Table of Contents

[GitHub administration for enterprise support and adoption 2](#_Toc132916104)

[Learning objectives 2](#_Toc132916105)

[Prerequisites 2](#_Toc132916106)

[Introduction 3](#_Toc132916107)

[Learning objectives 3](#_Toc132916108)

[Prerequisites 3](#_Toc132916109)

[GitHub Enterprise features 4](#_Toc132916110)

[Developer workflow basics 4](#_Toc132916111)

[Upkeep features 5](#_Toc132916112)

[How can you enable upkeep features? 6](#_Toc132916113)

[Automation features 8](#_Toc132916114)

[What are your automation responsibilities as an administrator? 8](#_Toc132916115)

[Identify organization strengths with activity statistics 9](#_Toc132916116)

[Help use tools as an administrator 10](#_Toc132916117)

[What are your tooling responsibilities as an administrator? 10](#_Toc132916118)

[Tools that focus on continuous integration and continuous delivery 11](#_Toc132916119)

[Support for GitHub Enterprise 12](#_Toc132916120)

[Availability of GitHub Enterprise Support 12](#_Toc132916121)

[Administrative responsibilities or GitHub Enterprise Support? 12](#_Toc132916122)

[Account 12](#_Toc132916123)

[Security 13](#_Toc132916124)

[Abuse 13](#_Toc132916125)

[Create and assign priority to a support ticket 13](#_Toc132916126)

[Other data that can accompany tickets 13](#_Toc132916127)

[Resolve and close support tickets 14](#_Toc132916128)

[Scale your enterprise deployment 15](#_Toc132916129)

[More about GitHub Enterprise single sign-on capabilities 15](#_Toc132916130)

[Use the GitHub APIs to extend your administrator capabilities 15](#_Toc132916131)

[Automate changes to organization members' access with SCIM 16](#_Toc132916132)

[Manage enterprise accounts with GraphQL 16](#_Toc132916133)

[Use GitHub applications and actions to help scale your enterprise organization 16](#_Toc132916134)

[Contrast marketplace applications and actions 17](#_Toc132916135)

[Tips for locating apps and actions on the GitHub Marketplace 17](#_Toc132916136)

[GitHub Enterprise Managed Users 19](#_Toc132916137)

[Access of managed users within your enterprise 19](#_Toc132916138)

[Authentication and user provisioning 20](#_Toc132916139)

[Other use cases for EMU accounts 21](#_Toc132916140)

[Summary 22](#_Toc132916141)

[Learn more 22](#_Toc132916142)

**GitHub administration for enterprise support and adoption**

In this module, you'll learn about some fundamental features and best practices that can improve the administrative operations of your GitHub Enterprise.

## Learning objectives

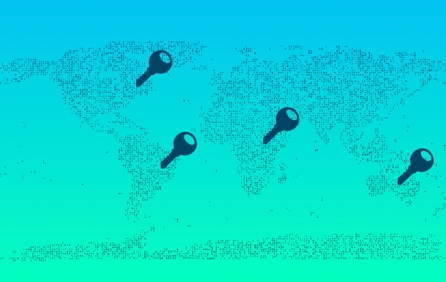
By the end of this module, you'll be able to:

* Set up authentication with SAML single sign-on and a connection between Enterprise Server and Enterprise Cloud with GitHub Connect.
* Automate security and version updates for your project's dependencies, as well as administrative duties and user interactions with GitHub.
* Identify when to open a ticket with GitHub Enterprise Support to resolve issues.
* Recognize the capabilities of centrally managing your GitHub enterprise members through GitHub Enterprise Managed User accounts (EMUs).

## Prerequisites

* A GitHub account
* The ability to navigate and edit files in GitHub
* A repository

# Introduction



As your organization's GitHub administrator, one of your responsibilities is to take advantage of the tools GitHub provides to both businesses large and small. GitHub Enterprise creates a custom environment where you can package software development tools and features together to improve the daily operations within your organization.

Suppose your organization lands a major software development contract, it will need to expand its workforce and capabilities right away. However, with greater headcount comes more authentication and more code that will require review for critical errors and vulnerabilities. Your organization's leaders have decided that GitHub's Enterprise package provides the necessary enhanced service. As an administrator, you need to assist with the setup of the Enterprise automation and upkeep tools.

In this module, you'll learn about some GitHub Enterprise features that can make your administrative duties easier.

## Learning objectives

By the end of this module, you'll be able to:

* Set up authentication with SAML single sign-on and a connection between Enterprise Server and Enterprise Cloud with GitHub Connect.
* Automate security and version updates for your project's dependencies, administrative duties, and user interactions with GitHub.
* Learn how to Interact with GitHub Enterprise Support to resolve issues with accounts, security, and abuse.
* Recognize the capabilities of centrally managing your GitHub enterprise members through GitHub Enterprise Managed User accounts (EMUs).

## Prerequisites

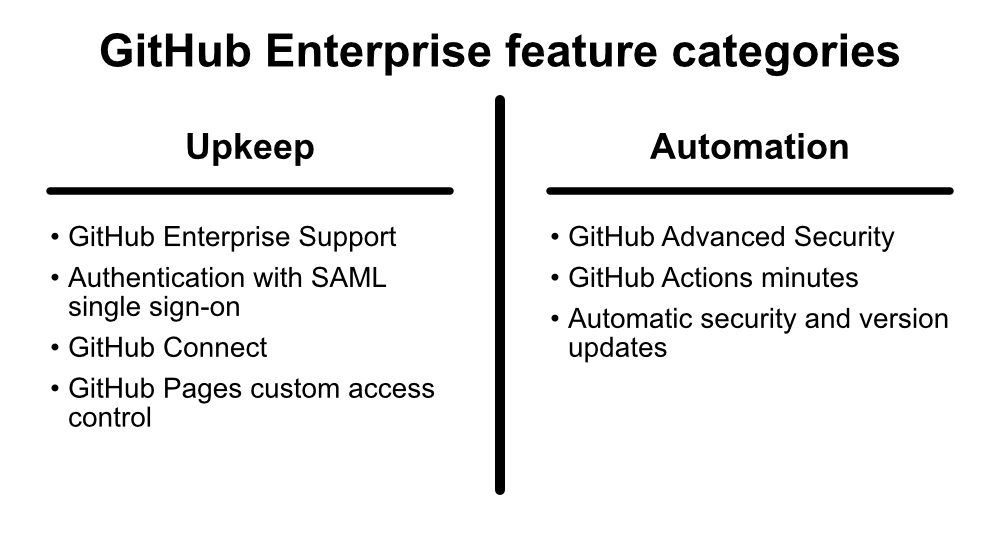
* A GitHub account
* The ability to navigate and edit files in GitHub
* A repository

