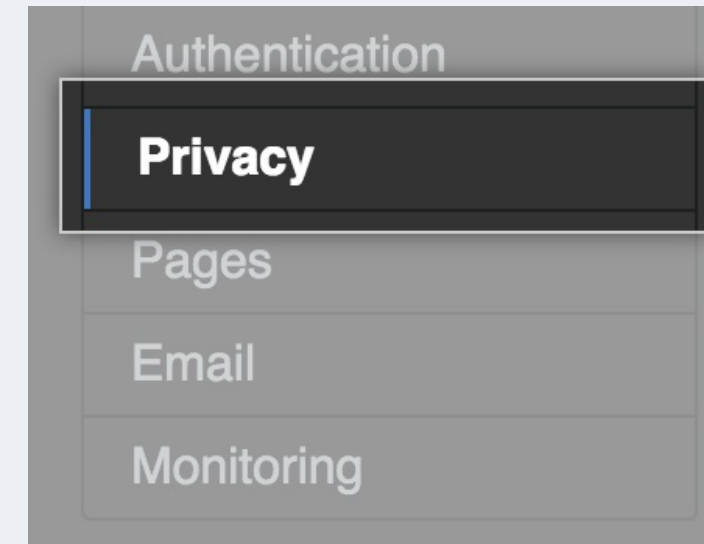
- **click + > 'New repository'**
- **select the account**
- **type a name**
- **choose Public or Private**
- **click 'Create repository'**

A screenshot of the GitHub 'Create repository' form. The form is white with a light gray border. It has two main sections: 'Owner' and 'Repository name'. The 'Owner' section shows a dropdown menu with 'octocat' selected. The 'Repository name' section has a text input field. Below these fields, there is a note: 'Great repository names are short and memorable. Need inspiration? How about ballin-bugfixes.' The 'Description (optional)' section has a text input field. The 'Public or Private' section has two radio buttons: 'Public' (selected) and 'Private'. The 'Public' option has a sub-note: 'Anyone can see this repository. You choose who can commit.' The 'Private' option has a sub-note: 'You choose who can see and commit to this repository.' The 'Initialize this repository with a README' section has a checkbox that is unchecked. Below this, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. At the bottom of the form is a green button labeled 'Create repository'.

