GitHub Docs

Enterprise administrators / Monitor, manage, and update your instance / Caching repositories / Configuring a repository cache

Configuring a repository cache

In this article

You can configure a repository cache for GitHub Enterprise Server by creating a new instance, connecting the repository cache to your primary instance, and configuring replication of repository networks to the repository cache.

About configuration for repository caching 🔗

You can configure repository caching by creating a special type of replica called a repository cache. Then, you can set data location policies that govern which repository networks are replicated to the repository cache.

Repository caching is not supported with clustering.

DNS for repository caches 🔗

The primary instance and repository cache should have different DNS names. For example, if your primary instance is at `github.example.com`, you might decide to name a cache `europe-ci.github.example.com` or `github.asia.example.com`.

To have your CI machines fetch from the repository cache instead of the primary instance, you can use Git's `url.<base>.insteadOf` configuration setting. For more information, see `git-config` in the Git documentation.

For example, the global `.gitconfig` for the CI machine would include these lines.

```
[url "https://europe-ci.github.example.com/"]
        insteadOf = https://github.example.com/
```

Then, when told to fetch `https://github.example.com/myorg/myrepo`, Git will instead fetch from `https://europe-ci.github.example.com/myorg/myrepo`.

Configuring a repository cache 🔗

1. Set up a new GitHub Enterprise Server instance on your desired platform. This instance will be your repository cache. For more information, see "Setting up a GitHub Enterprise Server instance."

2. Set an admin password that matches the password on the primary appliance and continue.

3 Click **Configure as Replica**.

4 Under "Add new SSH key", type your SSH key.

5 Click **Add key**.

6 Connect to the repository cache's IP address using SSH.

```
ssh -p 122 admin@REPLICA-IP
```

7 To generate a key pair for replication, use the `ghe-repl-setup` command with the primary appliance's IP address and copy the public key that it returns.

```
ghe-repl-setup PRIMARY_IP
```

8 To add the public key to the list of authorized keys on the primary appliance, browse to `https://PRIMARY-HOSTNAME/setup/settings` and add the key you copied from the replica to the list.

9 To verify the connection to the primary and enable replica mode for the repository cache, run `ghe-repl-setup` again.

- If the repository cache is your only additional node, no arguments are required.

```
ghe-repl-setup PRIMARY-IP
```

- If you're configuring a repository cache in addition to one or more existing replicas, use the `-a` or `--add` argument.

```
ghe-repl-setup -a PRIMARY-IP
```

10 To configure the repository cache, use the `ghe-repl-node` command and include the necessary parameters.

- Set a `cache-location` for the repository cache, replacing *CACHE-LOCATION* with an alphanumeric identifier, such as the region where the cache is deployed. The *CACHE-LOCATION* value must not be any of the subdomains reserved for use with subdomain isolation, such as `assets` or `media`. For a list of reserved names, see "Enabling subdomain isolation."
- Set a `cache-domain` for the repository cache, replacing *EXTERNAL-CACHE-DOMAIN* with the hostname Git clients will use to access the repository cache. If you do not specify a `cache-domain`, GitHub Enterprise Server will prepend the *CACHE-LOCATION* value as a subdomain to the hostname configured for your instance. For more information, see "Configuring the hostname for your instance."
- If you haven't already, set the datacenter name on the primary and any replica appliances, replacing DC-NAME with a datacenter name.

```
ghe-repl-node --datacenter DC-NAME
```

- New caches will attempt to seed from another cache in the same datacenter. Set a `datacenter` for the repository cache, replacing REPLICA-DC-NAME with the name of the datacenter where you're deploying the node.

```
ghe-repl-node --cache CACHE-LOCATION --cache-domain EXTERNAL-CACHE-DOMAIN --
```

```
datacenter REPLICA-DC-NAME
```

11 To start replication of the datastores, use the `ghe-repl-start` command.

```
ghe-repl-start
```

> **Warning:** `ghe-repl-start` causes a brief outage on the primary server, during which users may see internal server errors. To provide a friendlier message, run `ghe-maintenance -s` on the primary node before running `ghe-repl-start` on the replica node to put the appliance in maintenance mode. Once replication starts, disable maintenance mode with `ghe-maintenance -u`. Git replication will not progress while the primary node is in maintenance mode.

12 To verify the status of each datastore's replication channel, use the `ghe-repl-status` command.

```
ghe-repl-status
```

13 To enable replication of repository networks to the repository cache, set a data location policy. For more information, see "Data location policies."

Data location policies ⚘

You can control data locality by configuring data location policies for your repositories with the `spokesctl cache-policy` command. Data location policies determine which repository networks are replicated on which repository caches. By default, no repository networks will be replicated on any repository caches until a data location policy is configured.

Data location policies affect only Git content. Content in the database, such as issues and pull request comments, will be replicated to all nodes regardless of policy.

> **Note:** Data location policies are not the same as access control. You must use repository roles to control which users may access a repository. For more information about repository roles, see "Repository roles for an organization."

You can configure a policy to replicate all networks with the `--default` flag. For example, this command will create a policy to replicate a single copy of every repository network to the set of repository caches whose `cache_location` is "kansas".

```
ghe-spokesctl cache-policy set --default 1 kansas
```

To configure replication for a repository network, specify the repository that is the root of the network. A repository network includes a repository and all of the repository's forks. You cannot replicate part of a network without replicating the whole network.

```
ghe-spokesctl cache-policy set <owner/repository> 1 kansas
```

You can override a policy that replicates all networks and exclude specific networks by specifying a replica count of zero for the network. For example, this command specifies that any repository cache in location "kansas" cannot contain any copies of that network.

```
ghe-spokesctl cache-policy set <owner/repository> 0 kansas
```

Replica counts greater than one in a given cache location are not supported.