Table of Contents

[GitHub Enterprise Cloud vs GitHub Enterprise Server 1](#_Toc131068029)

# GitHub Enterprise Cloud vs GitHub Enterprise Server

The main differences between GitHub Enterprise Cloud and GitHub Enterprise Server are:

1. Deployment: GitHub Enterprise Cloud is hosted by GitHub, while GitHub Enterprise Server is self-hosted by the organization.
2. Management: GitHub Enterprise Cloud is managed by GitHub, while GitHub Enterprise Server is managed by the organization.
3. Control: GitHub Enterprise Cloud provides less control over the underlying infrastructure, while GitHub Enterprise Server provides greater control.
4. Customization: GitHub Enterprise Cloud provides limited customization options, while GitHub Enterprise Server provides more customization options.

**GitHub Enterprise Cloud is hosted by GitHub.**

**GitHub Enterprise Server is self-hosted.**

## Enterprise Account in GitHub Enterprise Cloud

An enterprise account is a type of account designed for organizations with multiple teams or business units that want to manage access and permissions centrally.

* You can create and manage multiple organizations within a single account and set permissions and access levels across all of them.
* You can also set up billing and usage monitoring for all your organizations in one place.

Additionally, enterprise accounts provide advanced security features such as SAML single sign-on, two-factor authentication, and security vulnerability scanning, which can help ensure the security of your repositories and data.

Some other key features of enterprise accounts include:

* Access to GitHub Connect, which allows you to connect your GitHub Enterprise Cloud instance with GitHub.com or other instances of GitHub Enterprise Server.
* Advanced auditing and reporting features, which allow you to monitor and track user activity across all your organizations.
* Granular permission controls, which allow you to define custom roles and permissions for different teams and users.
* Integration with third-party tools and services through GitHub Marketplace, which allows you to extend the functionality of your GitHub Enterprise Cloud instance.

In summary, an enterprise account in GitHub Enterprise Cloud provides organizations with the ability to manage multiple organizations, teams, and users centrally, with advanced security and customization options.

GitHub Enterprise Cloud includes an enterprise account, which allows you to manage multiple organizations. You can choose to let enterprise members create and manage their own personal accounts, or you can use Enterprise Managed Users.

Enterprise Managed Users is a feature of GitHub Enterprise Cloud that provides even greater control over enterprise members and resources. With Enterprise Managed Users, all members are provisioned and managed through your identity provider (IdP) instead of users creating their own accounts on GitHub. Organization and team membership can be managed using groups on your IdP. Managed user accounts are restricted to their enterprise and are unable to push code, collaborate, or interact with users, repositories, and organizations outside of their enterprise. For more information, see "[About Enterprise Managed Users](https://docs.github.com/en/enterprise-cloud@latest/admin/authentication/managing-your-enterprise-users-with-your-identity-provider/about-enterprise-managed-users)" in the GitHub Enterprise Cloud documentation.

## What is an IdP (Identity Provider)?

An IdP (Identity Provider) is a system or service that is responsible for authenticating and verifying the identity of users in a digital environment. In other words, an IdP is a service that manages and stores user identity information and provides this information to other systems or services that need to authenticate the user.

The primary role of an IdP is to act as a trusted source of user identity information. When a user tries to access a system or service, the IdP verifies their identity by asking for credentials such as a username and password, or by using other authentication methods such as multi-factor authentication.

Once the user's identity is verified, the IdP generates an authentication token, which is then passed to the requesting system or service. The requesting system or service can use this token to grant the user access to its resources and services.

Some examples of popular IdP services include Google, Microsoft Azure Active Directory, Okta, and Auth0. These services provide secure, scalable, and reliable identity management solutions for businesses and organizations of all sizes.

## What are encrypted secrets?

**Encrypted secrets are encrypted environment variables you can create to store sensitive information.**

Encrypted secrets are sensitive pieces of information, such as access tokens, API keys, and other credentials, that are stored securely within a GitHub repository. These secrets are encrypted to protect them from unauthorized access and ensure that they are only accessible to authorized users and workflows.

GitHub offers a feature called "secrets" which allows users to store and manage these encrypted secrets in a repository. Secrets can be used in GitHub Actions workflows to authenticate with external services, perform deployments, or access other resources.

Secrets can be added and managed through the GitHub web interface or the GitHub API. They are encrypted using the public key of a key pair unique to each repository, which ensures that only authorized users with the corresponding private key can decrypt and access the secrets.

By using encrypted secrets, users can avoid storing sensitive information in plain text within their repositories, which could be a security risk. It is important to follow best practices for managing and using secrets, such as limiting access to only authorized users, rotating secrets regularly, and revoking access to secrets when no longer needed.