

Monitoring a high-availability configuration

In this article

About observability for high availability

Monitoring replication from your instance

Monitoring replication using the REST API

Monitoring replication from a remote system

Troubleshooting replication issues

After configuration of high availability for your GitHub Enterprise Server instance, you can monitor the status of data replication among to your instance's replica nodes.

Who can use this feature

Site administrators can monitor a high-availability configuration for a GitHub Enterprise Server instance.

About observability for high availability

To support your plan for disaster recovery and supplement your backups, or to improve network and write performance for geographically distributed users, you can configure high availability for your GitHub Enterprise Server instance. For more information, see "[About high availability configuration](#)."

After you configure high availability, you can proactively ensure redundancy by monitoring the overall health of replication and the status of each of your instance's replica nodes. You can use command-line utilities on the instance, an overview dashboard, the instance's REST API, or a remote monitoring system such as Nagios.

With high availability, your instance uses several approaches to replicate data between primary and replica nodes. Database services that support a native replication mechanism, such as MySQL, replicate using the service's native mechanism. Other services, such as Git repositories, replicate using a custom mechanism developed for GitHub Enterprise Server, or using platform tools like rsync.

Monitoring replication from your instance

To monitor the replication status of an existing replica node for your GitHub Enterprise Server instance, connect to the node's administrative console (SSH) and run the `ghe-repl-status` command-line utility. For more information, see "[Command-line utilities](#)."

You can also monitor replication status from the overview dashboard on your instance. In a browser, navigate to the following URL, replacing HOSTNAME with your instance's hostname.

```
http(s)://HOSTNAME/setup/replication
```

Monitoring replication using the REST API

You can monitor replication status on your instance using the REST API. For more information, see "[Manage GitHub Enterprise Server](#)" in the REST API documentation.

Monitoring replication from a remote system

Output from the `ghe-repl-status` command-line utility conforms to the expectations of Nagios' `check_by_ssh` plugin. For more information, see "[Command-line utilities](#)."

Additionally, you can monitor the availability of your instance by parsing the status code returned by a request to the following URL. For example, if you deploy a load balancer as part of your failover strategy, you can configure health checks that parse this output. For more information, see "[Using GitHub Enterprise Server with a load balancer](#)."

Depending on where and how you configure monitoring, replace `HOST` with either your instance's hostname or an individual node's IP address.

```
http(s)://HOST/status
```

An active node for geo-replication, which can respond to user requests, will return status code `200` (OK). Requests to individual nodes or the instance's hostname may return a `503` (Service Unavailable) error for the following reasons.

- The individual node is a passive replica node, such as the replica node in a two-node high-availability configuration.
- The individual node is part of a geo-replication configuration, but is a passive replica node.
- The instance is in maintenance mode. For more information, see "[Enabling and scheduling maintenance mode](#)."

For more information about geo-replication, see "[About geo-replication](#)."

Troubleshooting replication issues

To troubleshoot replication issues on your instance, ensure replication is running and that nodes can communicate with each other over the network. You can also use command-line utilities to investigate under-replication.

Replication is not running

You must start replication on each node using the `ghe-repl-start` command-line utility. If replication is not running, connect to the affected node using SSH, then run `ghe-repl-start`. For more information, see "[Command-line utilities](#)."

Communication issues between nodes

Replication requires that the primary node and all replica nodes can communicate with each other over the network. At minimum, ensure that ports 122/TCP and 1194/UDP are open for bidirectional communication between all of your instance's nodes. For more information, see "[Network ports](#)."

For high availability, the latency between the network with the active nodes and the network with the replica nodes must be less than 70 milliseconds. We don't recommend configuring a firewall between the two networks. You can use `ping` or another network administration utility to test the network connectivity between nodes.

Under-replication

If you run the `ghe-repl-status` command-line utility on a replica node and Git repositories, repository networks, or storage objects are under-replicated, one or more replica nodes are not fully synchronized with the primary node. Under-replication may occur if the primary node is unable to communicate with the replica nodes, or if the replica nodes are unable to communicate with the primary node.

If you've recently configured high availability or geo-replication, the initial sync will take some time. The duration of the initial sync depends on how much data exists and network conditions.

- [Under-replicated repositories or repository networks](#)
- [Under-replicated storage objects](#)

Under-replicated repositories or repository networks

You can view a specific repository's replication status by connecting to a node and running the following commands, replacing OWNER with the repository's owner and REPOSITORY with the repository's name.

```
ghe-spokesctl check OWNER/REPOSITORY
ghe-spokesctl info OWNER/REPOSITORY
```

Alternatively, if you want to view a repository network's replication status, replace NETWORK-ID/REPOSITORY-ID with the network ID and repository ID number.

```
ghe-spokesctl check NETWORK-ID/REPOSITORY-ID
ghe-spokesctl info NETWORK-ID/REPOSITORY-ID
```

Under-replicated storage objects

You can view a specific storage object's status by connecting to a node and running the following command, replacing OID with the object's ID.

```
ghe-storage info OID
```

Getting support from GitHub

If you review the troubleshooting advice for replication and continue to experience issues on your instance, collect the following information, then contact us by visiting [GitHub Enterprise Support](#).

- On each affected node, run `ghe-repl-status -vv`, then copy the output to your ticket. For more information, see "[Command-line utilities](#)."
- On each affected node, create a support bundle to attach to your ticket. For more information, see "[Providing data to GitHub Support](#)."

Legal