# GitHub Enterprise features

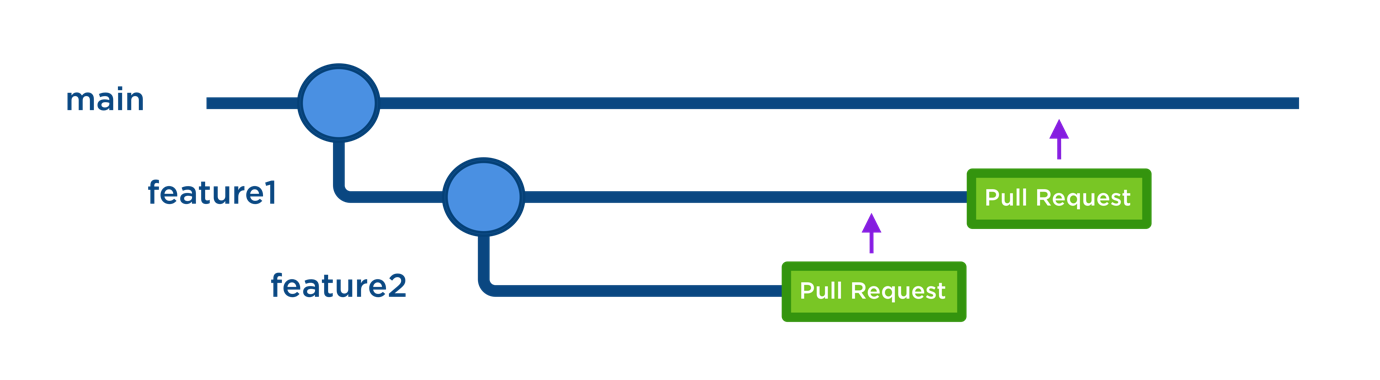
In this unit, you'll learn the difference between the two types of GitHub Enterprise features: upkeep and automation. As an administrator, you're responsible for knowing how and when to use them to make development easier.

Let's say your organization has formed a major partnership with a contractor who will help implement a new feature. As an administrator, you want to integrate each contractor employee with the internal team and help all collaborators cut down on repetitive tasks as they rapidly expand the code base. To accomplish these goals, you'll build a secure knowledge base by creating limited-access sites on GitHub Pages. You will also need to keep track of the number of GitHub Actions minutes that collaborators have used. Limited-access GitHub pages are an upkeep feature, and GitHub Actions minutes enable automation.

GitHub Enterprise is the GitHub product that provides the most assistance and features to developers. This means that, in addition to unique benefits, GitHub Enterprise also has all the features included at the GitHub Team pricing level.



## Developer workflow basics



Before we go further, you should understand the standard developer workflows that form the basis of GitHub collaboration, this will help you make sense of how GitHub Enterprise features facilitate those workflows. (A "workflow" here is distinct from "script workflows" that we'll discuss in unit 4.)

* When you create a repository, it has a "default" or "main" branch, which is the base branch for new pull requests and code commits.
* You create new branches from the default branch of your repository, usually to isolate the development of a feature. Authenticated users who have privileges in the repository can merge the changes after review of a pull request.
* As an administrator, you'll protect certain branches against deletion or force pushes to those branches. You can designate code owners who must review pull requests on those branches.
* Authenticated users without privileges in a public repository that wish to work with the repository's code base can "fork" the repository, creating an independent copy of the code base for outside use.
* Before committing changes, releasing new versions, and at other regular intervals, collaborators often run scripts on GitHub to complete common tasks, like scanning code for security issues. You can also automate these scripts.

## Upkeep features

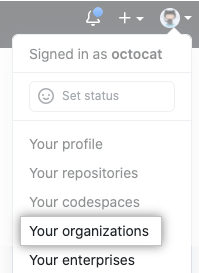
One of your primary roles as an administrator for a GitHub organization is to help minimize the friction users feel when interacting with GitHub. Friction refers to problems like difficulty signing in, finding files in the code base, and sharing project knowledge within the right circles.

If you enable your collaborators to manage a limited number of authentication credentials and portals, build a knowledge base without worrying about breaking non-disclosure agreements, and feel safe against threat actors, then they can get down to the business of the real work--coding and deployment. Upkeep features help achieve these tasks, and they include:

* GitHub Enterprise Support - the most in-depth level of support GitHub provides
* Authentication with SAML single sign-on and access provisioning with SAML or SCIM - makes it easier for users to authenticate and for administrators to ensure everyone has the appropriate level of access
* GitHub Connect - shares data between a GitHub Enterprise Server instance and the GitHub Enterprise Cloud, enabling additional redundancy
* Access control for GitHub Pages sites - allows you to limit the sharing of your organization's documentation

Some features are available to organizations running an instance of GitHub Enterprise Server on their own local server, but more features are available to organizations using GitHub Enterprise Cloud.

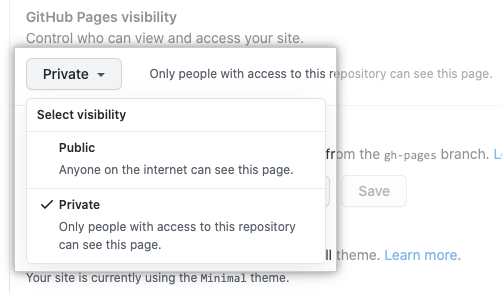
### How can you enable upkeep features?



You'll need to prepare to use upkeep features. Start with authentication using SAML single sign-on.

