# Initial Set Up

## BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

# Table of Contents

# Initial Set Up

## Install and Cable Hardware

Steps needed to install and cable hardware used to run BeeGFS on NetApp.

### Plan the Installation

Each BeeGFS file system will consist of some number of file nodes running BeeGFS services using backend storage provided by some number of block nodes. The file nodes are configured into one or more high availability clusters to provide fault tolerance for BeeGFS services. Each block node is a already an active/active HA pair. The minimum number of supported file nodes in each HA cluster is three, and the maximum number of supported file nodes in each cluster is ten. BeeGFS file systems can scale beyond ten node by deploying multiple independent HA clusters that work together to provide a single file system namespace.

Commonly each HA cluster is deployed as a series of "building blocks" where some number of file nodes (x86 servers) are directly connected to some number of block nodes (typically E-Series storage systems). This configuration creates an asymmetrical cluster, where BeeGFS services are only able to run on certain file nodes that have access to the backend block storage used for the BeeGFS targets. The balance of file-to-block nodes in each building block and the storage protocol in use for the direct-connects depend on the requirements of a particular installation.

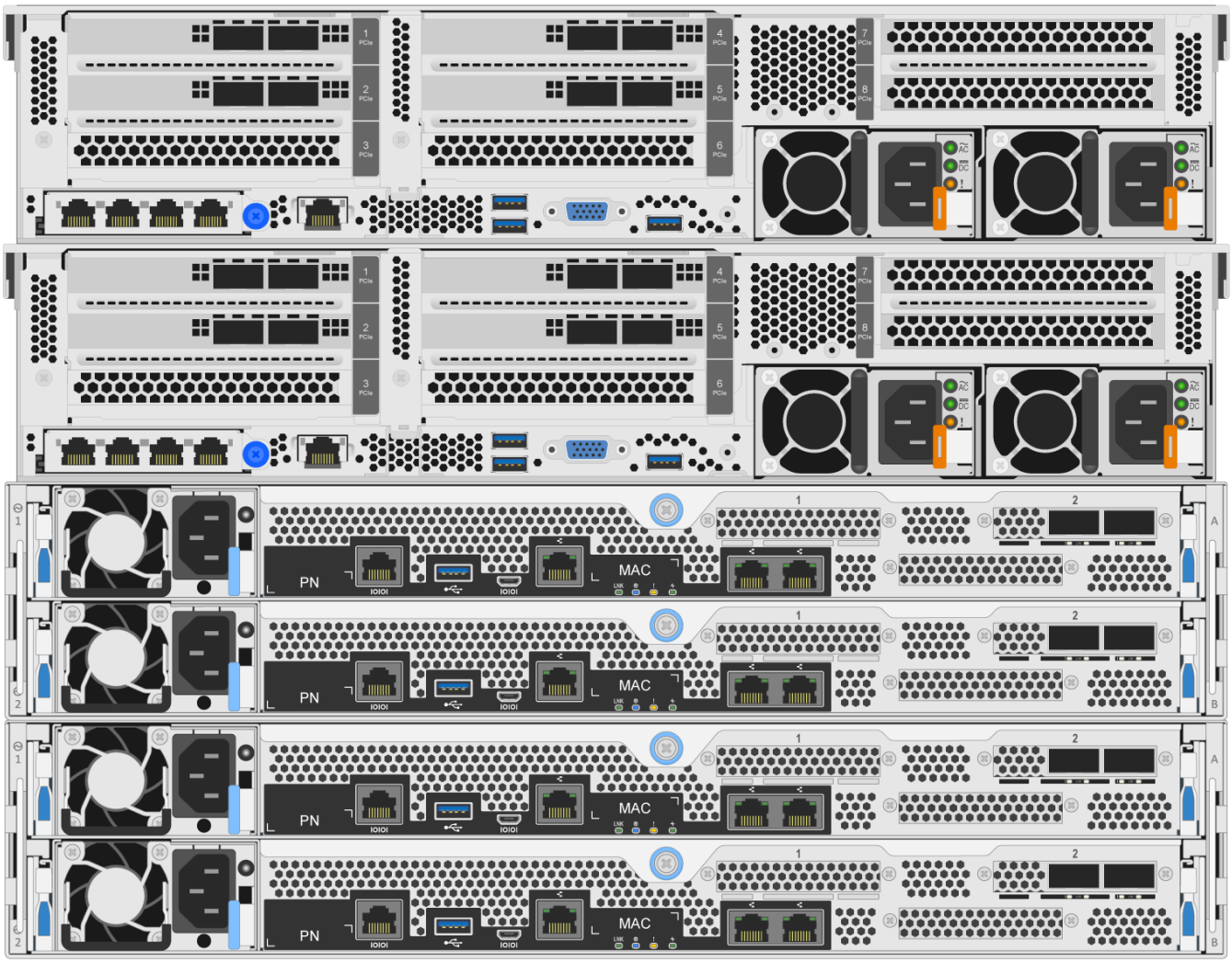
An alternative HA cluster architecture uses a storage fabric (also known as a storage area network or SAN) between the file and block nodes to establish a symmetrical cluster. This allows BeeGFS services to run on any file node in a particular HA cluster. As generally symmetrical clusters are not as cost effective due to the extra SAN hardware, this documentation presumes use of an asymmetrical cluster deployed as a series of one or more building blocks.

