Managing access for GitHub Enterprise Importer

In this article

Before you use GitHub Enterprise Importer, make sure you have appropriate access to both the source and destination of your migration.

About required access for GitHub Enterprise Importer

To protect your data, GitHub enforces specific access requirements to use GitHub Enterprise Importer. To prevent errors, you should review this article carefully and verify that you meet all of the requirements for the task you want to complete.

To run a migration, you need sufficient access to both the source and the destination for your migration. For repository migrations, the destination is an organization. For organization migrations, the destination is an enterprise account.

Before you can migrate from GitHub Enterprise Server 3.8 or higher for the first time, you also need someone with access to the Management Console to set up blob storage for your GitHub Enterprise Server instance.

For other tasks, you only need access to the target of the operation. For example, to grant the migrator role for an organization, you only need access to that organization.

To have sufficient access for either the source or destination, you need both of the following:

- A required role in the GitHub organization or enterprise account
- For Bitbucket Server, required permissions and SFTP or SMB access
- For GitHub products and Azure DevOps, a personal access token that can access the organization or enterprise account
    - The personal access token must have all the required scopes, which depend on your role and the task you want to complete.
    - If the source or destination uses SAML single sign-on for GitHub.com, you must authorize the personal access token for SSO.

Additionally, if you use IP allow lists with the source or destination, you may need to configure the allow lists to allow access by GitHub Enterprise Importer.

Required roles for GitHub

When migrating to or from GitHub products, different roles are required for different tasks.

Task	Enterprise owner	Organization owner	Migrator
Running a migration (source organization)		X	X
Running an organization migration (destination enterprise)	X		
Assigning the migrator role for repository migrations		X	
Running a repository migration (destination organization)		X	X
Downloading a migration log		X	X
Reclaiming mannequins		X	

Required permissions for Bitbucket Server

To migrate from Bitbucket Server, you need:

- The username and password of a Bitbucket Server account that has admin or super admin permissions
- If your Bitbucket Server instances runs on Linux, SFTP access to the Bitbucket Server instance (see "SSH keys"). In general, if you can access the server via SSH, then you can also use SFTP.
- If your Bitbucket Server instance runs on Windows, file sharing (SMB) access to the Bitbucket Server instance

SSH keys

If your Bitbucket Server instance runs on Linux, you must use an SSH key that meets the following requirements:

- Does not have a passphrase
- Uses one of the following ciphers
  - `aes256-ctr`
  - `3des-cbc`
  - `aes128-cbc`
  - `aes192-cbc`
  - `aes256-cbc`
  - `blowfish-cbc`
  - `twofish-cbc`
  - `twofish192-cbc`
  - `twofish128-cbc`
  - `twofish256-cbc`

- `arcfour`
- `arcfour128`
- `arcfour256`
- `cast128-cbc`
- `aes128-ctr`
- `aes192-ctr`

If you receive an error like `cipher name aes256-ctr for openssh key file is not supported` when running a migration, your SSH private key uses an unsupported cipher. For more information about how to generate a compatible private key, see "Troubleshooting your migration with GitHub Enterprise Importer."

Required scopes for personal access tokens

To run a migration, you need a personal access token that can access the destination organization (for repository migrations) or enterprise account (for organization migrations). If your source is a GitHub product or Azure DevOps, you also need another personal access token that can access the source organization.

For other tasks, such as downloading a migration log, you only need one personal access token that can access the target of the operation.

Personal access tokens for GitHub products

The scopes that are required for your GitHub personal access token (classic) depend on your role and the task you want to complete.

> **Note**: You can only use a personal access token (classic), not a fine-grained personal access token. This means that you cannot use GitHub Enterprise Importer if your organization uses the "Restrict personal access tokens (classic) from accessing your organizations" policy. For more information, see "Enforcing policies for personal access tokens in your enterprise."

Task	Enterprise owner	Organization owner	Migrator
Running a migration (source organization)	-	`read:org`, `repo`	`read:org`, `repo`
Running an organization migration (destination enterprise)	`read:enterprise`, `admin:org`, `repo`, `workflow`	-	-
Assigning the migrator role for repository migrations	-	`admin:org`	-
Running a repository migration (destination organization)	-	`repo`, `admin:org`, `workflow`	`repo`, `read:org`, `workflow`
Downloading a migration log	-	`repo`, `admin:org`, `workflow`	`repo`, `read:org`, `workflow`
Reclaiming mannequins	-	`admin:org`	-

Personal access tokens for Azure DevOps

To run a migration from Azure DevOps (ADO), your ADO personal access token must

have `work item (read)`, `code (read)`, and `identity (read)` scopes.

If you want to migrate from multiple organizations, allow the personal access token to access all accessible organizations. For more information, see Use personal access tokens in Microsoft Docs.

Creating a personal access token for GitHub Enterprise Importer

1. Verify that you have a sufficient role for the task you want to complete. For more information, see "Required roles."
2. Create a personal access token (classic), making sure to grant all the scopes required for the task you want to complete. You can only use a personal access token (classic), not a fine-grained personal access token. For more information, "Managing your personal access tokens" and "Required scopes for personal access token."
3. If SAML single sign-on is enforced for the organization(s) you need to access, authorize the personal access token for SSO. For more information, see "Authorizing a personal access token for use with SAML single sign-on."

Configuring IP allow lists for migrations

If you use IP allow lists with your migration source or destination, you may need to configure the lists to allow access to GitHub Enterprise Importer.

To configure IP allow lists correctly, please read the following sections carefully. Depending on your migration, more than one section may apply to you.

The source or destination of your migration is GitHub.com

You need to configure IP allow lists on GitHub.com if **both** of the following apply to your migration:

- The source or destination of your migration is GitHub.com
- The source or destination uses an IP allow list, either GitHub's IP allow list feature or your identity provider's (IdP) IP allow list restrictions (such as Azure CAP)

If you use GitHub's IP allow list feature, you must add the GitHub IP ranges below to the allow list for the source and/or destination organizations.

If you use your IdP's IP allow list to restrict access to your enterprise on GitHub.com, you should disable these restrictions in your enterprise account settings until after your migration is complete.

For more information, see "Managing allowed IP addresses for your organization" and "Restricting network traffic to your enterprise with an IP allow list."

The source of your migration is GitHub Enterprise Server

If the source of your migration is GitHub Enterprise Server, you do not need to add any GitHub IP ranges to your firewall configuration or the IP allow list on your GitHub Enterprise Server instance.

However, depending on the setup of your blob storage provider, you may need to update your blob storage provider's configuration to allow access to the GitHub IP ranges below.

Identifying GitHub's IP ranges

You'll need to add the following IP ranges to your IP allowlist(s):

- 192.30.252.0/22
- 185.199.108.0/22
- 140.82.112.0/20
- 143.55.64.0/20
- 40.71.233.224/28
- 2a0a:a440::/29
- 2606:50c0::/32
- 20.125.12.8/29 *(active from 00:00 UTC on November 8, 2023)*

You can get an up-to-date list of IP ranges used by GitHub Enterprise Importer at any time with the "Get GitHub meta information" endpoint of the REST API.

The `github_enterprise_importer` key in the response contains a list of IP ranges used for migrations.

For more information, see "Meta" in the REST API documentation.

Further reading

- "Roles in an organization"