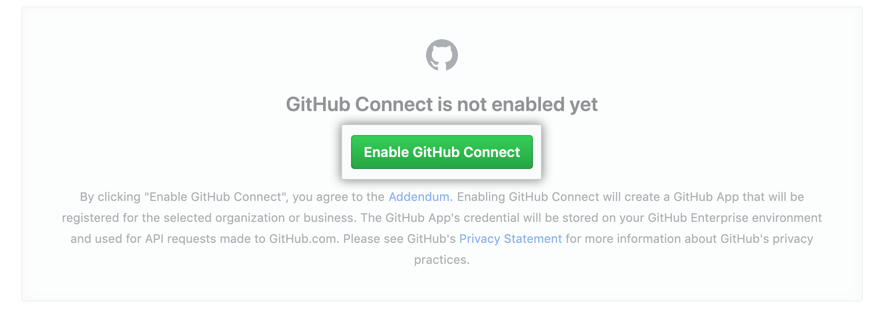
1. Connect your identity provider to your GitHub organization.
2. Access the settings for your organization.
3. Review the existing security configurations for that organization.
4. If you want to unify all the Enterprise organizations under SAML single sign-on, select **Require SAML authentication** and provide the requested technical information (sign on URL, optional issuer, and public certificate).

Then, secure the knowledge base for any of your private repositories that exist on the GitHub Enterprise Cloud.



1. Navigate to your site's repository and access its settings.
2. Change **GitHub Pages visibility** to "Private."
3. Verify your privacy by checking that no URLs appear under **GitHub Pages**.

Finally, if you're running an instance of GitHub Enterprise Server and also using the GitHub Enterprise Cloud, use GitHub Connect to enable unified search and contributions between the two platforms.



1. Sign in to GitHub Enterprise Server and GitHub.com.
2. In Enterprise Server, access **Enterprise settings** and click "Enable GitHub Connect."
3. Choose the organizations you want to connect from the list.

GitHub Support can assist you with managing users' accounts, security threats, and abuse of social features. You'll learn more about these use cases in the next unit.

## Automation features

While upkeep features make using GitHub easier for your organization, automation features make collaborating on creative work easier and safer. Automation features decrease the need for manual intervention in common programming tasks. For example, programmers have a strong demand for automated processes that increase the security of the code bases for their projects. As an administrator, you can enable and track the use of several built-in GitHub processes that serve this purpose.

Automation features include:

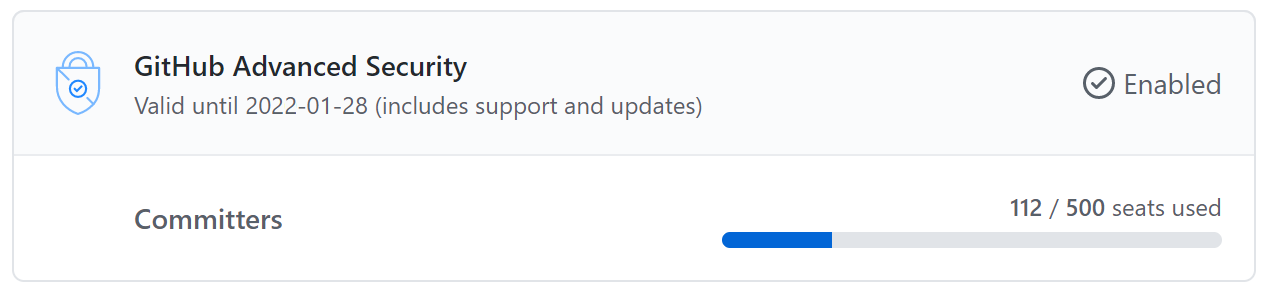
* GitHub Advanced Security (GHAS) - a specialized add-on to GitHub Enterprise that scans code and other resources for errors, security vulnerabilities, and potential sensitive data
* GitHub Actions minutes - expand the kinds of actions you can automate in response to common GitHub events
* Automatic security and version updates

Like with upkeep, certain automation features are limited to organizations using GitHub Enterprise Cloud.

### What are your automation responsibilities as an administrator?

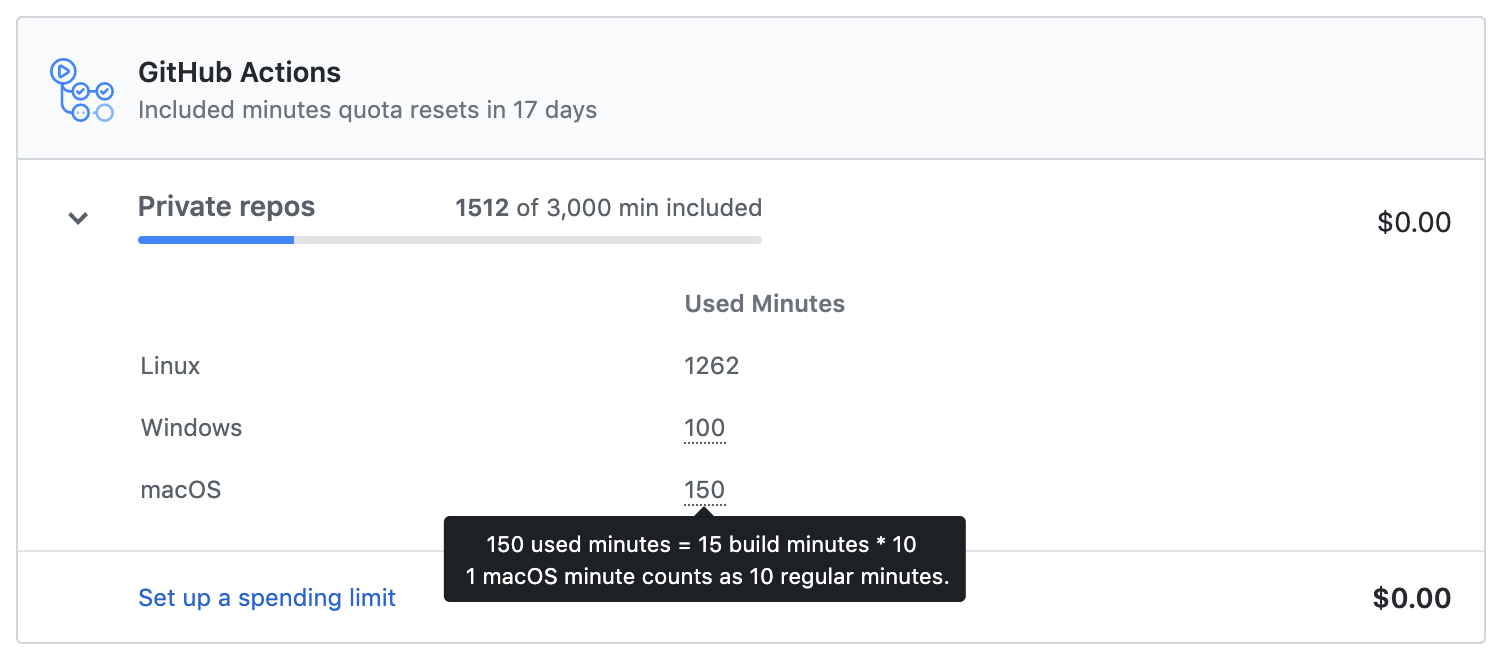
During new feature development, collaborators across your organization will take advantage of GitHub Actions scripts, Advanced Security scanning (if purchased), and automatic updates to dependencies. You are responsible for keeping track of the rate at which your coworkers are using these services, because GitHub provides these features to specific numbers of seats and within a given runtime.

