



Monitoring the health of your cluster nodes with Node Eligibility Service

In this article

About Node Eligibility Service

About health and eligibility of cluster nodes

Enabling Node Eligibility Service for your cluster

Configuring TTL settings for Node Eligibility Service

Managing whether Node Eligibility Service can take a node offline

Viewing an overview of node health

Re-enabling an ineligible node to join the cluster

Viewing logs for Node Eligibility Service

Further reading

You can monitor when nodes in a GitHub Enterprise Server cluster have been offline long enough to cause issues by using Node Eligibility Service.

Who can use this feature

People with administrative SSH access to a GitHub Enterprise Server instance can monitor cluster nodes.

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 Node Eligibility Service &

In a GitHub Enterprise Server cluster, an individual node may become unreachable by other nodes due to a hardware or software failure. After time, even if you restore the node's health, the subsequent synchronization of data can negatively impact your instance's performance.

You can proactively mitigate the impact of reduced node availability by using Node Eligibility Service. This service monitors the state of your cluster's nodes and emits a warning if a node has been offline for too long. You can also prevent an offline node from rejoining the cluster. Optionally, you can allow Node Eligibility Service to take ineligible nodes offline.

By default, Node Eligibility Service is disabled. If you enable Node Eligibility Service, your instance will alert you of unhealthy nodes by displaying a banner in the administrative web UI for GitHub Enterprise Server, and in CLI output for some cluster-related utilities, such as ghe-config-apply and ghe-cluster-diagnostics.

Node Eligibility Service allows you to monitor the health of individual nodes. You can also monitor the overall health of your cluster. For more information, see "Monitoring the health of your cluster."

About health and eligibility of cluster nodes &

To determine whether to emit a warning or automatically adjust the configuration of your cluster, Node Eligibility Service continuously monitors the health of each node. Each node regularly reports a timestamped health state, which Node Eligibility Service compares to a Time To Live (TTL) duration.

Each node has a health state and an eligibility state.

- Health refers to the accessibility of the node within the cluster and has three possible states: healthy, warning, or critical.
- Eligibility refers to the ability of the node to work in the cluster and has two possible states: eligible or ineligible.

Node Eligibility Service provides a configurable TTL setting for two states, warn and fail .

- warn: The node has been offline for a short period of time. This may indicate something is wrong with the node and that administrators should investigate. The default setting is 15 minutes.
- fail: The node has been offline for a long period of time, and reintroduction into the cluster could cause performance issues due to resynchronization. The default setting is 60 minutes.

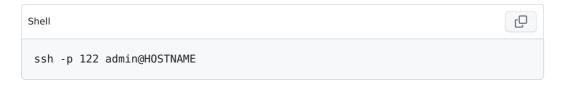
For each node, Node Eligibility Service determines health and eligibility for participation in the cluster in the following ways.

- If a node has been observed to be healthy, the health state is healthy and the eligibility state is eligible.
- If a node hasn't been observed to be healthy for longer than the warn TTL, the health state is warning and the eligibility state is eligible.
- If a node hasn't been observed to be healthy for longer than the fail TTL, the health state is critical and its eligibility state is ineligible.

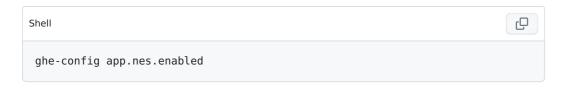
Enabling Node Eligibility Service for your cluster $\mathscr P$

By default, Node Eligibility Service is disabled. You can enable Node Eligibility Service by setting the value for app.nes.enabled using ghe-config.

To connect to your GitHub Enterprise Server instance, SSH into any of your cluster's nodes. From your workstation, run the following command. Replace HOSTNAME with the node's hostname. For more information, see "Accessing the administrative shell (SSH)."



2 To verify whether Node Eligibility Service is currently enabled, run the following command.



To enable Node Eligibility Service, run the following command. Q Shell ghe-config app.nes.enabled true To apply the configuration, run the following command. Note: During a configuration run, services on your GitHub Enterprise Server instance may restart, which can cause brief downtime for users. Shell Q ghe-config-apply Wait for the configuration run to complete. To verify that Node Eligibility Service is running, from any node, run the following command. Q Shell nomad status nes **Configuring TTL settings for Node Eligibility Service** O To determine how Node Eligibility Service notifies you, you can configure TTL settings for fail and warn states. The TTL for the fail state must be higher than the TTL for the warn state. 1 To connect to your GitHub Enterprise Server instance, SSH into any of your cluster's nodes. From your workstation, run the following command. Replace HOSTNAME with the node's hostname. For more information, see "Accessing the administrative shell (SSH)." Shell Q ssh -p 122 admin@HOSTNAME To verify the current TTL settings, run the following command. Q

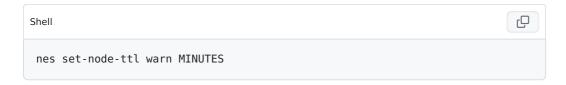
To set the TTL for the fail state, run the following command. Replace MINUTES with the number of minutes to use for failures.

Shell

nes get-node-ttl all



4 To set the TTL for the warn state, run the following command. Replace MINUTES with the number of minutes to use for warnings.



