GitHub Docs

Enterprise administrators / Monitor, manage, and update your appliance / Caching repositories / About repository caching

This version of GitHub Enterprise was discontinued on 2023-03-15. No patch releases will be made, even for critical security issues. For better performance, improved security, and new features, [upgrade to the latest version of GitHub Enterprise](). For help with the upgrade, [contact GitHub Enterprise support]().

About repository caching

You can increase the performance of Git read operations for distributed teams and CI farms with repository caching.

> **Note:** Repository caching is currently in beta and subject to change.

If you have teams and CI farms located around the world, you may experience reduced performance on your primary GitHub Enterprise Server instance. While active geo-replicas can improve the performance of read requests, this comes at the cost of limiting write throughput. To reduce load on your primary instance and improve write throughput performance, you can configure a repository cache, an asynchronous read-only mirror of repositories located near these geographically-distributed clients.

A repository cache eliminates the need for GitHub Enterprise Server to transmit the same Git data over a long-haul network link multiple times to serve multiple clients, by serving your repository data close to CI farms and distributed teams. For instance, if your primary instance is in North America and you also have a large presence in Asia, you will benefit from setting up the repository cache in Asia for use by CI runners there.

The repository cache listens to the primary instance, whether that's a single instance or a geo-replicated set of instances, for changes to Git data. CI farms and other read-heavy consumers clone and fetch from the repository cache instead of the primary instance. Changes are propagated across the network, at periodic intervals, once per cache instance rather than once per client. Git data will typically be visible on the repository cache within several minutes after the data is pushed to the primary instance. The `cache_sync` `webhook` can be used by CI systems to react to data being available in the cache.

You have fine-grained control over which repositories are allowed to sync to the repository cache. Git data will only be replicated to the locations you specify.

You can configure repository caching by creating a special type of replica called a repository cache. For more information, see "[Configuring a repository cache]()."