Upgrade requirements

In this article

Recommendations

Requirements

Next steps

Before upgrading GitHub Enterprise Server, review these recommendations and requirements to plan your upgrade strategy.

> **Notes:**
>
> - Upgrade packages are available at enterprise.github.com for supported versions. Verify the availability of the upgrade packages you will need to complete the upgrade. If a package is not available, contact GitHub Enterprise Support for assistance.
> - If you're using GitHub Enterprise Server Clustering, see "Upgrading a cluster" in the GitHub Enterprise Server Clustering Guide for specific instructions unique to clustering.
> - The release notes for GitHub Enterprise Server provide a comprehensive list of new features for every version of GitHub Enterprise Server. For more information, see the releases page.

Recommendations 🔗

- Include as few upgrades as possible in your upgrade process. For example, instead of upgrading from GitHub Enterprise 3.7 to 3.8 to 3.9, you could upgrade from GitHub Enterprise 3.7 to 3.9. Use the Upgrade assistant to find the upgrade path from your current release version.

- If you're several versions behind, upgrade your GitHub Enterprise Server instance as far forward as possible with each step of your upgrade process. Using the latest version possible on each upgrade allows you to take advantage of performance improvements and bug fixes. For example, you could upgrade from GitHub Enterprise 2.7 to 2.8 to 2.10, but upgrading from GitHub Enterprise 2.7 to 2.9 to 2.10 uses a later version in the second step.

- Use the latest patch release when upgrading. Browse to the GitHub Enterprise Server Releases page. Next to the release you are upgrading to, click **Download**, then click the **Upgrading** tab.

- Use a staging instance to test the upgrade steps. For more information, see "Setting up a staging instance."

- When running multiple upgrades, wait at least 24 hours between feature upgrades to allow data migrations and upgrade tasks running in the background to fully complete.

- Take a snapshot before upgrading your virtual machine. For more information, see "Upgrading GitHub Enterprise Server."

- Ensure you have a recent, successful backup of your instance. For more information, see the [GitHub Enterprise Server Backup Utilities README.md file](#).

Requirements 🔗

- You must upgrade from a feature release that's **at most** two releases behind. For example, to upgrade to GitHub Enterprise 3.9, you must be on GitHub Enterprise 3.8 or 3.7.
- When upgrading using an upgrade package, schedule a maintenance window for GitHub Enterprise Server end users.
- You can upgrade GitHub Enterprise Server to the latest patch release using a hotpatch.

You can use hotpatching to upgrade to a newer patch release, but not a feature release. For example, you can upgrade from 2.10.1 to 2.10.5 because they are in the same feature series, but not from 2.10.9 to 2.11.0 because they are in a different feature series.

Hotpatches do not generally require a reboot. If a hotpatch does require a reboot, the GitHub Enterprise Server release notes will indicate the requirement.

Hotpatches require a configuration run, which can cause a brief period of errors or unresponsiveness for some or all services on your GitHub Enterprise Server instance. You are not required to enable maintenance mode during installation of a hotpatch, but doing so will guarantee that users see a maintenance page instead of errors or timeouts. For more information, see "[Enabling and scheduling maintenance mode](#)."

- A hotpatch may require downtime if the affected services (like kernel, MySQL, or Elasticsearch) require a VM reboot or a service restart. You'll be notified when a reboot or restart is required. You can complete the reboot or restart at a later time.
- Additional root storage must be available when upgrading through hotpatching, as it installs multiple versions of certain services until the upgrade is complete. Pre-flight checks will notify you if you don't have enough root disk storage.
- When upgrading through hotpatching, your instance cannot be too heavily loaded, as it may impact the hotpatching process.
- Upgrading to GitHub Enterprise Server 2.17 migrates your audit logs from Elasticsearch to MySQL. This migration also increases the amount of time and disk space it takes to restore a snapshot. Before migrating, check the number of bytes in your Elasticsearch audit log indices with this command:

  ```
  curl -s http://localhost:9201/audit_log/_stats/store | jq ._all.primaries.store.
  ```

  Use the number to estimate the amount of disk space the MySQL audit logs will need. The script also monitors your free disk space while the import is in progress. Monitoring this number is especially useful if your free disk space is close to the amount of disk space necessary for migration.

Next steps 🔗

After reviewing these recommendations and requirements, you can upgrade GitHub Enterprise Server. For more information, see "[Upgrading GitHub Enterprise Server](#)."