You can track your GitHub Enterprise organization's use of GitHub Advanced Security scanning by checking in the settings for **Your enterprises**.



1. Select an Enterprise organization in **Your enterprises** and access **Settings**.
2. Check the number of seats your Enterprise organization is using. If you're near to or exceeding the limit your organization has purchased, discuss with organizational stakeholders.
3. Select "GitHub Advanced Security" in **Billing** in the left sidebar to learn more details about Advanced Security use, like whether an Enterprise organization has many committers using GHAS only in that organization.

Tracking users' GitHub Actions use is even easier. Your Enterprise account allows 50,000 minutes of GitHub Actions runtime per month and 50 GB of Actions storage. Check in the settings for **Your enterprises** to learn how much of that allotment your organizations have that month.



1. Select an Enterprise organization in **Your enterprises** and access **Settings**.
2. Select "GitHub Actions" in **Billing** to see how many minutes or how much storage your organizations have used. If you're near to or exceeding the limit, discuss with organizational stakeholders.
3. Keep your collaborators' operating systems in mind; Windows runners use minutes at twice the rate of Linux runners, and macOS runners use minutes at 10 times the rate of Linux runners.

### Identify organization strengths with activity statistics

You can track more statistics than just feature use. A user within a GitHub Enterprise Cloud organization can learn about the overall health of their organization by using the beta feature "organization activity insights" and the feature "organization dependency insights." As an administrator, leaders of your GitHub Enterprise Cloud organization may expect you to track these insights.

Organization activity insights display issue and pull request activity, top programming languages used, and other data about where collaborators spend their time. This knowledge is useful when deciding where to direct collaborators to serve as additional resources for a complex feature deployment, or whether additional training might expand a team's language capabilities.

Access your organization's activity insights by choosing an Enterprise Cloud organization in **Your organizations**. Then, click **Insights**. You can view the past week, month, or year's data for up to three organizations at once.

Organization dependency insights are available when you've enabled the Dependency Graph, and display information about vulnerabilities, licenses, and other data about open source projects integrated into or related to your organization's projects. These insights can help identify if too many of your dependencies have outstanding security advisories, prompting a discussion about how to better secure your code base.

Access your organization's dependency insights by choosing an Enterprise Cloud organization in **Your organizations**. Then, click **Insights** and click the **Dependencies** tab. You can view all dependency insights, **Open security advisories**, or explore more granular data.

## Help use tools as an administrator

In addition to managing the built-in features that GitHub provides to Enterprise organizations, administrators can help choose and implement tools built by independent developers.

In this context, a "tool" is usually an extension (sometimes called an "integration") or an action that helps improve a user or team's interaction with the GitHub workflow. An extension might connect GitHub with a full-featured editor like Visual Studio Code, allowing easier pull request management, or with a project management tool like Jira Cloud, making it possible to quickly follow up on Jira tracking tickets.

### What are your tooling responsibilities as an administrator?

You can install integrations in your personal account or in organizations that you own. You can also install GitHub Apps from a third-party in a specific repository where you have admin permissions or that is owned by your organization. This means that, as an administrator, you'll be in a position to approve or deny adding integrations or tools to the GitHub workflow.

Third-party tools usually require permissions at the repository level, and these permissions may change. If a tool's developer changes its permissions and the permissions are for a repository only, you'll be able to review and accept the new permissions.

The best way to find trustworthy tools for your Enterprise organization, and to construct workflows from trustworthy tools, is to take advantage of the GitHub Marketplace's categorization system to find tools that have received independent security verification.

Although anyone can publish an action in the GitHub Marketplace, publishers must complete a verification process in order to be listed under the "Verified Creator" category in the Marketplace. This process includes establishing a line of communication between GitHub and the publisher, demonstrating that the publisher meets basic security requirements, and verifying the publisher's domain. (However, GitHub does not analyze apps and actions on the GitHub Marketplace.)



The verified creator badge, which appears as a check mark next to the publisher's name in a listing on the GitHub Marketplace, signifies verification status.

* For GitHub Apps, the badge means that GitHub has verified the publisher's domain and email address, and the publisher has required two-factor authentication for their organization.
* For GitHub Actions, the badge means that GitHub is in [active partnership](https://partner.github.com/) with the organization.

### Tools that focus on continuous integration and continuous delivery

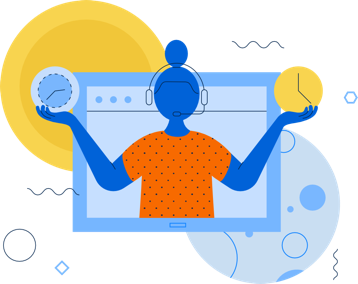
Your team probably places a high priority on building a "CI/CD pipeline"--a series of automated workflows that help DevOps teams cut down on manual tasks:

* Continuous integration (CI) automatically **builds**, **tests**, and **integrates** code changes within a shared repository; then
* Continuous delivery (CD) automatically **delivers** code changes to production-ready environments for approval; or
* Continuous deployment (CD) automatically **deploys** code changes to customers directly.

Your administrative duties will likely extend to approving, implementing, and maintaining tools that focus entirely on improving this pipeline. With that in mind, GitHub Enterprise organizations that make the most of their GitHub Actions minutes and automatic security and version updates will choose tools that meet these criteria:

* Tools should use **open standards**, because these tools are easiest to teach
* Tools should use **dynamic variables**, so you can define variables externally and effect major improvements with minor changes
* Tools should be **portable and flexible**, to adapt easily to a change in cloud architecture

# Support for GitHub Enterprise



In this unit, you'll learn about the enhanced support available with GitHub Enterprise, and common administrator duties when interacting with GitHub Enterprise Support. You can apply this knowledge whether you're operating a private copy of GitHub contained within a virtual appliance ("GitHub Enterprise Server") or you've deployed GitHub Enterprise services in GitHub's cloud ("GitHub Enterprise Cloud").

