# Using GitHub Enterprise Server with a load balancer

**Increase the fault tolerance of your instance**
5 of 5 in learning path

**More guides →**

**In this article**

Use a load balancer in front of a single GitHub Enterprise Server instance or a pair of instances in a High Availability configuration.

## About load balancers

A load balancer design uses a network device to direct Git and HTTP traffic to individual GitHub Enterprise Server appliances. You can use a load balancer to restrict direct traffic to the appliance for security purposes or to redirect traffic if needed without DNS record changes. We strongly recommend using a TCP-based load balancer that supports the PROXY protocol.

DNS lookups for the GitHub Enterprise Server hostname should resolve to the load balancer. We recommend that you enable subdomain isolation. If subdomain isolation is enabled, an additional wildcard record ( `*.HOSTNAME` ) should also resolve to the load balancer. For more information, see "Enabling subdomain isolation."

## Handling client connection information

Because client connections to GitHub Enterprise Server come from the load balancer, the client IP address can be lost.

If your load balancer can support it, we strongly recommend implementing the PROXY protocol. When no PROXY support is available, it is also possible to load balance the HTTP and HTTPS ports using the `X-Forwarded-For` header.

**Security Warning**: When either PROXY support or HTTP forwarding is enabled, it is critical that no external traffic can directly reach the GitHub Enterprise Server appliances. If external traffic is not properly blocked, the source IP addresses can be forged.

**Warning:** When terminating HTTPS connections on a load balancer, the requests from the load balancer to GitHub Enterprise Server also need to use HTTPS. Downgrading the connection to HTTP is not supported.

## Enabling PROXY protocol support on your GitHub Enterprise Server instance

We strongly recommend enabling PROXY protocol support for both your instance and the load balancer. Use the instructions provided by your vendor to enable the PROXY protocol on your load balancer. For more information, see the PROXY protocol documentation.

> **Note:** GitHub Enterprise Server supports PROXY Protocol V1, which is incompatible with AWS Network Load Balancers. If you use AWS Network Load Balancers with GitHub Enterprise Server, do not enable PROXY support.

1. From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click 🚀.
2. If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
3. In the "🚀 Site admin" sidebar, click **Management Console**.
4. In the "Settings" sidebar, click **Privacy**.
5. Under "External load balancers", select **Enable support for PROXY protocol**.
6. Under the "Settings" sidebar, click **Save settings**.

   > **Note:** Saving settings in the Management Console restarts system services, which could result in user-visible downtime.

7. Wait for the configuration run to complete.

## PROXY protocol TCP port mappings

| Source port | Destination port | Service description |
|---|---|---|
| 22 | 23 | Git over SSH |
| 80 | 81 | HTTP |
| 443 | 444 | HTTPS |
| 8080 | 8081 | Management Console HTTP |
| 8443 | 8444 | Management Console HTTPS |
| 9418 | 9419 | Git |

## Enabling X-Forwarded-For support on your GitHub Enterprise Server instance

Use the X-Forwarded-For protocol **only** when the PROXY protocol is unavailable. The `X-Forwarded-For` header only works with HTTP and HTTPS. The IP address reported for Git connections over SSH will show the load balancer IP.

**Warning**: If you configure `X-Forwarded-For` support on your GitHub Enterprise Server instance and load balancer, you may not be able to connect to the Management Console. For more information, see "Using GitHub Enterprise Server with a load balancer."

1. From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click 🚀.

2. If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.

3. In the "🚀 Site admin" sidebar, click **Management Console**.

4. In the "Settings" sidebar, click **Privacy**.

5. Under **External load balancers**, select **Allow HTTP X-Forwarded-For header**.

6. Under the "Settings" sidebar, click **Save settings**.

   **Note:** Saving settings in the Management Console restarts system services, which could result in user-visible downtime.

7. Wait for the configuration run to complete.

### Protocol TCP port mappings for use without PROXY support

| Source port | Destination port | Service description |
|---|---|---|
| 22 | 22 | Git over SSH |
| 25 | 25 | SMTP |
| 80 | 80 | HTTP |
| 443 | 443 | HTTPS |
| 8080 | 8080 | Management Console HTTP |
| 8443 | 8443 | Management Console HTTPS |

## Configuring health checks

Health checks allow a load balancer to stop sending traffic to a node that is not responding if a pre-configured check fails on that node. If the instance is offline due to maintenance or unexpected failure, the load balancer can display a status page. In a High Availability (HA) configuration, a load balancer can be used as part of a failover strategy. However, automatic failover of HA pairs is not supported. You must manually promote the replica instance before it will begin serving requests. For more information, see "Configuring high availability."

Configure the load balancer to check the following URL.

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

The endpoint will return status code `200` (OK) if the node is healthy and available to service end-user requests. For more information, see "Monitoring a high-availability configuration."

**Note:** When the appliance is in maintenance mode, the `https://HOSTNAME/status` URL will return status code `503` (Service Unavailable). For more information, see "Enabling and scheduling maintenance mode."

## Troubleshooting connectivity through a load balancer

If you cannot connect to services on your GitHub Enterprise Server instance through a load balancer, you can review the following information to troubleshoot the problem.

**Note**: Always test changes to your network infrastructure and instance configuration in a staging environment. For more information, see "Setting up a staging instance."

### Error: "Your session has expired" for connections to the Management Console

If you enable support for the `X-Forwarded-For` header on your instance and load balancer, you may not be able to access your instance's Management Console. For more information about the Management Console and ports required for connections, see "Administering your instance from the web UI" and "Network ports."

If your GitHub Enterprise Server instance indicates that your session has expired when you connect to the Management Console through a load balancer, try one of the following configurations on your load balancer.

- Disable `X-Forwarded-For` headers for connections to your instance on ports 8080 and 8443.
- Configure your load balancer to operate on Layer 4, and use the PROXY protocol instead of `X-Forwarded-For` for passthrough of client IP addresses. For more information, see "Enabling PROXY protocol support on your GitHub Enterprise Server instance."

For more information, refer to the documentation for your load balancer.

### Live updates to issues and check runs not working

When your GitHub Enterprise Server instance is accessed via a load balancer or reverse proxy, expected live updates, such as new comments on issues and changes in notification badges or check run output, may not display until the page is refreshed. This is most common when the reverse proxy or load balancer is running in a layer 7 mode or does not support the required websocket protocol.

To enable live updates, you may need to reconfigure the load balancer or proxy. For more information, refer to the documentation for your load balancer.

Previous
**Creating a high availability replica**