

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](#).

Replacing a cluster node

In this article

About replacement of GitHub Enterprise Server cluster nodes

Replacing a functional node

Replacing a node in an emergency

If a node fails in a GitHub Enterprise Server cluster, or if you want to add a new node with more resources, mark any nodes to replace as offline, then add the new node.

GitHub determines eligibility for clustering, and must enable the configuration for your instance's license. Clustering requires careful planning and additional administrative overhead. For more information, see "[About clustering](#)."

About replacement of GitHub Enterprise Server cluster nodes

You can replace a functional node in a GitHub Enterprise Server cluster, or you can replace a node that has failed unexpectedly.

Warning: To avoid conflicts, do not reuse a hostname that was previously assigned to a node in the cluster.

Replacing a functional node

You can replace an existing, functional node in your cluster. For example, you may want to provide a virtual machine (VM) with additional CPU, memory, or storage resources.

To replace a functional node, install the GitHub Enterprise Server appliance on a new VM, configure an IP address, add the new node to the cluster configuration file, initialize the cluster and apply the configuration, then take the node you replaced offline.

- 1 [Provision and install GitHub Enterprise Server](#) with a unique hostname on the replacement node.
- 2 Using the administrative shell or DHCP, **only** configure the IP address of the replacement node. Don't configure any other settings.
- 3 To add the newly provisioned replacement node, on any node, modify the `cluster.conf` file to remove the failed node and add the replacement node. For example, this modified `cluster.conf` file replaces `ghe-data-node-3` with the newly

provisioned node, `ghe-replacement-data-node-3` :

```
[cluster "ghe-replacement-data-node-3"]
hostname = ghe-replacement-data-node-3
ipv4 = 192.168.0.7
# ipv6 = fd12:3456:789a:1::7
git-server = true
pages-server = true
mysql-server = true
elasticsearch-server = true
redis-server = true
memcache-server = true
metrics-server = true
storage-server = true
```

- 4 From the administrative shell of the node with the modified `cluster.conf` , run `ghe-cluster-config-init` . This will initialize the newly added node in the cluster.
- 5 From the same node, run `ghe-cluster-config-apply` . This will validate the configuration file, copy it to each node in the cluster, and configure each node according to the modified `cluster.conf` file.
- 6 If you're taking a node offline that provides data services, such as `git-server` , `pages-server` , or `storage-server` , evacuate the node. For more information, see "[Evacuating a cluster node running data services](#)."
- 7 To mark the failed node offline, on any node, modify the [cluster configuration file](#) (`cluster.conf`) in the relevant node section to include the text `offline = true` .

For example, this modified `cluster.conf` will mark the `ghe-data-node-3` node as offline:

```
[cluster "ghe-data-node-3"]
hostname = ghe-data-node-3
offline = true
ipv4 = 192.168.0.6
# ipv6 = fd12:3456:789a:1::6
```

- 8 From the administrative shell of the node where you modified `cluster.conf` , run `ghe-cluster-config-apply` . This will validate the configuration file, copy it to each node in the cluster, and mark the node offline.
- 9 If you're replacing the primary MySQL or Redis node, in `cluster.conf` , modify the `mysql-master` or `redis-master` value with the replacement node name.

For example, this modified `cluster.conf` file specifies a newly provisioned cluster node, `ghe-replacement-data-node-1` as the primary MySQL and Redis node:

```
mysql-master = ghe-replacement-data-node-1
redis-master = ghe-replacement-data-node-1
```

Replacing a node in an emergency

You can replace a failed node in your cluster. For example, a software or hardware issue may affect a node's availability.

To replace a node in an emergency, install the GitHub Enterprise Server appliance on a

new VM, configure an IP address, take the failed node offline, apply the configuration, add the new node to the cluster configuration file, initialize the cluster and apply the configuration, and optionally, evacuate the failed node.

- 1 [Provision and install GitHub Enterprise Server](#) with a unique hostname on the replacement node.
- 2 Using the administrative shell or DHCP, **only** configure the IP address of the replacement node. Don't configure any other settings.
- 3 To mark the failed node offline, on any node, modify the [cluster configuration file](#) (`cluster.conf`) in the relevant node section to include the text `offline = true` .

For example, this modified `cluster.conf` will mark the `ghe-data-node-3` node as offline:

```
[cluster "ghe-data-node-3"]
hostname = ghe-data-node-3
offline = true
ipv4 = 192.168.0.6
# ipv6 = fd12:3456:789a:1::6
```

- 4 From the administrative shell of the node where you modified `cluster.conf` , run `ghe-cluster-config-apply` . This will validate the configuration file, copy it to each node in the cluster, and mark the node offline.
- 5 To add the newly provisioned replacement node, on any node, modify the `cluster.conf` file to remove the failed node and add the replacement node. For example, this modified `cluster.conf` file replaces `ghe-data-node-3` with the newly provisioned node, `ghe-replacement-data-node-3` :

```
[cluster "ghe-replacement-data-node-3"]
hostname = ghe-replacement-data-node-3
ipv4 = 192.168.0.7
# ipv6 = fd12:3456:789a:1::7
git-server = true
pages-server = true
mysql-server = true
elasticsearch-server = true
redis-server = true
memcache-server = true
metrics-server = true
storage-server = true
```

- 6 If you're replacing the primary MySQL or Redis node, in `cluster.conf` , modify the `mysql-master` or `redis-master` value with the replacement node name.

For example, this modified `cluster.conf` file specifies a newly provisioned cluster node, `ghe-replacement-data-node-1` as the primary MySQL and Redis node:

```
mysql-master = ghe-replacement-data-node-1
redis-master = ghe-replacement-data-node-1
```

- 7 From the administrative shell of the node with the modified `cluster.conf` , run `ghe-cluster-config-init` . This will initialize the newly added node in the cluster.
- 8 From the same node, run `ghe-cluster-config-apply` . This will validate the configuration file, copy it to each node in the cluster, and configure each node according to the modified `cluster.conf` file.

- 9 If you're taking a node offline that provides data services, such as `git-server`, `pages-server`, or `storage-server`, evacuate the node. For more information, see "[Evacuating a cluster node running data services](#)."

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)