## Availability of GitHub Enterprise Support

The table below shows the availability of [GitHub Enterprise Support](https://enterprise.github.com/support).

|  |  |  |
| --- | --- | --- |
| **Availability** | **GitHub Enterprise Support** | **Premium and Premium Plus Support** |
| Days of operation | Monday through Friday | Every day |
| Time of initial response (High-level issues) | Within eight hours | Within four hours |
| Time of initial response (Urgent-level issues) | Within eight hours | Within 30 minutes |

[Premium and Premium Plus Support](https://github.com/premium-support) customers also receive other benefits.

## Administrative responsibilities or GitHub Enterprise Support?

Knowing where to go for support when an issue arises can be confusing. Is this problem an issue for the organization's administrator or is it something that GitHub Support can handle? GitHub Support can help troubleshoot issues that arise on GitHub Enterprise Server for three areas; account, security, and abuse issues.

### Account

Human error-related authentication problems are an example of account issues. When a user is locked out of their account and failed to retain keys to back up their two-factor authentication process, GitHub technical support can help you prove the user's identity.

### Security

You can request GitHub support when responding to threat actors hacking an account. If the problem is a security issue, GitHub support technicians can provide help in rolling back damage to organization repositories and settings.

### Abuse

Abuse issues involve responding to violations of the site's terms of service or organizational policies in GitHub's social setting--someone is using GitHub's communication's features to abuse other users. GitHub support technicians can help you evaluate the situation and can remove harmful content or ban abusive actors.

On the other hand, you should deal with some issues internally. You'll administer tasks like integrating CI/CD servers or internal tools, hardware setup, writing scripts, configuring SAML or other external authentication systems, working with Open Source projects, resolving problems that require immediate intervention, and running command-line utilities (except ghe-dbconsole).

## Create and assign priority to a support ticket

Once you have determined that it's time to get help from GitHub Enterprise Support, use the [GitHub Enterprise Support Portal](https://enterprise.githubsupport.com/hc/en-us) to submit support tickets.

The goal of the support ticket submission process is to provide as much issue-related information as possible to GitHub support technicians. As with creating support tickets in other information technology environments, the most valuable information is:

* Steps to reproduce the issue
* Any special circumstances surrounding the discovery of the issue (for example, the first occurrence or occurrence after a specific event, frequency of occurrence, business impact of the problem, and suggested urgency)
* Exact wording of error messages

GitHub prioritizes support tickets in the following manner:

* **Urgent** tickets report critical system failure.
* **High**-tickets report problems impacting business operations or critical bugs, including removing sensitive data (commits, issues, pull requests, uploaded attachments) from your own accounts and organization restorations.
* **Normal**-tickets request account recovery or spam unflagging, report user login issues, and report non-critical bugs.
* **Low**-tickets ask general questions and submit requests for new features, purchases, training, or health checks.

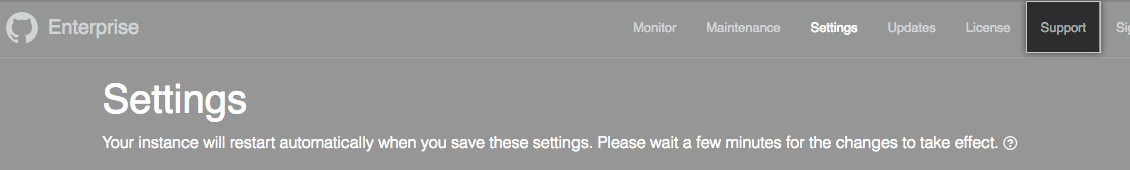
## Other data that can accompany tickets

Diagnostic files and support bundles provide more information to GitHub support technicians about a server's settings, environment, statistics, and logs.

The settings and history of a GitHub Enterprise environment are helpful when GitHub Enterprise Support needs to assist the administration with rolling back the environment to a state before damage occurred (most often related to security problems).

Logs present in support bundles retain information from the past two days, but in cases when GitHub support technicians require more data, it is possible to generate an extended support bundle containing logs that retain information from the past seven days.

Creating a basic diagnostic file from the Management Console is as simple as logging into the Console, selecting **Support**, and selecting **Download diagnostics info**. You can also obtain a support bundle by choosing **Download support bundle** instead.



The best way to provide this support bundle to GitHub Enterprise Support as an administrator is to select the relevant enterprise from **Your enterprises**, select **Enterprise licensing** under **Settings**, and then select **Upload a support bundle** in the GitHub Enterprise Help section of the webpage. This user-friendly direct line to GitHub Support is the interface you should use first when GitHub Enterprise Support requests a support bundle in response to your issue.

## Resolve and close support tickets

[The following information appears at the GitHub Doc "About GitHub Premium Support for GitHub Enterprise Cloud."](https://docs.github.com/en/github/working-with-github-support/about-github-premium-support-for-github-enterprise-cloud#resolving-and-closing-support-tickets)

GitHub Premium Support may consider a ticket solved after providing an explanation, recommendation, usage instructions, or workaround instructions.

If you use a custom or unsupported plug-in, module, or custom code, GitHub Premium Support may ask you to remove the unsupported plug-in, module, or code while attempting to resolve the issue. If the problem is fixed when the unsupported plug-in, module, or custom code is removed, GitHub Premium Support may consider the ticket solved.

GitHub Premium Support may close tickets if they're outside the scope of support or if multiple attempts to contact you have gone unanswered. If GitHub Premium Support closes a ticket due to lack of response, you can request that GitHub Premium Support reopen the ticket.

# Scale your enterprise deployment

In this unit, you'll learn more about the GitHub Enterprise features that support your Enterprise organization as it grows. You will also learn how to install tools from the GitHub Marketplace to help enhance your Enterprise organization.