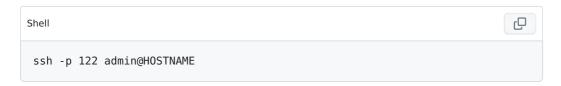
Managing whether Node Eligibility Service can take a node offline *∂*

By default, Node Eligibility Service provides alerts to notify you about changes to the health of cluster nodes. Optionally, if the service determines that an unhealthy node is ineligible to rejoin the cluster, you can allow the service to take the node offline.

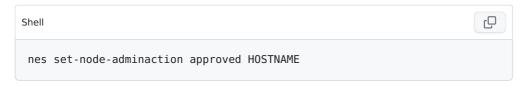
When a node is taken offline, the instance removes job allocations from the node. If the node runs data storage services, Node Eligibility Service updates the configuration to reflect the node's ineligibility to rejoin the cluster.

To manage whether Node Eligibility Service can take a node and its services offline, you can configure adminaction states for the node. If a node is in the approved state, Node Eligibility Service can take the node offline. If a node is in the none state, Node Eligibility Service cannot take the node offline.

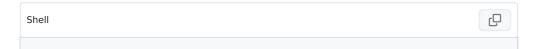
1 To connect to your GitHub Enterprise Server instance, SSH into any of your cluster's nodes. From your workstation, run the following command. Replace HOSTNAME with the node's hostname. For more information, see "Accessing the administrative shell (SSH)."



- 2 To configure whether Node Eligibility Service can take a node offline, run one of the following commands.
 - To allow the service to automatically take administrative action when a node goes offline, run the following command. Replace HOSTNAME with the node's hostname.



 To revoke Node Eligibility Service's ability to take a node offline, run the following command. Replace HOSTNAME with the node's hostname.



Viewing an overview of node health &

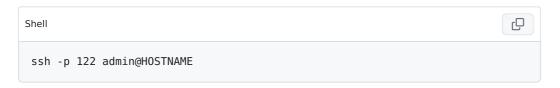
To view an overview of your nodes' health using Node Eligibility Service, use one of the following methods.

- SSH into any node in the cluster, then run nes get-cluster-health.
- Navigate to the Management Console's "Status" page. For more information, see
 "Accessing the Management Console."

Re-enabling an ineligible node to join the cluster &

After Node Eligibility Service detects that a node has exceeded the TTL for the fail state, and after the service marks the node as ineligible, the service will no longer update the health status for the node. To re-enable a node to join the cluster, you can remove the ineligible status from the node.

1 To connect to your GitHub Enterprise Server instance, SSH into any of your cluster's nodes. From your workstation, run the following command. Replace HOSTNAME with the node's hostname. For more information, see "Accessing the administrative shell (SSH)."



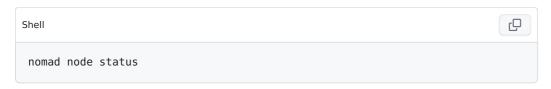
2 To check the current adminaction state for the node, run the following command. Replace HOSTNAME with the hostname of the ineligible node.



3 If the adminaction state is currently set to approved, change the state to none by running the following command. Replace HOSTNAME with the hostname of the ineligible node.



4 To ensure the node is in a healthy state, run the following command and confirm that the node's status is ready .



o If the node's status is ineligible, make the node eligible by connecting to the

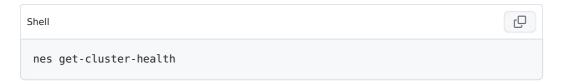
node via SSH and running the following command.



5 To update the node's eligibility in Node Eligibility Service, run the following command. Replace HOSTNAME with the node's hostname.



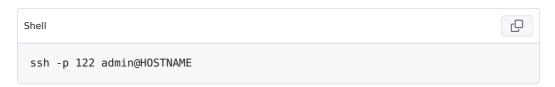
6 Wait 30 seconds, then check the cluster's health to confirm the target node is eligible by running the following command.



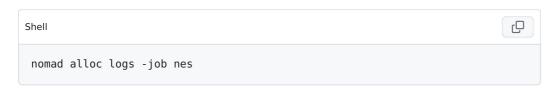
Viewing logs for Node Eligibility Service *№*

You can view logs for Node Eligibility Service from any node in the cluster, or from the node that runs the service. If you generate a support bundle, the logs are included. For more information, see "Providing data to GitHub Support."

1 To connect to your GitHub Enterprise Server instance, SSH into any of your cluster's nodes. From your workstation, run the following command. Replace HOSTNAME with the node's hostname. For more information, see "Accessing the administrative shell (SSH)."



2 To view logs for Node Eligibility Service from any node in the cluster, run the following command.

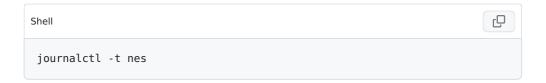


- 3 Alternatively, you can view logs for Node Eligibility Service on the node that runs the service. The service writes logs to the systemd journal.
 - To determine which node runs Node Eligibility Service, run the following command.



nomad job status "nes" | grep running | grep "\${nomad_node_id}" | awk
'NR==2{ print \$1 }' | xargs nomad alloc status | grep "Node Name"

• To view logs on the node, connect to the node via SSH, then run the following command.



Further reading @

• "Command-line utilities"

Legal

© 2023 GitHub, Inc. <u>Terms</u> <u>Privacy</u> <u>Status</u> <u>Pricing</u> <u>Expert services</u> <u>Blog</u>