# Private Mode

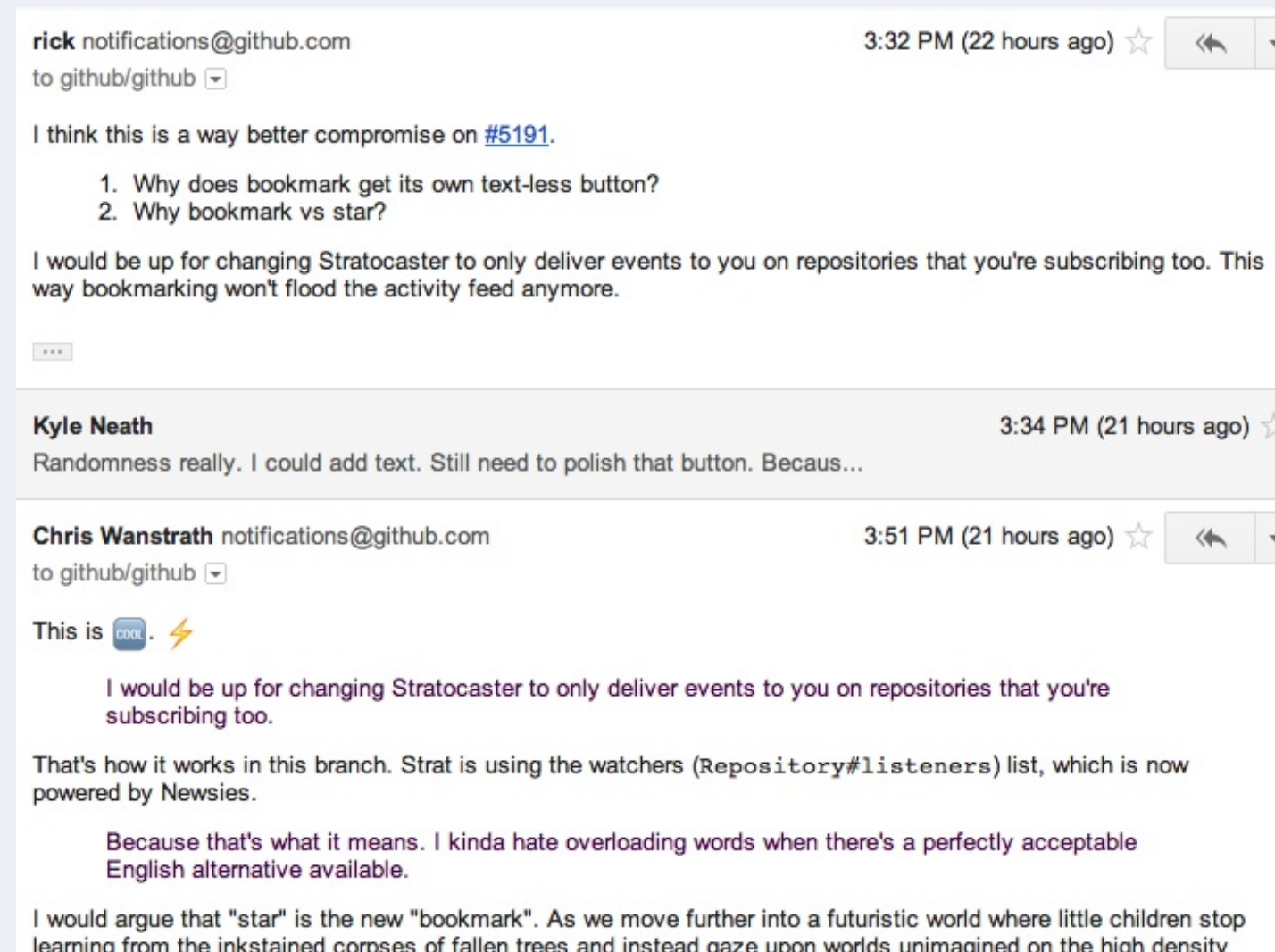
In [private mode](#), GitHub Enterprise requires every user to sign in to access the installation. To enable private mode:

- [Access the Management Console](#)
- click 'Settings'
- click 'Privacy'
- Select 'Private mode'



# Notification Emails

You can choose to receive [notification emails](#) about changes made to any repository, issue, or pull request that you have access to. GitHub sends multipart emails, which contain both HTML and plain text copies of the email content. We'll automatically format Markdown, @mentions, emojis, hash-links, and more:



# Git Large File Storage

**An open source Git extension for versioning large files**

**Git Large File Storage (LFS)** replaces large files such as audio samples, videos, datasets, and graphics with text pointers inside Git, while storing the file contents on a remote server like GitHub.com or GitHub Enterprise.





# Git LFS Features



Large file versioning:

- **Version large files—even those as large as a couple GB in size—with Git.**



More repository space:

- **Host more in your Git repositories. External file storage makes it easy to keep your repository at a manageable size.**



Faster cloning and fetching:

- **Download less data. This means faster cloning and fetching from repositories that deal with large files.**



Same Git workflow:

- **Work like you always do on Git—no need for additional commands, secondary storage systems, or toolsets.**




Same access controls and permissions:

- **Keep the same access controls and permissions for large files as the rest of your Git repository when working with a remote host like GitHub.**

# Configure Git Large File Storage

You can enable or disable [Git LFS](#) on a per-repository basis:

- As a site admin, sign in to your GitHub Enterprise instance at `http(s)://[hostname]/login`
- In the upper-right corner of any page, click 
- Search for the name of the repository
- In the left sidebar, click 'Advanced Settings'
- In the Key field, type `git-lfs`
- In the Value field, type `true` or `false`