As we discussed previously, a change in your organization's headcount, like a new contractor partnership, can increase the friction between GitHub and users or administrators. Suddenly, many more people require secure access to the code base, and as an administrator, you'll be responsible for tasks like providing and revoking access to user identities during the defined work period, using GitHub APIs to query the organization's audit log for compliance purposes, and installing GitHub applications from the Marketplace Azure Pipelines to establish standard workflows for deployment.

## More about GitHub Enterprise single sign-on capabilities

Previously, you learned that the first thing you should do as an administrator for a GitHub Enterprise organization is to enable authentication using SAML single sign-on. When your organization has deployed GitHub Enterprise services in GitHub's cloud ("GitHub Enterprise Cloud"), users may authenticate with their identity provider (IdP), which interfaces with GitHub's SAML single sign-on (SSO). Organization owners and administrators centrally manage users' identities, allowing review and revocation of SAML entities.

Employing an identity provider, like Okta, streamlines administrators' responsibilities to authenticate and provide ease of use to organization members. Use of an IdP provides security benefits including viewing and revoking a member's linked identity, active sessions, and authorized credentials.

You'll get the most out of this feature as your Enterprise organization scales up, because it's possible to perform those actions from a centralized location on GitHub. All you have to do is select your organization from **Your organizations** and then select a user from that organization's **People** tab. Under the **SAML identity linked** menu, you can identify the credentials through which the user has verified their identity and revoke those credentials if necessary.

**Warning:** If your organization uses team synchronization, revoking a person's SSO identity will remove that person from any teams mapped to IdP groups.

## Use the GitHub APIs to extend your administrator capabilities

GitHub Enterprise Server supports the REST and GraphQL APIs. Working with these APIs can automate many administrative tasks, including performing changes to the Management Console, configuring LDAP sync, automating changes to organization members' access, and managing your enterprise account to monitor and log important changes to the organization.

Those last two capabilities will be most relevant to you as your GitHub Enterprise organization enters a growth period, because they deal with security and compliance.

### Automate changes to organization members' access with SCIM

The System for Cross-Domain Identity Management (SCIM) standard uses the Rest API and common JSON requests to automate or streamline the provisioning and deprovisioning of users within an organization. This means less manual interaction with user data and credentials by administrators, increasing response time and decreasing the possibility of security issues.

The following identity providers are compatible with the GitHub SCIM API for organizations:

* Azure AD
* Okta
* OneLogin

### Manage enterprise accounts with GraphQL

Use the Enterprise Accounts API and the Audit Log API to help monitor and make changes in your organizations and maintain compliance. These APIs are only available as GraphQL APIs.

The enterprise account endpoints work for both GitHub Enterprise Cloud and GitHub Enterprise Server.

GraphQL allows you to request and return just the data you specify. For example, you can create a GraphQL query, or request for information, to see all the new organization members added to your organization. Or you can make a mutation, or change, to invite an administrator to your enterprise account.

With the Audit Log API, you can monitor when someone:

* Accesses your organization or repository settings.
* Changes permissions.
* Adds or removes users in an organization, repository, or team.
* Promotes users to admin.
* Changes permissions of a GitHub App.

The Audit Log API enables you to keep copies of your audit log data. For queries made with the Audit Log API, the GraphQL response can include data for up to 120 days.

## Use GitHub applications and actions to help scale your enterprise organization

Installing the right applications or actions in your GitHub Enterprise organization to enable connection to external services can automate a series of complex actions. For instance, a Marketplace action can report an incident to a server monitoring service like Datadog. The service can then signal the creation of an issue in the proper GitHub repository, populate the issue with information from logs, assign it to a predesignated troubleshooter, and flag the server monitoring service to check for the event's reoccurrence after a version update.

### Contrast marketplace applications and actions

Actions are the most basic tools that an administrator can enable from the GitHub Marketplace.

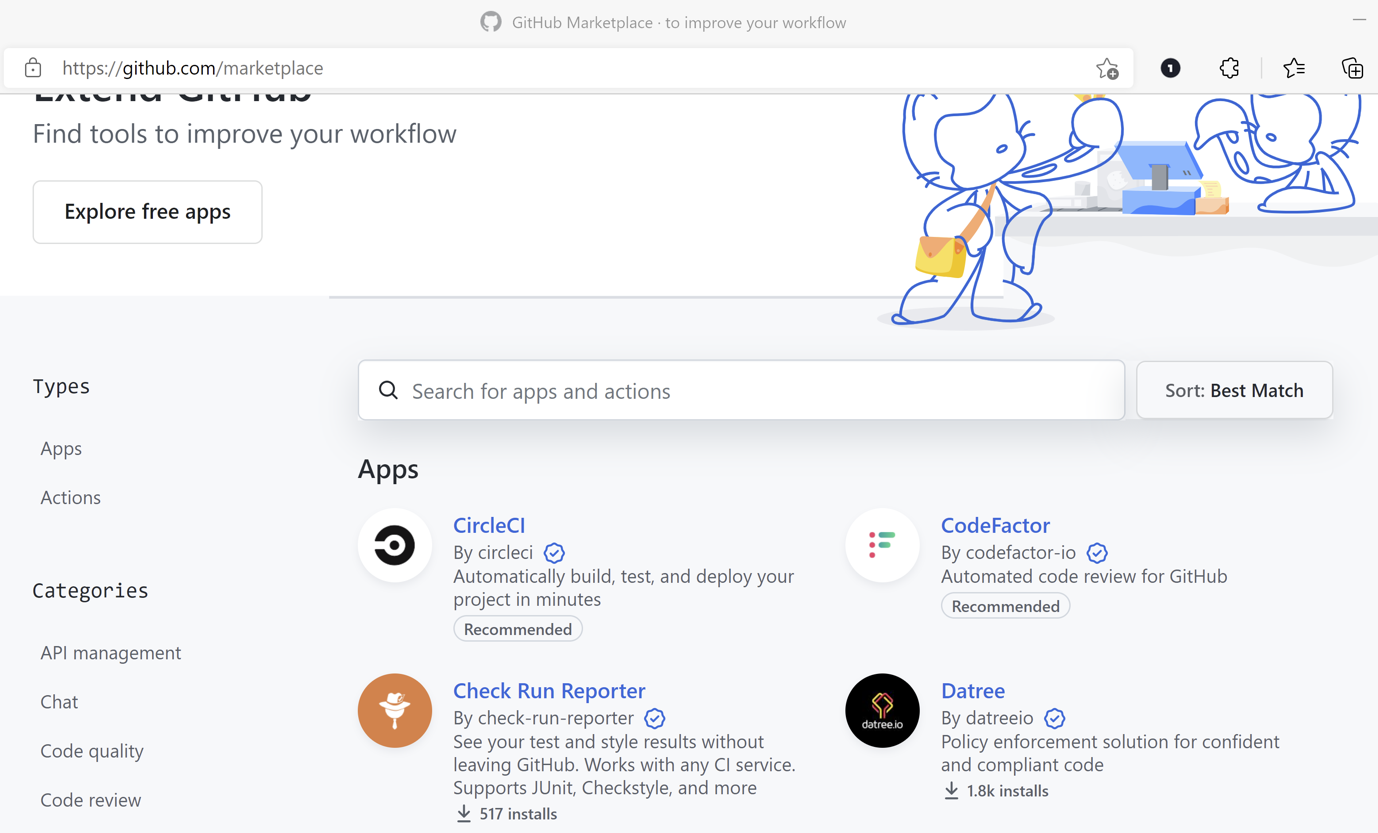
* They're reusable scripts that form the basic building blocks of a workflow, or a larger script file that defines automated responses to known events. They usually execute simple responses to simple triggers (like creating an issue or pull request on GitHub itself after a timer expires or an external system you've integrated sends a message).
* Running actions consumes GitHub Actions minutes (as discussed in previous units), and a cloud-hosted organization that has purchased a GitHub Enterprise plan has access to 50,000 GitHub Actions minutes.
* Actions obtain access to your organization's repositories on a limited basis, like per instance of a target event or within a defined period.
* A runner hosted by GitHub or a runner hosted on a server under your control can run actions as part of a workflow, and you also have the ability to modify or otherwise maintain these scripts.
* Actions don't have a user interface and don't extend the user's abilities to interact with GitHub.