Ensure the desired file system architecture for a particular BeeGFS deployment is well understood before proceeding with the installation.

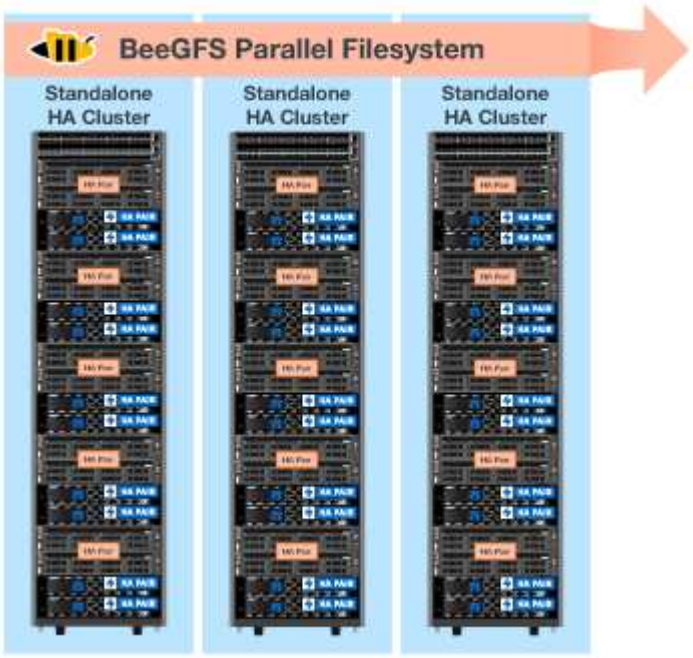
### Rack Hardware

When planning the installation it is important all equipment in each building block is racked in adjacent rack units. Best practice is for file nodes to be racked immediately above block nodes in each building block. Follow the documentation for the model(s) of file and block nodes you are using as you install rails and hardware into the rack.

Example of a single building block:

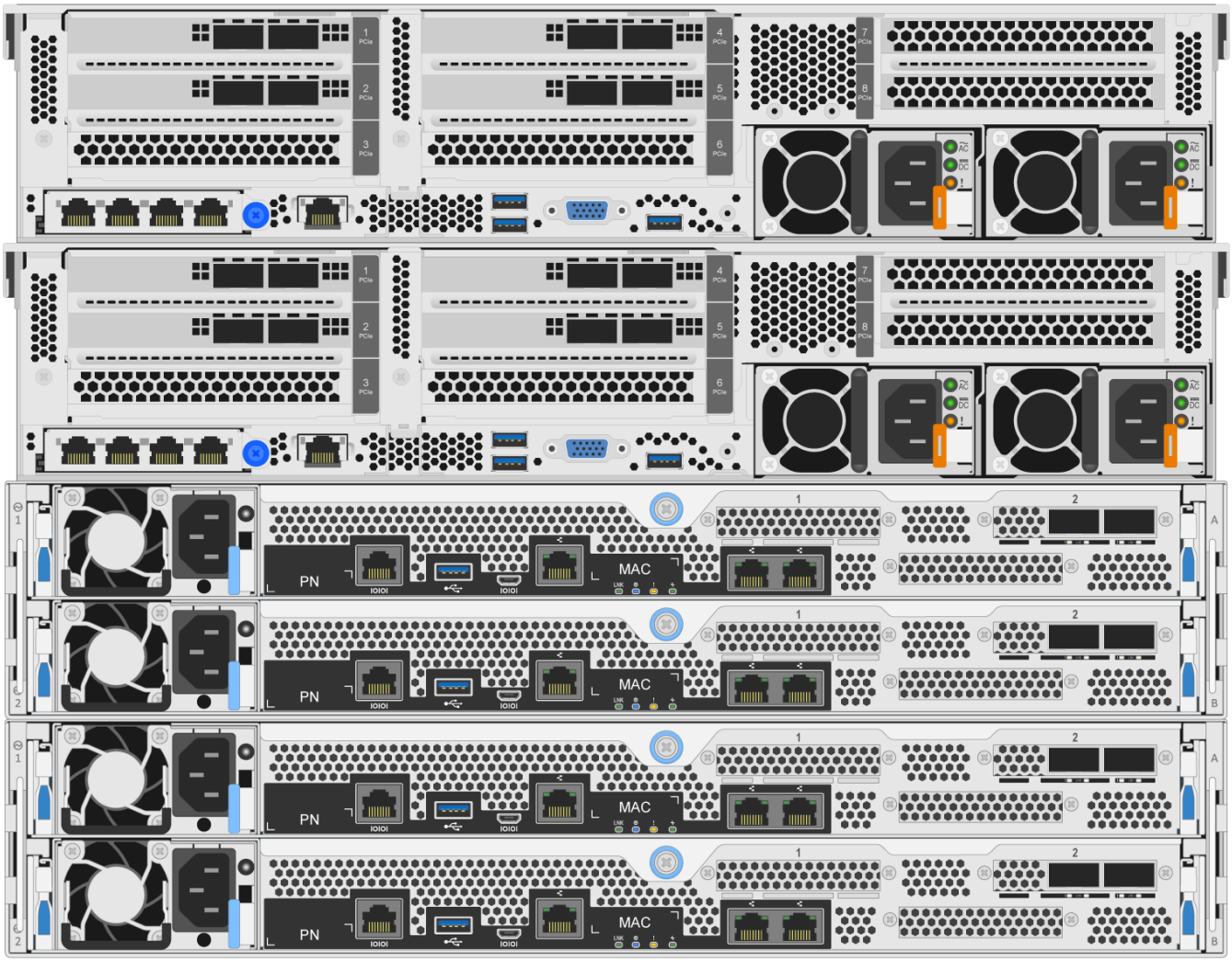



Example of a large BeeGFS installation where there are multiple building blocks in each HA cluster, and multiple HA clusters in the file system:



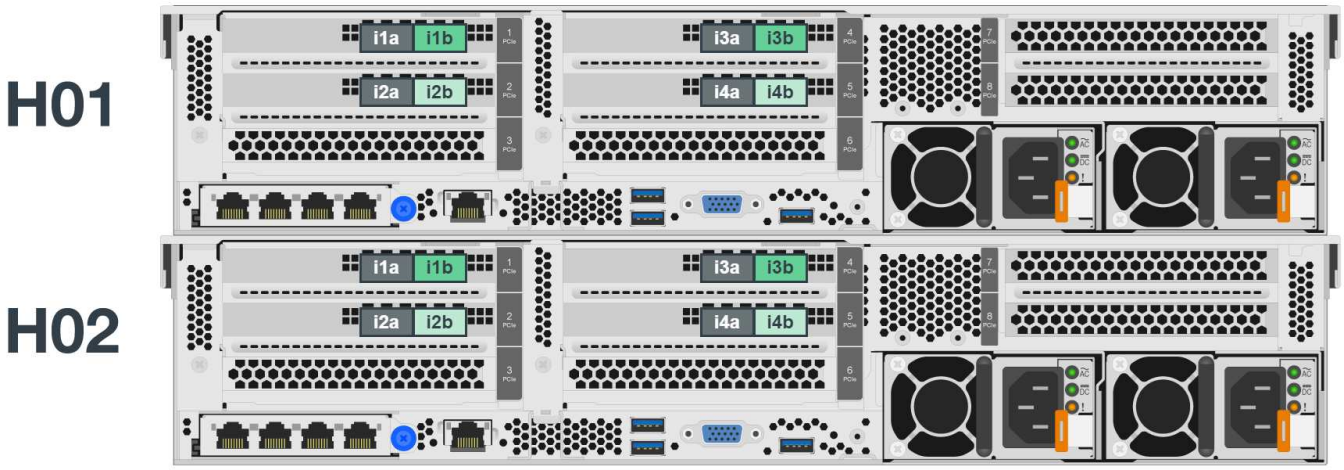
## Cable File and Block Nodes

Typically you will direct-connect the HIC ports of the E-Series block nodes to the designated host channel adapter (for InfiniBand protocols) or host bus adapter (for fibre channel and other protocols) ports of the file nodes. The exact way to establish these connections will depend on the desired file system architecture, here is an example based on the second-generation BeeGFS on NetApp verified architecture:




## Cable File Nodes to the Client Network

Each file node will have some number of InfiniBand or Ethernet ports designated for BeeGFS client traffic. Depending on the architecture each file node will have one or more connections to a high performance client/storage network, potentially to multiple switches for redundancy and increased bandwidth. Here is an example of client cabling using redundant network switches, where ports highlighted in dark green versus light green are connected to separate switches:




### Connect Management Networking and Power

Establish any network connections needed for in-band and out-of-band network.

Connect all power supplies ensuring each file and block node has connections to multiple power distribution units for redundancy (if available).

# Set Up File and Block Nodes

Manual steps required to set up file and block nodes before running Ansible.

## File Nodes

#### Configure the Baseboard Management Controller (BMC)

A baseboard management controller (BMC), sometimes referred to as a service processor, is the generic name for the out-of-band management capability built into various server platforms that can provide remote access even if the operating system is not installed or accessible. Vendors typically market this functionality with their own branding. For example, on the Lenovo SR665, the BMC is referred to as the Lenovo XClarity Controller (XCC).

Follow the server vendor's documentation to enable any licenses needed to access this functionality and ensure the BMC is connected to the network and configured appropriately for remote access.

> ⓘ If BMC based fencing using Redfish is desired, ensure Redfish is enabled and the BMC interface is accessible from the OS installed on the file node. Special configuration may be required on the network switch if the BMC and operating share the same physical network interface.

#### Tune System Settings

Using the system setup (BIOS/UEFI) interface, ensure settings are set to maximize performance. The exact settings and optimal values will vary based on the server model in use. Guidance is provided for verified file node models, otherwise refer to the server vendor's documentation and best practices based on your model.

#### Install an Operating System

Install a supported operating system based on the file node requirements listed here. Refer to any additional steps below based on your Linux distribution.

##### Red Hat

Use Red Hat Subscription Manager to register and subscribe the system to allow installation of the required packages from the official Red Hat repositories and to limit updates to the supported version of Red Hat: `subscription-manager release --set=<MAJOR_VERSION>.<MINOR_VERSION>`. For instructions, see How to register and subscribe a RHEL system and How to limit updates.

Enable the Red Hat repository containing the packages required for high availability:

```
subscription-manager repo-override --repo=rhel-9-for-x86_64
-highavailability-rpms --add=enabled:1
```

#### Configure Management Network

Configure any network interfaces needed to allow in-band management of the operating system. The exact steps will depend on the specific Linux distribution and version in use.

Ensure SSH is enabled and all management interfaces are accessible from the Ansible control node.

#### Update HCA and HBA Firmware

Ensure all HBAs and HCAs are running supported firmware versions listed on the NetApp Interoperability Matrix and upgrade if necessary. Additional recommendations for NVIDIA ConnectX adapters can be found here.

### Block Nodes

Follow the steps to get up and running with E-Series to configure the management port on each block node controller and optionally set the storage array name for each system.

No additional configuration beyond ensuring all block nodes are accessible from the Ansible control node is necessary. The remaining system configuration will be applied/maintained using Ansible.

# Set Up Ansible Control Node

Set up an Ansible control node to deploy and manage the file system.

## Overview

An Ansible control node is a physical or virtual Linux machine used to manage the cluster. It must meet the following requirements:

- Meet the requirements for the BeeGFS HA role including the installed versions of Ansible, Python, and any

additional Python packages.

- Meet the official Ansible control node requirements including operating system versions.
- Have SSH and HTTPS access to all file and block nodes.

Detailed installation steps can be found here.

## Copyright information



## Trademark information