



Service and maintain

BeeGFS on NetApp with E-Series Storage

NetApp

January 27, 2026

This PDF was generated from <https://docs.netapp.com/us-en/beegfs/administer/clusters-failover-failback.html> on January 27, 2026. Always check docs.netapp.com for the latest.

Table of Contents

- Service and maintain 1
 - Failover and failback services 1
 - Overview 1
 - Steps 1
 - Place the cluster in maintenance mode 3
 - Overview 3
 - Steps 4
 - Stop and start the cluster 4
 - Overview 4
 - Steps 4
 - Replace file nodes 5
 - Overview 5
 - Steps: 5
 - Expand or shrink the cluster 6
 - Overview 6
 - Adding a Building Block to the Cluster 6
 - Removing a Building Block from the Cluster 7

Service and maintain

Failover and failback services

Moving BeeGFS services between cluster nodes.

Overview

BeeGFS services can failover between nodes in the cluster to ensure clients are able to continue accessing the file system if a node experiences a fault, or you need to perform planned maintenance. This section describes various ways administrators can heal the cluster after recovering from a failure, or manually move services between nodes.

Steps

Failover and Failback

Failover (Planned)

Generally when you need to bring a single file node offline for maintenance you'll want to move (or drain) all BeeGFS services from that node. This can be accomplished by first putting the node in standby:

```
pcs node standby <HOSTNAME>
```

After verifying using `pcs status` all resources have been restarted on the alternate file node, you can shutdown or make other changes to the node as needed.

Failback (after a planned failover)

When you are ready to restore BeeGFS services to the preferred node first run `pcs status` and verify in the "Node List" the status is standby. If the node was rebooted it will show offline until you bring the cluster services online:

```
pcs cluster start <HOSTNAME>
```

Once the node is online bring it out of standby with:

```
pcs node unstandby <HOSTNAME>
```

Lastly relocate all BeeGFS services back to their preferred nodes with:

```
pcs resource relocate run
```

Failback (after an unplanned failover)

If a node experience a hardware or other fault, the HA cluster should automatically react and move its services to a healthy node, providing time for administrators take corrective action. Before proceeding reference the

[troubleshooting](#) section to determine the cause of the failover and resolve any outstanding issues. Once the node is powered back on and healthy you can proceed with failback.

When a node boots following an unplanned (or planned) reboot, cluster services are not set to start automatically, so you will first need to bring the node online with:

```
pcs cluster start <HOSTNAME>
```

Next cleanup any resource failures and reset the node's fencing history:

```
pcs resource cleanup node=<HOSTNAME>
pcs stonith history cleanup <HOSTNAME>
```

Verify in `pcs status` the node is online and healthy. By default BeeGFS services will not automatically failback to avoid accidentally moving resources back to an unhealthy node. When you are ready return all resources in the cluster back to their preferred nodes with:

```
pcs resource relocate run
```

Moving individual BeeGFS services to alternate file nodes

Permanently move a BeeGFS service to a new file node

If you want to permanently change the preferred file node for an individual BeeGFS service, adjust the Ansible inventory so the preferred node is listed first and rerun the Ansible playbook.

For example in this sample `inventory.yml` file, `beegfs_01` is the preferred file node to run the BeeGFS management service:

```
mgmt:
  hosts:
    beegfs_01:
    beegfs_02:
```

Reversing the order would cause the management services to be preferred on `beegfs_02`:

```
mgmt:
  hosts:
    beegfs_02:
    beegfs_01:
```

Temporarily move a BeeGFS service to an alternate file node

Generally if a node is undergoing maintenance you will want to use the [failover and failback

steps](#failover-and-failback) to move all services away from that node.

If for some reason you do need to move an individual service to a different file node run:

```
pcs resource move <SERVICE>-monitor <HOSTNAME>
```



Do not specify individual resources or the resource group. Always specify the name of the monitor for the BeeGFS service you wish to relocate. For example to move the BeeGFS management service to beegfs_02 run: `pcs resource move mgmt-monitor beegfs_02`. This process can be repeated to move one or more services away from their preferred nodes. Verify using `pcs status` the services were relocated/started on the new node.

To move a BeeGFS service back to its preferred node first clear the temporary resource constraints (repeating this step as needed for multiple services):

```
pcs resource clear <SERVICE>-monitor
```

Then when ready to actually move service(s) back to their preferred node(s) run:

```
pcs resource relocate run
```

Note this command will relocate any services that no longer have temporary resource constraints not located on their preferred nodes.

Place the cluster in maintenance mode

Prevent the HA cluster from accidentally reacting to intended changes in the environment.

Overview

Putting the cluster in maintenance mode disables all resource monitoring and prevents Pacemaker from moving or otherwise managing resources in the cluster. All resources will remain running on their original nodes, regardless if there is a temporary failure condition that would prevent them from being accessible. Scenarios where this is recommended/useful include:

- Network maintenance that may temporarily disrupt connections between file nodes and BeeGFS services.
- Block Node upgrades.
- File Node operating system, kernel, or other package updates.

Generally the only reason to manually put the cluster in maintenance mode is to prevent it from reacting to external changes in the environment. If an individual node in the cluster requires physical repair do not use maintenance mode and simply place that node in standby following the procedure above. Note that rerunning Ansible will automatically put the cluster in maintenance mode facilitating most software maintenance including upgrades and configuration changes.