GitHub applications are more complex tools that sacrifice customization and simplicity of implementation for greater functionality.

* Your organization consumes applications on a software-as-a-service basis (through subscriptions that you purchase in the GitHub Marketplace as an administrator).
* Instead of manually providing limited-lifetime authorization to access your organization's repositories (as with actions), you'll authorize an ongoing trust relationship with the application.
* Some applications interact with your repositories in the background, but most have their own custom user interface.
* Custom-built applications built and installed by an organization can use GitHub's REST and GraphQL APIs and interact with Git through cloning, updating and pushing a repository.
* Applications don't consume resources delivered through a GitHub Enterprise plan.

### Tips for locating apps and actions on the GitHub Marketplace

Let's say you want to reduce the workload for the DevOps teams within your organization because deadlines are getting tighter and capable users need to be able to shift to help with coding at any time. There's probably an app or action for that in the GitHub Marketplace! Here are a few tips for locating the right tools for the job.



* **App or action?** Determine the scope of the problem you're trying to solve. If you need to automate a repetitive task in response to an event in your repository, you probably need an action. If you want to make it easier to complete numerous related tasks, or you want to extend GitHub's functionality, you probably need an application. Filter the available tools on the Marketplace by choosing between **Apps** and **Actions** in the **Types** section of the Marketplace sidebar.
* **Do you want to spend money?** Depending on your budget, you might seek apps that are available on a free or trial basis. Alternately, some of the most popular apps on the Marketplace have versions that you must pay for on a users-per-month basis. Narrow your results along these lines by expanding the **Filters** section of the Marketplace sidebar and choosing between **Free**, **Free Trials**, **GitHub Enterprise** (apps with special functionality for organizations with a GitHub Enterprise plan), and **Paid**. (There aren't any paid actions.)
* **Do you know the part of the process you want to improve?** The **Categories** section of the Marketplace sidebar allows you to filter by type of business operation, and it can also show **Recently added** apps and actions.

From the example, it sounds like you need a tool to improve the general CI/CD process, and because the original scenario supposes that your organization has landed a major software development project, team leaders are likely willing to spend money to build the team's infrastructure. An app is more likely than an action to provide general or varied process improvements, CI/CD is the category of business operation you want to improve, and you can pay for the tool. Further, as an administrator for a GitHub Enterprise organization, extended functionality may be available from specific apps. App, CI/CD, Paid or GitHub Enterprise--are also all filters in the Marketplace.

**GitHub Enterprise Managed Users**

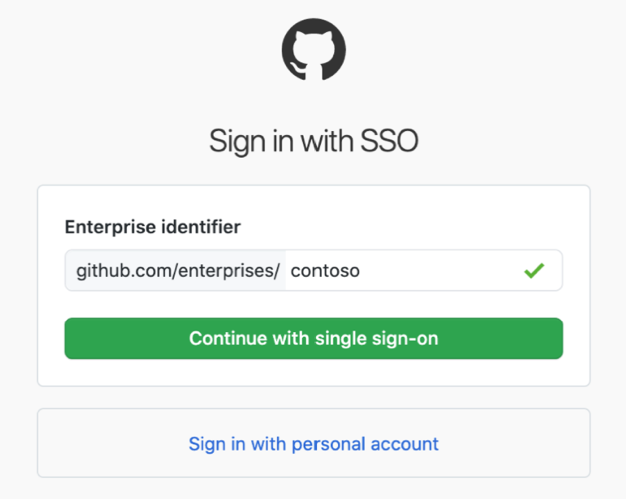
Here, you'll explore the capabilities of centrally managing your GitHub enterprise members through GitHub Enterprise Managed Users.

GitHub Enterprise Managed Users (EMUs) are GitHub Enterprise Cloud accounts owned by companies and not individual users. These accounts are controlled, managed, and audited by the company’s GitHub administrators. Using EMUs, you can connect your GitHub Enterprise Account to an identity provider (IdP); either Azure Active Directory or Okta. From here, you'll be able to centrally manage the user provisioning lifecycle and create a true single sign-on experience for your members and contractors. To ensure everything works smoothly, Admins have complete jurisdiction over the EMU accounts and their corresponding private repositories.

Because these EMU accounts are owned by the company, an administrator of the company can grant EMU access to any organization within the enterprise. In addition, you can control the profile information of your EMUs, including usernames, first and last names, and emails, all from your connected IdP. In short, With an EMU account, you can reduce the number of tasks that need to be manually duplicated from your IdP on GitHub.

