

About ghe-migrator

In this article

Types of migrations

Migrated data

About migration of external authentication data

You can use `ghe-migrator` to transfer data from a *source* location (either a GitHub.com organization or a GitHub Enterprise Server instance) to a *target* GitHub Enterprise Server instance.

Types of migrations

There are three types of migrations you can perform:

- A migration from a GitHub Enterprise Server instance to another existing GitHub Enterprise Server instance. You can migrate any number of repositories owned by any user or organization on the instance. Before performing a migration, you must have site administrator access to both instances.
- A migration from a GitHub.com organization to a GitHub Enterprise Server instance. You can migrate any number of repositories owned by the organization. Before performing a migration, you must have [administrative access](#) to the GitHub.com organization as well as site administrator access to the target instance.
- *Trial runs* are migrations that import data to a [staging instance](#). These can be useful to see what *would* happen if a migration were applied to your GitHub Enterprise Server instance. **We strongly recommend that you perform a trial run on a staging instance before importing data to your production instance.**

Note: The use of ghe-migrator is **not recommended** for transferring a GitHub Enterprise Server instance between hypervisors. Instead, we suggest either backing up and restoring to the new location with GitHub Enterprise Server Backup Utilities, or creating a replica in the new location and then failing over to the replica appliance. For more information, see "[Configuring backups on your instance](#)", "[Creating a high availability replica](#)" and "[Initiating a failover to your replica appliance](#)."

Migrated data

With ghe-migrator, everything revolves around a repository. Most data associated with a repository can be migrated. For example, a repository within an organization will migrate the repository *and* the organization, as well as any users, teams, issues, and pull requests associated with the repository.

The items in the table below can be migrated with a repository. Any items not shown in the list of migrated data can not be migrated, including Git LFS assets.

Note: Fork relationships do not persist after a migration.

Data associated with a migrated repository	Notes
Users	@ mentions of users are rewritten to match the target.
Organizations	An organization's name and details are migrated.
Repositories	Links to Git trees, blobs, commits, and lines are rewritten to match the target. The migrator follows a maximum of three repository redirects. Internal repositories are migrated as private repositories. Archive status is unset.
Wikis	All wiki data is migrated.
Teams	@ mentions of teams are rewritten to match the target.
Milestones	Timestamps are preserved.
Project boards	Project boards associated with the repository and with the organization that owns the repository are migrated.
Issues	Issue references and timestamps are preserved.
Issue comments	Cross-references to comments are rewritten for the target instance.
Pull requests	Cross-references to pull requests are rewritten to match the target. Timestamps are preserved.
Pull request reviews	Pull request reviews and associated data are migrated.
Pull request review comments	Cross-references to comments are rewritten for the target instance. Timestamps are preserved. File-level comments are not migrated.
Commit comments	Cross-references to comments are rewritten for the target instance. Timestamps are preserved.
Releases	All releases data is migrated.
Actions taken on pull requests or issues	All modifications to pull requests or issues, such as assigning users, renaming titles, and modifying labels are preserved, along with timestamps for each action.
File attachments	File attachments on issues and pull requests are migrated. You can choose to disable this as part of the migration.
Webhooks	Only active webhooks are migrated.
Repository deploy keys	Repository deploy keys are migrated.
Protected branches	Protected branch settings and associated data are migrated.

About migration of external authentication data

If the source location for your migration is a GitHub product that uses LDAP or SAML authentication, `ghe-migrator` does not migrate external authentication data linked to user accounts. For more information about authentication options, see GitHub Enterprise Server, see "About authentication for your enterprise" in the [GitHub Enterprise Server docs](#) or the [GitHub Enterprise Cloud docs](#).

If you migrate to a destination instance and then configure external authentication, users must sign in to the destination instance with a user account that has the same username or user ID as the account on the source instance. Administrators can review the external attribute that an instance uses to map user account names from the Management Console. For more information, see "[Accessing the Management Console](#)."

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)