Steps

To check if the cluster is in maintenance mode run:

```
pcs property config
```

The `maintenance-mode` property will not appear if the cluster is operating normally. If the cluster is currently in maintenance mode, the property will report as `true`. To enable maintenance mode run:

```
pcs property set maintenance-mode=true
```

You can verify by running `pcs status` and ensuring all resources show "(unmanaged)". To take the cluster out of maintenance mode run:

```
pcs property set maintenance-mode=false
```

Stop and start the cluster

Gracefully stopping and starting the HA cluster.

Overview

This section describes how to gracefully shutdown and restart the BeeGFS cluster. Example scenarios where this may be required include electrical maintenance or migrating between datacenters or racks.

Steps

If for any reason you need to stop the entire BeeGFS cluster and shutdown all services run:

```
pcs cluster stop --all
```

It is also possible to stop the cluster on individual nodes (which will automatically failover services to another node), though it is recommended to first put the node in standby (see the [failover](#) section):

```
pcs cluster stop <HOSTNAME>
```

To start cluster services and resources on all nodes run:

```
pcs cluster start --all
```

Or start services on a specific node with:

```
pcs cluster start <HOSTNAME>
```

At this point run `pcs status` and verify the cluster and BeeGFS services start on all nodes, and services are running on the nodes you expect.



Depending on cluster size, it may take seconds or minutes for the entire cluster to stop, or show as started in `pcs status`. If `pcs cluster <COMMAND>` hangs for more than five minutes, before running "Ctrl+C" to cancel the command, login to each node of the cluster and use `pcs status` to see if cluster services (Corosync/Pacemaker) are still running on that node. From any node where the cluster is still active you can check what resources are blocking the cluster. Manually address the issue and the command should either complete or can be rerun to stop any remaining services.

Replace file nodes

Replacing a file node if the original server is faulty.

Overview

This is an overview of the steps needed to replace a file node in the cluster. These steps presume the file node failed due to a hardware issue, and was replaced with a new identical file node.

Steps:

1. Physically replace the file node and restore all cabling to the block node and storage network.
2. Reinstall the operating system on the file node including adding Red Hat subscriptions.
3. Configure management and BMC networking on the file node.
4. Update the Ansible inventory if the hostname, IP, PCIe-to-logical interface mappings, or anything else changed about the new file node. Generally this is not needed if the node was replaced with identical server hardware and you are using the original network configuration.
 - a. For example if the hostname changed, create (or rename) the node's inventory file (`host_vars/<NEW_NODE>.yaml`) then in the Ansible inventory file (`inventory.yaml`), replace the old node's name with the new node name:

```
all:
  ...
  children:
    ha_cluster:
      children:
        mgmt:
          hosts:
            node_h1_new:  # Replaced "node_h1" with "node_h1_new"
            node_h2:
```

5. From one of the other nodes in the cluster, remove the old node: `pcs cluster node remove`

<HOSTNAME>.



DO NOT PROCEED BEFORE RUNNING THIS STEP.

6. On the Ansible control node:

a. Remove the old SSH key with:

```
`ssh-keygen -R <HOSTNAME_OR_IP>`
```

b. Configure passwordless SSH to the replace node with:

```
ssh-copy-id <USER>@<HOSTNAME_OR_IP>
```

7. Rerun the Ansible playbook to configure the node and add it to the cluster:

```
ansible-playbook -i <inventory>.yaml <playbook>.yaml
```

8. At this point, run `pcs status` and verify the replaced node is now listed and running services.

Expand or shrink the cluster

Add or remove building blocks from the cluster.

Overview

This section documents various considerations and options to adjust the size of your BeeGFS HA cluster. Typically cluster size is adjusted by adding or removing building blocks, which are typically two file nodes setup as an HA pair. It is also possible to add or remove individual file nodes (or other types of cluster nodes) if needed.

Adding a Building Block to the Cluster

Considerations

Growing the cluster by adding additional building blocks is a straightforward process. Before you begin keep in mind restrictions around the minimum and maximum number of cluster nodes in each individual HA cluster, and determine if you should add nodes to the existing HA cluster, or create a new HA cluster. Typically each building block consists of two file nodes, but three nodes is the minimum number of nodes per cluster (to establish quorum), and ten is the recommended (tested) maximum. For advanced scenarios it is possible to add a single "tiebreaker" node that does not run any BeeGFS services when deploying a two node cluster. Please contact NetApp support if you are considering such a deployment.

Keep in mind these restrictions and any anticipated future cluster growth when deciding how to expand the cluster. For example if you have a six node cluster and need to add four more nodes, it would be recommended to just start a new HA cluster.



Remember, a single BeeGFS file system can consist of multiple independent HA clusters. This allows file systems to continue scaling far past the recommended/hard limits of the underlying HA cluster components.

Steps

When adding a building block to your cluster, you will need to create the `host_vars` files for each of the new file nodes and block nodes (E-Series arrays). The names of these hosts need to be added to the inventory, along with the new resources that are to be created. The corresponding `group_vars` files will need to be created for each new resource. See the [use custom architectures](#) section for details.

After creating the correct files, all that is needed is to rerun the automation using the command:

```
ansible-playbook -i <inventory>.yaml <playbook>.yaml
```

Removing a Building Block from the Cluster

There are a number of considerations to keep in mind when you need to retire a building block, for example:

- What BeeGFS services are running in this building block?
- Are just the file nodes retiring and the block nodes should be attached to new file nodes?
- If the entire building block is being retired, should the data be moved to a new building block, dispersed into existing nodes in the cluster, or moved to a new BeeGFS file system or other storage system?
- Can this happen during an outage or should it be done non-disruptively?
- Is the building block actively in use, or does it primarily contain data that is no-longer active?

Because of the diverse possible starting points and desired end states, please contact NetApp support so we can identify and help implement the best strategy based on your environment and requirements.

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.