Once provisioning is enabled in your IdP, you will be able to accomplish the following tasks.

* Manage team membership and repository access
* Control profile and personal data
* Provision, update, and deprovision IdP security groups to team and user memberships in GitHub EMU
* Simplify authentication with single sign-on (SSO)



### Access of managed users within your enterprise

In addition to the above mentions about EMU access, permissions, and restrictions. Here is a high-level overview of the differences between managed users and regular users within GitHub Enterprise.

|  |  |  |
| --- | --- | --- |
|  | **GitHub Enterprise Managed Users (EMU)** | **GitHub Enterprise Account** |
| **Ownership** | Created for and controlled by the identity provider. | Multiple authorization and authentication initiatives required in GitHub to ensure secure enterprise access. |
| **Membership** | Automated user provisioning syncs IdP group membership with GitHub teams. | Administrators manage user membership on teams in the enterprise's organizations on GitHub. |
| **Policies** | Policies can be created at the enterprise level that apply to all users on the account. | Users are required to confirm compliance with GitHub user policies. |
| **Audit log** | Enterprise owners can audit all of the managed users' actions on GitHub.com | Only audit log activity that happens in the enterprise account is available for your review. |
| **Privacy** | No public repositories on EMU-enabled enterprises. | Enterprise users can create public repositories. |
| **SSO** | True single sign-on experience for your users. | Multiple authentication steps required for your users to sign on. |
| **Metadata** | User GitHub metadata is controlled by the enterprise and cannot be modified by managed users. This metadata includes email addresses, display names, and usernames. | Enterprise users create their own account, provide email, and choose their GitHub handle. |

### Authentication and user provisioning

With managed users, you'll provision user accounts for your enterprise's members on GitHub.com directly from your IdP. To access resources in your enterprise, all members will authenticate to GitHub.com solely through your IdP.

To manage identity and access for your enterprise, you'll need to configure an application on your IdP and your enterprise on GitHub.com. Authentication requires SAML SSO and user provisioning requires System for Cross-domain Identity Management (SCIM). After you complete the configuration, your IdP will communicate with GitHub.com each time you make the following changes.

* When you assign the application to a user on your IdP, your IdP will create a user account as a member of your enterprise on GitHub.com and send a notification email to the user.
* When you update information associated with a user's identity on your IdP, your IdP will update the user's account on GitHub.com.
* When you unassign the application from a user or deactivate a user's account on your IdP, your IdP will communicate with GitHub.com to invalidate any SAML sessions and disable the member's account.

### Other use cases for EMU accounts

There are several unique ways you can use EMU accounts within your organization, let's look at some common use cases for EMUs found within most companies.

#### Onboard new employees

Employees are constantly joining the company all across the organization, departments, and teams. As an administrator, you can provision an EMU account for a new employee with all of their required permissions in the morning of their first day. EMU accounts also allow you to assign a GitHub handle that matches their work email to make identification easy across all systems. All of this setup occurs within the IdP and not within GitHub. The benefit of this configuration is to configure these settings in the same automatic flow as their benefits, HR, and expense systems to create a fast, efficient, and consistent onboarding.

#### Off board employees

On the contrary to onboarding, when an employee leaves the company, an administrator can instantly suspend their GitHub account with the same flow used to turn off the rest of their work access and systems. This turn off includes HR, email, internal documents, and benefits. Since EMUs are governed by the IdP, the setup creates an efficient, fast, and secure method to remove access to sensitive source code when an employee leaves the company. This automatic turn-off prevents any lingering GitHub access.

#### Reduce accidental IP leakage

With the rise of open-source software contributions, employees may need to switch between projects with different user identities and permissions throughout the day. To remove the possibility of accidental IP leakage and contributions made to a project using the wrong credentials, EMUs clearly identify the current account and ensure that the employee cannot accidentally expose IP to personal or open-source repositories. This secures the company, IP, and the employee from accidental leakage.

#### Consultant administration

Similar to granting access to new employees with an EMU account, consultants can be granted access to an EMU account through the company's IdP. When their project is finished, they will lose access to GitHub using the same flow that they would lose access to everything else. This automatic removal prevents lingering access for days or even weeks after their contract expires. In addition, if the consultant is working with multiple clients at the same time, by using an EMU account the consultant won't be able to accidentally leak the company's IP into the other clients' repositories. This setup allows for tight control over access to the company's source code and IP while reducing lingering access or accidental IP leakage.

# Summary

The goal of this module was to help you understand how GitHub Enterprise can simplify some of your duties as a GitHub administrator. GitHub Enterprise includes tools for streamlining the operations of large organizations such as managing the increased administrative workload that comes with greater headcount.

In this module, you learned how to:

* Set up authentication with SAML single sign-on as well as a connection between Enterprise Server and Enterprise Cloud with GitHub Connect.
* Automate security and version updates for your project's dependencies, as well as administrative duties and user interactions with GitHub.
* Interact with GitHub Enterprise Support to resolve issues with accounts, security, and abuse.
* Recognize the capabilities of centrally managing your GitHub enterprise members through GitHub Enterprise Managed User accounts (EMUs).

## Learn more

Review the links below for more details about the information covered in this module.

* [GitHub product docs](https://docs.github.com/get-started/learning-about-github/githubs-products)
* [Submitting a ticket](https://docs.github.com/en/github/working-with-github-support/submitting-a-ticket#submitting-a-ticket-using-the-github-enterprise-support-portal)
* [Providing data to GitHub Support](https://docs.github.com/enterprise-server@3.1/admin/enterprise-support/receiving-help-from-github-support/providing-data-to-github-support)
* [Viewing and managing a member's SAML access to your organization](https://docs.github.com/en/organizations/granting-access-to-your-organization-with-saml-single-sign-on/viewing-and-managing-a-members-saml-access-to-your-organization)
* [Synchronizing a team with an identity provider](https://docs.github.com/en/organizations/organizing-members-into-teams/synchronizing-a-team-with-an-identity-provider-group)
* [GitHub API documentation on SCIM](https://docs.github.com/en/rest/reference/scim)
* [About SCIM](https://docs.github.com/en/enterprise-cloud@latest/organizations/managing-saml-single-sign-on-for-your-organization/about-scim-for-organizations)
* [AuditEntry interface](https://docs.github.com/en/enterprise-server@3.1/graphql/reference/interfaces#auditentry)